FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

Evaluación de algoritmos criptográficos para mejorar la seguridad de la información en el envío de texto plano por internet

PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

Autor(es)

Bach. Guerrero Vega Ericks Tito

ORCID: https://orcid.org/0000-0002-0019-4250

Bach. Lozano Delgado Kelkin Heiminn

ORCID: https://orcid.org/0000-0002-9815-0175

Asesor

Mg. Minguillo Rubio Cesar Augusto
ORCID: https://orcid.org/0000-0002-5203-7863

Línea de Investigación

Infraestructura, Tecnología y Medio Ambiente

Pimentel - Perú

2023

Evaluación de algoritmos criptográficos para mejorar la seguridad de la información en el envío de texto plano por internet

Aprobación del jurado

DR. VASQUEZ LEYVA, OLIVER

Presidente del Jurado de Tesis

MG. MINGUILLO RUBIO, CESAR AUGUSTO

Secretario del Jurado de Tesis

MG. BRAVO RUIZ, JAIME ARTURO

Vocal de Jurado de Tesis

DECLARACIÓN JURADA DE ORIGINALIDAD

Quien(es) suscribe(n) la DECLARACIÓN JURADA, soy(somos) **Guerrero Vega Ericks Tito, Lozano Delgado Kelkin Heiminn.** del Programa de Estudios de **Ingeniería de Sistemas** de la Universidad Señor de Sipán S.A.C, declaro (amos) bajo juramento que soy (somos) autor(es) del trabajo titulado:

EVALUACIÓN DE ALGORITMOS CRIPTOGRÁFICOS PARA MEJORAR LA SEGURIDAD DE LA INFORMACIÓN EN EL ENVÍO DE TEXTO PLANO POR INTERNET

El texto de mi trabajo de investigación responde y respeta lo indicado en el Código de Ética del Comité Institucional de Ética en Investigación de la Universidad Señor de Sipán, conforme a los principios y lineamientos detallados en dicho documento, en relación con las citas y referencias bibliográficas, respetando el derecho de propiedad intelectual, por lo cual informo que la investigación cumple con ser inédito, original y autentico.

En virtud de lo antes mencionado, firman:

Guerrero Vega Ericks Tito	DNI: 46659948	
Lozano Delgado Kelkin Heiminn	DNI: 74751359	

Pimentel, 04 de diciembre de 2023.

Dedicatoria

Esta tesis está dedicada en primer lugar a Dios, quien ha sido mi guía y fortaleza que me permitió alcanzar este momento tan crucial en mi formación profesional, uno de mis anhelos más deseados. Quiero expresar mi profundo agradecimiento a mis padres, Eugenio Lozano y Amelida Delgado, por su inmenso amor, arduo trabajo y sacrificio a lo largo de todos estos años. Su constante motivación ha sido fundamental para que hoy en día sea una persona profesional. Agradezco también a mis hermanas y a todas aquellas personas tan significativas en mi vida por su apoyo incondicional durante este proceso, permitiéndome alcanzar una de mis metas más importantes.

Kelkin Lozano

A mis familiares, en especial a mis queridos padres, María Rosa Vega Contreras y Dagoberto Guerrero Delgado así mismo a mis hermanos, Por brindarme el apoyo mutuo en el desarrollo de este proceso de formación personal y profesional. A ustedes les dedico cada logro y éxito, pues su amor y respaldo fueron el cimiento que sustentó cada paso en este camino.

A mis docentes, agradezco sinceramente su dedicación, motivación y enseñanzas que no solo me brindaron conocimientos, sino que también me impulsaron a alcanzar mis objetivos. Cada lección compartida contribuyó de manera invaluable a mi crecimiento académico y personal. ¡Gracias por ser fundamentales en mi trayectoria educativa!

Ericks Guerrero

Agradecimiento

Quiero expresar mi agradecimiento a mi familia extendida, amigos cercanos y seres queridos por su constante aliento y comprensión. Cada palabra de ánimo ha sido un impulso de superación en los momentos difíciles.

Expresar mi gratitud a los docentes que, de diversas maneras han intervenido en el desarrollo de esta tesis. Su orientación y conocimientos han enriquecido de manera significativa mi experiencia, brindándome perspectivas valiosas y contribuyendo al crecimiento académico de esta investigación.

Índice

Índice de tablas, figuras y fórmulas

Índice de tablas

Resumen

Actualmente, los sitios y aplicaciones, se han catalogado como el principal medio de comunicación para transmitir todo tipo de información, generando diariamente más de 2.5 quintillones de bytes de datos, las cuales son el blanco de los ciberdelincuentes, quienes continuamente buscan vulnerarlas. En esta investigación se pretendió evaluar algoritmos criptográficos para identificar con mayor certeza cuales de ellos garantizan la seguridad de la información en el envío del texto plano por internet. Para ello, en primera instancia, se realizó una revisión de la literatura que permitió seleccionar aquellos algoritmos existentes; del mismo modo se identificaron los principales ataques que vulneran la seguridad de la información; posteriormente, se desarrollaron los algoritmos en lenguaje de programación Phyton, empleando IDE PyCharm, considerando un contexto de archivos pertenecientes a documentos empresariales con extensión ".docx" y finalmente se mide la satisfacción del usuario. Los resultados obtenidos revelaron que AES-256 garantizó mayor seguridad en la encriptación de documentos, en el rendimiento de cifrado y descifrado teniendo mejores resultados, en cuanto a los seis Principios Criptográficos de Kerckhoffs obtuvo 100% de cumplimiento, en comparación de ChaCha20 y Blowfish, en la integridad AES-256 y ChaCha20 alcanzaron un 100% superando a Blowfish. Se concluyó que, en un contexto de envío de información mediante documentos empresariales, AES-256 presenta mejor valoración, protegiendo la información que es transportadas en el envío del texto plano por internet.

Palabras Clave: Algoritmos criptográficos, Criptografía, Criptoanálisis, seguridad de la información, Texto cifrado.

Abstract

Currently, websites and applications have become the main means of communication for transmitting all kinds of information, generating more than 2.5 quintillion bytes of data daily, which are the target of cybercriminals, who continually seek to breach them. The aim of this research was to evaluate cryptographic algorithms in order to identify with greater certainty which of them guarantee the security of information when sending plain text over the Internet. In order to do this, firstly, a literature review was carried out to select the existing algorithms; in the same way, the main attacks that breach information security were identified; subsequently, the algorithms were developed in Phyton programming language, using IDE PyCharm, considering a context of files belonging to business documents with extension ".docx" and finally, user satisfaction was measured. The results obtained revealed that AES-256 guaranteed greater security in the encryption of documents, in the performance of encryption and decryption having better results, in terms of the six Kerckhoffs Cryptographic Principles it obtained 100% compliance, compared to ChaCha20 and Blowfish, in integrity AES-256 and ChaCha20 reached 100%, surpassing Blowfish. It was concluded that, in a context of sending information via business documents, AES-256 is better rated, protecting the information that is transported in sending plaintext over the internet.

Keywords: Cryptographic algorithms, Cryptography, Cryptanalysis, information security, Ciphertext.

I. INTRODUCCIÓN

1.1. Realidad problemática

En la actualidad, los sitios y aplicaciones que están alojadas en la nube, se han catalogado como el principal medio de comunicación para difundir e intercambiar todo tipo de información documentos empresariales, videos, imágenes y más [1]. Se estima que existen aproximadamente 1.1 billones de sitios web activos [2], generando diariamente más de 2.5 quintillones de bytes de datos [3]. WikiLeaks, la plataforma de filtraciones creada por Julian Assange, logró filtrar documentos confidenciales estadounidenses, desencadenando importantes escándalos políticos y financieros [4]. A pesar de los beneficios de la tecnología, toda esta información transmitida por la red es blanco de los ciberdelincuentes, quienes utilizan diferentes técnicas de ataques, métodos y herramientas para vulnerar, generando pérdidas económicas significativas para individuos, empresas y gobiernos [5].

Las organizaciones nacionales como internacionales, en gran medida por desconocimiento y/u otros factores son deficientes los controles de seguridad que han implementado para asegurar la confidencialidad e integridad de la información, haciendo que los documentos confidenciales que se transmiten por internet sean vulnerables a los diferentes ataques informáticos [6]. La complejidad del ecosistema de las aplicaciones y software es otro problema que enfrentan las organizaciones [7].

América Latina no es indiferente ante esta problemática. Los ciberdelitos se incrementan todos los días, las denuncias ascienden en lo referido a delito informático y suplantación de identidad, en un promedio de 300 denuncias al mes según el reporte de la División de Investigación de Delitos de Alta Tecnología (DIVINDAD) registrado en Perú [8], siendo las organizaciones del sector financiero y telecomunicaciones los más afectados con un total de 136 denuncias por mes, debido al incumplimiento con la protección y seguridad de los datos personales [9].

En Norte América, WikiLeaks, la plataforma de filtraciones creada por Julian Assange, logró filtrar documentos confidenciales estadounidenses, desencadenando importantes escándalos políticos y financieros, donde se filtró cientos de miles de informes relacionados con la guerra de Irak, unos 90.000 informes de la guerra en Afganistán, alrededor de 800 informes desde la prisión de Guantánamo y más de 250.000 cables diplomáticos redactados en diferentes regiones del mundo [4]. El departamento de justicia de Estados Unidos retuvo medio millón de dólares en bitcoins a supuestos piratas informáticos norcoreanos, su principal objetivo de ataque fueron los proveedores de atención médica, con un nuevo tipo de ransomware, un software de captura de documentos confidenciales [10]. Se reporta que los sistemas de salud de una región en el sureste fueron comprometidos por hackers, logrando acceder a información personal y financiera confidencial a más de 1,3 millones de individuos. De ese total, alrededor de 470 personas afectadas residen en Maine. Asimismo, se registró un ataque de ransomware en la división de los Ángeles de planificación familiar, lo que ocasionó la exposición de datos personales de alrededor de 400.000 pacientes [11].

En España durante el año 2022, se registraron 375.506 delitos informáticos, representando un aumento del 72% en comparación con el año 2019, que se tomó como referencia previa antes de la pandemia. La gran mayoría de estos delitos digitales fueron fraudes o estafas, siendo 336.778 de las infracciones registradas de este tipo, lo que equivale a casi el 90% del total [12]. La Oficina de Seguridad del Internauta (OSI) de España, manifiesta que existen casos donde los ciberdelincuentes logran enviar falsos mensajes con el objetivo de hace creer que son pertenecientes a las verdaderas entidades bancarias. El objetivo principal es conseguir dinero de diferentes modalidades, según la OSI existen actualmente 9 casos de suplantación de identidad en bancos que funcionan en España [13]. El mayor ataque de ciberdelitos se ha realizado en contra del patrimonio de los usuarios, el 20% de delitos han sido denunciados. Estos delitos han incrementado un 24% en el primer semestre del 2021

con un total de 9824 incidentes y en el año 2020 registró 7962 [14].

La seguridad de los datos transmitidos por la red pública de internet no es segura, se evaluaron tres técnicas de cifrado (DES, AES y RSA) y una técnica de esteganografía (LSB) para proteger la información confidencial durante la transmisión, el rendimiento de estas técnicas fue comparado en términos de tiempo de encriptar y desencriptar. El algoritmo AES presentó un tiempo de encriptado de 2,0 segundos para paquetes de 868 (KB), los otros algoritmos ocuparon más tiempo para la encriptación, mientras que, para la desencriptación AES presentó un tiempo de 1,2 segundos, siendo más efectivo que DES y RSA en la protección de la información durante la transmisión. Esto indica que el algoritmo AES es más eficiente y efectivo para proteger la información transmitida en comparación con los otros algoritmos evaluados [15].

Para determinar el algoritmo de criptografía simétrica más adecuado para un sistema independiente, se realizó un análisis comparativo de diversos algoritmos criptográficos. Se evaluaron aspectos como los tiempos de encriptación, desencriptación, el rendimiento y el tamaño del texto cifrado en comparación con el texto plano para los algoritmos AES, ARC2, Blowfish, CAST y 3DES. Los algoritmos seleccionados se cifraron con un tamaño de 8 MB, y se observó que el algoritmo Blowfish presentó un tiempo de cifrado de 0,137474 segundos, mientras que los demás algoritmos presentaron tiempos de cifrado más extensos. Para el descifrado, Blowfish también fue el más eficiente, con un tiempo de 0,137718 segundos. Los resultados demostraron que el algoritmo Blowfish es el más efectivo en comparación con los otros algoritmos evaluados [16].

Se evaluaron tres algoritmos de encriptación, AES, 3DES y ChaCha20, para determinar su eficiencia y rapidez. Teniendo en cuenta los aspectos de tiempo de ejecución, consumo de energía y el porcentaje de la Unidad de Procesamiento Gráfico (GPU), para cifrar

y descifrar un conjunto de 150 archivos con distintos formatos y tamaños. En la encriptación, Chacha20 tuvo un consumo de CPU del 10-15%, mientras que los demás algoritmos tuvieron un consumo del 15-30%, en la desencriptación presentan resultados similares. El consumo de energía en la encriptación fue de 15-20 Watts para Chacha20 y de 20-25 Watts para los otros algoritmos. Se determina que ChaCha20 es el algoritmo más eficiente, mientras que 3DES es el menos eficiente de los tres algoritmos evaluados, obteniendo bajos resultados en todos los aspectos medidos [17].

En cuanto a otros métodos de encriptación de información se realizó una solución innovadora en el área de seguridad de la información al combinar los algoritmos One Time Pad (OTP) y Vigenere, el enfoque implicó la generación de claves aleatorias, utilizando listas enlazadas (de 4 tuplas) distribuidas en varias direcciones IP de sitios web. Se calculó la prueba promedio de tiempo en el proceso con el algoritmo OTP, con la cantidad 100 de caracteres obteniendo el tiempo de ejecución de 7.33 milisegundos, en cambio; con el algoritmo Vigenere con la misma cantidad de caracteres con un tiempo de 132.333 milisegundos, mostrando mejores resultados en tiempo de ejecución el algoritmo OTP [18].

El modo de operación de cifrado de bloque Cipher-block chaining (CBC) se utiliza comúnmente junto con funciones hash para garantizar la autenticidad de la información, este algoritmo es especialmente efectivo para cifrar grandes cantidades de datos [19], Los algoritmos Output feedback (OFB), Cipher feedback (CFB) y Rivest Cipher (RC2) utilizan un conjunto de claves para combinar el texto plano en el cifrado de información. Al hacer esto, convierte el cifrado de bloque en un flujo, lo que elimina la necesidad de relleno y facilita la encriptación de datos [20], El algoritmo Chacha20 es conocido por su eficiente cifrado de flujo, lo que hace muy seguro en términos de protección de datos. Este algoritmo fue específicamente diseñado para permitir implementaciones de software rápidas y seguras, lo que lo hace muy popular en el ámbito de la seguridad informática. [21], Digital Signature

Algorithm (DSA) es utilizado para encriptar y autenticar firmas digitales. Este algoritmo fue desarrollado por el Instituto Nacional de Estándares y Tecnología (NIST) [22], Es importante destacar que el algoritmo de cifrado Base64 es vulnerable, por lo que no se recomienda utilizarlo como medida de protección para datos confidenciales [23].

Como se puede evidenciar en lo antes mencionado, son diferentes los métodos y algoritmos implementados para encriptar documentos confidenciales que se transmiten por internet, sin embargo; se evidencia que aún existen vacíos por explorar e investigar. Se han seleccionado cuidadosamente los algoritmos criptográficos de clave simétrica Blowfish, AES-256 y Chacha20, los cuales han demostrado excelentes resultados en cuanto a seguridad de la información, específicamente, en términos de rendimiento de encriptar, desencriptar, así como garantizar la integridad. Para ello, se realizó una exhaustiva investigación consultando diversos artículos científicos relevantes en el área de la criptografía. Todos los datos recopilados fueron organizados en una hoja de cálculo para realizar un análisis detallado de cada algoritmo y poder seleccionar aquellos que resultaron ser más adecuados para los objetivos de este trabajo, para esta selección se tuvo en cuenta los criterios de tiempo de cifrado y descifrado en segundos (s), tamaños de archivos en bytes (B), rendimiento de cifrado y descifrado, fortaleza del algoritmo donde está relacionado con el cumplimiento de los seis (06) Principios Criptográficos de Kerckhoffs y la integridad.

1.2. Formulación del problema

¿Cómo asegurar los datos de un texto plano transmitida por la red pública de internet?

1.3. Hipótesis

Mediante la implementación de la criptografía permitirá asegurar los datos de un texto plano transmitido por la red pública de internet

1.4. Objetivos

1.4.1. Objetivo general

Evaluar algoritmos criptográficos para mejorar la seguridad de la información en el envío de texto plano por internet.

1.4.2. Objetivos específicos

a) Seleccionar los algoritmos criptográficos más relevantes que garanticen la seguridad de los datos.

b) Identificar y analizar los principales ataques criptográficos dirigidos a documentos encriptados.

c) Desarrollar en lenguaje de programación los algoritmos criptográficos para cifrar los datos de un texto plano.

d) Proponer recomendaciones precisas que promuevan la ejecución segura y medir la satisfacción del usuario.

1.5. Teorías relacionadas al tema

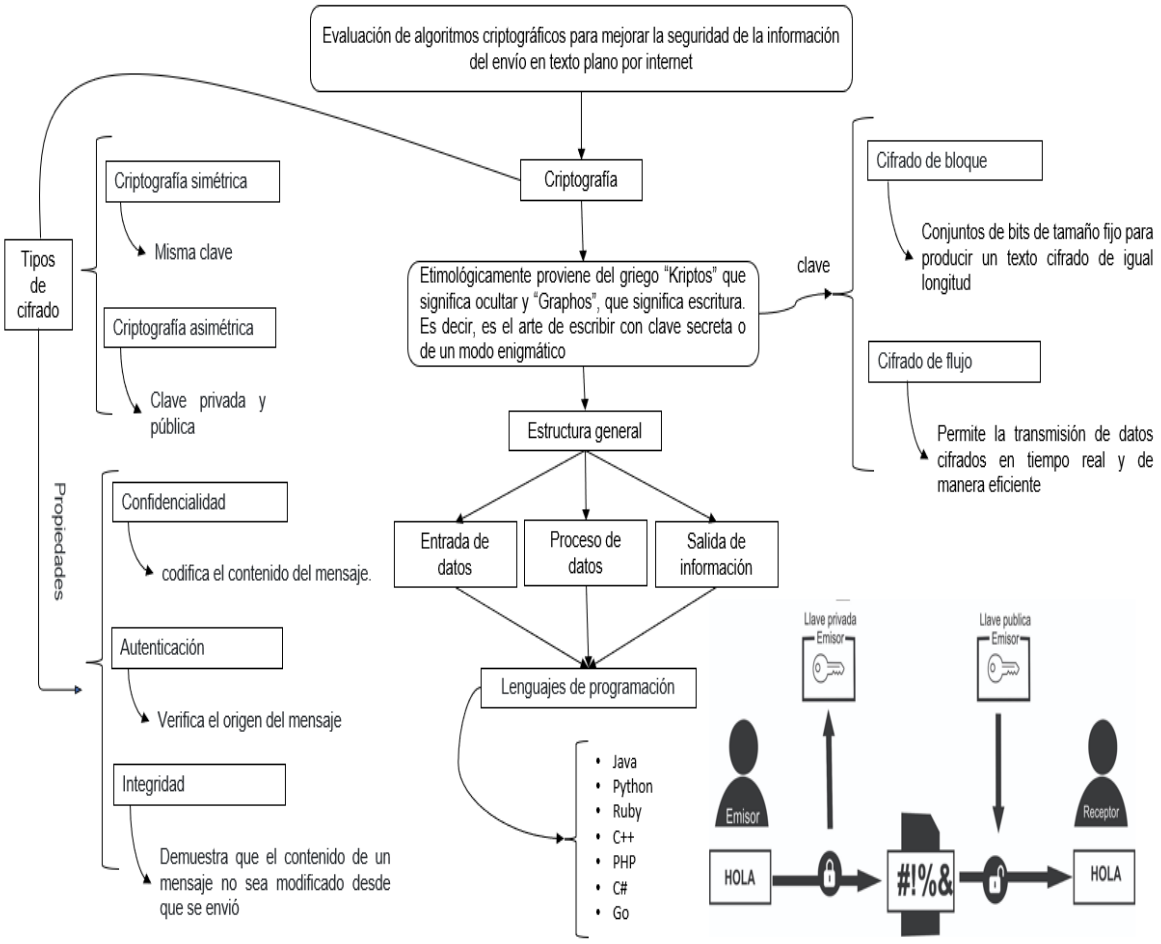


Fig. 1. Teorías relacionadas al tema.

1.5.1. Seguridad

La seguridad es ampliamente utilizada, en otras palabras, hace referencia a grandes rasgos, este concepto proviene del latín "securitas" de la (Real Academia Española, 2001), hace hincapié a las características de seguridad, es decir; hace mención a algo donde no va a ocurrir algún peligro, daños ni riesgos. Se logra entender que una entidad es segura si el comportamiento es como se desea, sin que tenga lugar a registrarse actos inoportunos. Para gran parte de los expertos la palabra seguridad informática es imposible porque no existe un sitio 100% seguro, la debilidad en los sistemas se interpreta como la exposición a ataques informáticos. Para lograr obtener un sistema seguro se debe garantizar y cubrir las propiedades de confidencialidad, integridad, disponibilidad y no repudio [24].

1.5.2. Confidencialidad

Encargada de gestionar la seguridad de la información para mitigar los posibles riesgos en caso existan, y que estén asociados a los riesgos de personas no autorizadas de la información, brindando una seguridad en la privacidad de datos [25] . En el campo de la criptografía, se emplea la técnica de cifrado para transformar los datos en un estado ilegible para cualquier individuo que carezca de la clave de descifrado apropiada, lo que garantiza que solo las personas con los permisos pertinentes puedan acceder a la información.

1.5.3. Integridad

La integridad implica asegurar que los datos no sean modificados de manera no autorizada, accidental o malintencionada, y que la información se mantenga consistente y exacta. Es fundamental para garantizar que la información permanezca completa y sin cambios desde su origen hasta el momento en que se consulta o emplea.

1.5.4. Disponibilidad

En este punto se encarga de la verificación que la información esté disponible para las personas autorizadas, la función principal es prevenir interrupciones no autorizadas, y sean controlados por los recursos informáticos.

1.5.5. Datos

Un dato es una forma de representar una variable, ya sea cuantitativa o cualitativa, asignándole un número, letra o símbolo. Usualmente, los datos se derivan de la realidad, lo que significa que tienen una base empírica y pueden ser utilizados para analizar hechos específicos. Sin embargo, para llevar a cabo un análisis adecuado, es necesario organizar los datos y contar con una base teórica sólida [26].

1.5.6. Información

La información es un conjunto de datos que tiene significado y, cuando está organizado, proporciona conocimiento para establecer ideas, metas o acciones relacionadas con un tema en particular. El término "información" tiene muchas definiciones y se aplica

ampliamente en campos de investigación como las ciencias sociales, las comunicaciones, la biología y la informática [27].

1.5.7. Seguridad de la información

La seguridad desempeña un papel crucial en la protección de la información independientemente de su formato, ya sea en papel, electrónico u otro medio. Esto incluye la protección de datos personales, información financiera, propiedad intelectual, información gubernamental, sistemas y redes de una organización, El objetivo principal de la seguridad es garantizar que solo las personas autorizadas tengan acceso a la información, asegurando su integridad y disponibilidad cuando sea necesario. Además, se busca proteger cualquier otra información confidencial que pueda ser de interés para posibles atacantes [28].

1.5.8. Seguridad informática

La seguridad informática engloba una serie de acciones, enfoques, técnicas, herramientas y protocolos empleados para salvaguardar la integridad de los sistemas informáticos y la información que se encuentra en ellos. El propósito final de estas medidas es asegurar que tanto los equipos informáticos como los datos que contienen no sean vulnerados o alterados de manera no autorizada, y se encuentren en un estado óptimo de funcionamiento [29].

1.5.9. Criptografía

La Criptografía se enfoca en la investigación de los sistemas de códigos y contraseñas utilizados para mantener la confidencialidad en la comunicación de información entre una fuente y un destinatario. Esta fuente y destinatario pueden corresponder tanto a individuos, como a procesos dentro de un sistema informático conectado a una red, o incluso a datos almacenados en medios como discos de computadora u otros dispositivos de almacenamiento [30].

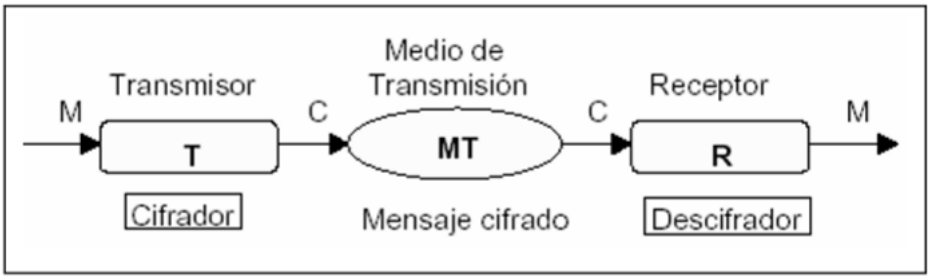


Fig. 2. Proceso de cifrado.

Nota. Fuente: [30].

Criptografía es el término colectivo para dos campos opuestos pero complementarios, criptografía y criptoanálisis. La criptografía se encarga del diseño de métodos de cifrado. En otras palabras, enmascara cierta información sensible. El criptoanálisis cubre la función de romper estos pasos y recuperar la información. Ambos campos siempre se han desarrollado en paralelo, cada método de cifrado siempre va acompañado de un criptoanálisis correspondiente [31].

1.5.10. Algoritmo

Los algoritmos es una rama de la ciencia que se enfoca en la construcción de algoritmos para resolver una variedad de problemas. Un algoritmo es un proceso o flujo de trabajo que está estructurado con órdenes específicas, que son configuradas previamente por el creador del algoritmo. Aunque los algoritmos son utilizados con mayor frecuencia en matemáticas, lógica, ciencias de la computación y disciplinas relacionadas, también se están utilizando cada vez más en otras áreas, como el periodismo, debido al aumento del proceso y análisis de altas cantidades de datos y el fácil acceso a la información [32].

El propósito de un algoritmo criptográfico es hacer que el descifrado de datos sea extremadamente difícil cuando no se utiliza la clave correcta. Si se utiliza un algoritmo de encriptación robusto, es muy poco probable que se pueda utilizar una técnica que pruebe sistemáticamente todas las posibles claves para descifrar los datos [33].

1.5.11. Encriptado de bloque

La encriptación en bloque es utilizada en conjuntos de bits de tamaño fijo para producir

un texto encriptado de igual longitud. Sin embargo, en la realidad, es necesario cifrar datos de tamaño variable y para ello se combinan diferentes técnicas llamadas Modos de Operación con el algoritmo de cifrado de bloque. Para que un bloque cifrado sea considerado seguro, debe ser una permutación pseudoaleatorio de bits, es decir, un conjunto de bits que sean indistinguibles de bits generados completamente de manera aleatoria. Esto se asemeja a lanzar un grupo de bits y afirmar que es la salida del algoritmo [34].

1.5.12. Encriptado de flujo

En el campo de la criptografía, existe un interés particular en el encriptado de flujo, centrándose en la combinación de la información a transmitir con una secuencia pseudo-aleatoria generada por un dispositivo específico. La seguridad que se obtiene en este tipo de encriptado depende de las propiedades del generador utilizado. El encriptado de flujo tiene como objetivo replicar el sistema de cifrado One-Time-Pad que fue propuesto por G. Vernam en 1917 [35].

1.5.13. Plaintext

Son archivos que contienen solamente texto, sin incluir información sobre el tipo de letra, formato (negritas, subrayados) o tamaño. Los archivos de texto plano, también conocidos como archivos de texto llano, simple o sin formato, consisten únicamente de texto sin ningún tipo de formato, lo que significa que no requieren ser interpretados para ser leídos [36].

1.5.14. Nonce

Es un valor aleatorio y no repetido que se utiliza en sistemas de seguridad para cifrar información. Este valor es generado mediante un dispositivo que genera números aleatorios, ya sea mediante software o hardware, y es utilizado en una función de seguridad específica después de su creación. La función del nonce es garantizar la seguridad de la comunicación al evitar la repetición de la misma clave de cifrado y al dificultar la predicción de su valor por parte de atacantes [37].

1.5.15. TLS

El protocolo de Seguridad de la Capa de Transporte (TLS) es ampliamente empleado como una medida de protección para garantizar la privacidad y seguridad de los datos mientras se transmiten. Su función principal consiste en cifrar la información, brindando así una capa adicional de seguridad, independientemente si son texto plano o texto cifrado, con el objetivo de evitar que sean vulnerables ante ataques de hackers y otros ciberdelincuentes. La implementación de TLS a través del tráfico encriptado es una herramienta valiosa de protección para usuarios y empresas que buscan mantener la confidencialidad de sus datos [38].

1.5.16. SSL

Las Capas de Sockets Seguros (SSL), también conocidas como Capa de Conexiones Seguras, son un protocolo que utiliza certificados digitales para garantizar la seguridad de las comunicaciones en línea. Su función principal es proporcionar un ambiente seguro y protegido para la transmisión de datos a través de la autenticación y el cifrado de la información. Aunque ha sido reemplazado por el protocolo TLS, ambos están basados en SSL y son compatibles entre sí. El protocolo SSL es comúnmente utilizado en servicios que manejan información sensible, como bancos y tiendas en línea, para asegurar la transmisión segura de datos personales y contraseñas [39].

1.5.17. HTTPS

A través del Protocolo de Transferencia de Hipertexto Seguro (HTTPS), se establece una conexión cifrada entre el cliente y el servidor, asegurando que solo ellos puedan acceder y leer la información que se está transmitiendo. De este modo, cualquier máquina intermedia que intente intervenir en la comunicación, tendrá la dificultad añadida de tener que descifrar la información encriptada para acceder a ella [40].

1.5.18. Criptografía Simétrica

Esta técnica es la más antigua, en la actualidad mantiene un alto nivel de seguridad

debido al menor tiempo en velocidad al encriptar y desencriptar, la criptografía simétrica se usa para asegurar la protección de información en diferentes sistemas que hacen referencia a dispositivos y ordenadores, está basada en hacer uso de una clave individual que sirve para encriptar y desencriptar la información en tránsito con protocolos como Transport Layer Security (TLS), o que está alojados en algún disco extraíble [41].

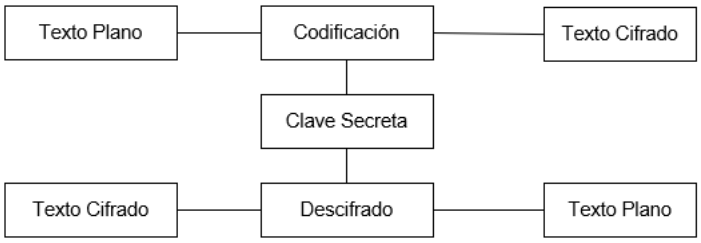


Fig. 3. Representación del proceso de clave simétrica.

Nota. Fuente: [42].

1.5.19. Criptografía Asimétrica

Está compuesta por una fórmula matemática que utiliza dos claves diferentes, la primera clave de manera pública y la segunda clave de forma privada, la clave pública está construida para todas las personas con quien se desee compartir la información, en cambio; la llave privada es aquella que tiene un destinatario fijo y solo esa persona podrá desencriptar la información, el mayor beneficio que se obtiene con la criptografía asimétrica es que genera una mayor proporción en cuanto a la seguridad de la información, este proceso es el más confiable porque los usuarios no tienen que revelar o compartir su propia clave privada [43].

Los algoritmos asimétricos derivan de los algoritmos simétricos en aspectos muy importantes, al generar una clave simétrica, simplemente selecciona un número aleatorio de longitud adecuada, el proceso es más complicado cuando se generan claves asimétricas. Los algoritmos de llave pública y privada se denominan asimétricos, utilizando una llave para el encriptado y el desencriptado, hacen uso de dos llaves diferentes. Estas dos llaves están relacionadas matemáticamente, y su propiedad esencial es hacer el proceso de cifrado difícil para que no se puede descifrar por personas fuera del entorno. Después de completar la

asignación de claves asimétricas, se definen una clave de encriptado (clave pública) y una clave de desencriptado (clave privada). La primera clave puede ser conocida por todos, en cambio, para generar la segunda clave de tipo privada se tiene que realizar con mucha precaución y poder ocultarlo de manera segura [44].

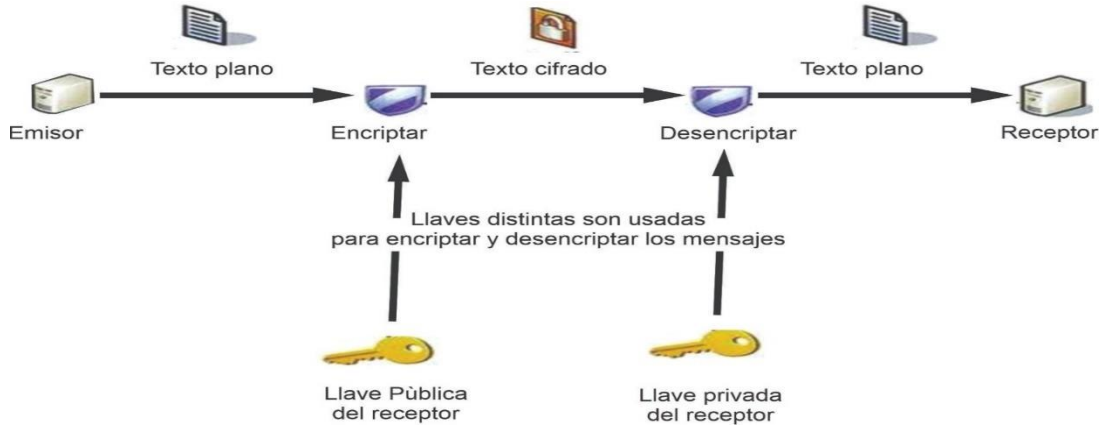


Fig. 4. Representación del funcionamiento de clave asimétrica.

Nota. Fuente: [45].

1.6. Algoritmos criptográficos de clave simétrica

1.6.1. Advanced Encription Standard (AES-128)

Tiene una función de cifrado simétrico, actualmente se usa ampliamente en protocolos de confianza, como la seguridad de transporte de capa de transporte (TLS), el protocolo de transferencia de archivos (FTPES) y la red privada virtual. Este algoritmo se utiliza para ambos softwares y hardware [46].

Ejemplo

Al utilizar una clave de 16 caracteres corresponde a un cifrado de 128 bits, esto implica la presencia de $(2)^{128}$ combinaciones o claves únicas, lo que equivale a más de 340 quintillones de posibles combinaciones, es decir, más de 340 millones de mil millones de mil millones de mil millones.

Funcionamiento

AES es un método encriptado que funciona en el bloque de datos, y su estándar se establece un bloque fijo del bloque de 128 bits. Aquí, los datos que deben estar encriptados

se dividen en un segmento de 16 bytes (128 bits), y cada uno se considera un bloque o matriz de 4x4 bytes conocido como un "estado".

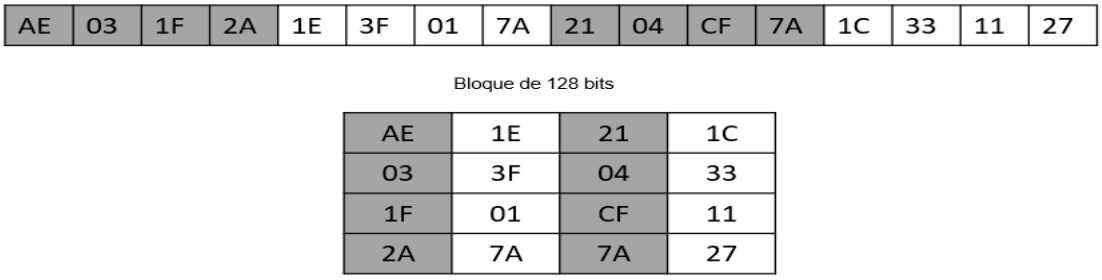


Fig. 5. Representación del algoritmo AES-128.

Nota. Fuente: [47].

1.6.2. AES-256

Nombrado en honor a sus dos creadores, los criptógrafos belgas Vicent Rijmen y Joan Daemen, el algoritmo de cifrado Rijndael fue desarrollado en 1997 como resultado de un concurso público promovido por el Instituto Nacional de Normas y Tecnología de Estados Unidos (NIST) [48].

Actualmente, se considera que el cifrado de 256 bits es altamente seguro y no se ha encontrado ninguna vulnerabilidad conocida. Aunque ha habido intentos de romper claves de AES-128 bits, intentar descifrar una clave de 256 bits por fuerza bruta se requiere de una cantidad de potencia de cómputo inimaginable, realizando la ejecución de 2^{128} posibles valores a 2^{256} posibles valores, al utilizar el cifrado AES-256 garantiza la seguridad en proteger nuestros datos, ya que su seguridad es prácticamente inexpugnable [49].

La organización de la información es un aspecto importante del proceso de cálculo en el cifrado AES-256. No sigue una estructura secuencial, los datos se dividen en bloques de matrices de 128 bits que constan de 4 filas y 4 columnas. Una vez que se han organizado los datos en estos bloques, se procede a realizar las operaciones necesarias en cada uno de ellos.

Uso del algoritmo de encriptación AES-256

a. Robustez Criptográfica: Longitud de la clave AES-256 utiliza una clave de 256 bits, lo que significa que hay $(2)^{256}$ posibles claves, lo que hace que sea extremadamente difícil de romper mediante ataques de fuerza bruta o de búsqueda exhaustiva.

b. Aprobación Estándar: Estándar Nacional de EE. UU. fue adoptado como estándar por el Instituto Nacional de Estándares y Tecnología (NIST) de los Estados Unidos en 2001 después de un proceso de selección riguroso. Su adopción como estándar nacional es un respaldo importante a su seguridad.

c. Eficiencia Computacional: Eficiencia en Hardware y Software ha sido diseñado para ser eficiente tanto en hardware como en software, lo que significa que puede ser implementado de manera rápida y sin consumir demasiados recursos.

d. Amplia Adopción: Uso Generalizado AES-256 es ampliamente utilizado en todo el mundo en una variedad de aplicaciones, desde el cifrado de datos en tránsito (por ejemplo, HTTPS) hasta el cifrado de datos almacenados.

e. Resistencia a Ataques Conocidos: Seguridad contra Ataques Criptográficos Conocidos.

Ejemplo

Cuando se emplea una clave de 32 caracteres, equivalente a un cifrado de 256 bits en criptografía, la representación de la clave se expresa como una secuencia de 256 dígitos binarios compuesta por ceros y unos, ("11010101011001101010101101010111"). Posteriormente, se realiza la conversión a formato hexadecimal para lograr una representación aún más compacta. Agrupando los bits de cuatro en cuatro, se obtiene la siguiente representación: "1101 0101 0110 0110 1010 1011 0101 0111". Luego, al convertir cada grupo de cuatro bits a su equivalente en hexadecimal, se obtendría la forma más

condensada y legible, como se ilustra a continuación: "CA D2 CA DB CA D6 73 66 CC D5 DD 6E 66 75 67".

El cifrado de 256 bits, según los estudios realizados en esta investigación, se considera invulnerable. A pesar de los intentos realizados para descifrar claves de 256 bits con AES-128, es crucial tener en cuenta que el intento de vulnerar una clave de 256 bits requeriría una potencia de cómputo por fuerza bruta $(2)^{128}$ veces mayor, ya que se pasa de $(2)^{128}$ posibles valores a $(2)^{256}$ posibles valores. Incluso con una capacidad de cálculo considerable y utilizando la tecnología actual, el tiempo necesario para descifrar una sola clave superaría más del doble de la edad total del universo.

Funcionamiento

a. **Expansión de Claves y AddRoundKey**. La primera ronda del cifrado AES en la cual se manejan las claves y se realiza una operación lógica XOR con una porción específica de la clave en cada ronda.

b. **SubBytes**. En la fase de Sustitución de Bytes, se reemplazan los bytes de todo el bloque por valores alternativos que son virtualmente imposibles de descifrar utilizando técnicas convencionales de computación.

c. **ShiftRows**. Implica la transferencia de los datos contenidos en la matriz.

d. **MixColumns**. Se ejecuta una operación XOR y una multiplicación de matrices, lo cual produce una matriz con 4 filas y 1 columna como resultado. De esta manera, los datos que anteriormente estaban organizados en una matriz retornan a ser secuenciales.

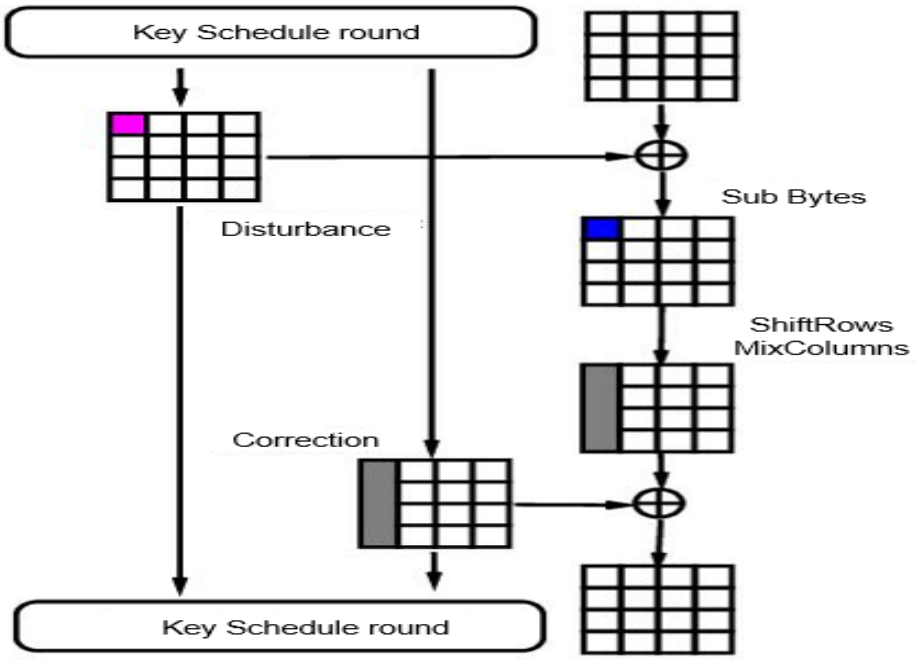


Fig. 6. Representación del algoritmo AES-256.

Nota. Fuente: [48].

1.6.3. ChaCha20

Es un algoritmo para el cifrado de flujo, diseñado para garantizar una alta seguridad y velocidad en el cifrado de datos. Fue desarrollado por Daniel J. Bernstein en 2008 como una alternativa más efectiva a otros algoritmos de cifrado de flujo, como RC4 y SALSA20. El algoritmo utiliza una clave de 256 bits y Nonce (el número se usa solo una vez) de 96 bits para generar una secuencia de bypass pseudo - elemental, que se usa para encriptar y descifrar datos [50].

ChaCha20 es un algoritmo de encriptación simétrica que tiene la capacidad de soportar claves de 128 y 256 bits y de avanzada velocidad, ChaCha20 se caracteriza por un cifrado de flujo, es mucho más eficiente que AES, por lo cual se ha distribuido rápidamente y se ha implementado en la actualidad, se emplea ampliamente en conexiones de Hypertext Transfer Protocol Secure (HTTPS), proporcionando un alto rendimiento así se tenga aceleración por Hardware, también se emplea en las conexiones Intérprete de Ordenador Seguro (SSH) para administrar los servidores, de este modo no solo se podrá administrar, sino, utilizar el protocolo de transferencia de fichero relacionado en SSH que es Protocolo de

Transferencia Seguro de Archivos (SFTP) [21].

En cada ronda del algoritmo ChaCha20, la función de trimestre se aplica cuatro veces al bloque de inicio. Los valores seleccionados para la función en cada ronda cambian inicialmente a una ronda de columna, seguido de una ronda diagonal, y este patrón se repite hasta que se llevan a cabo las 20 rondas del algoritmo [17].

Funcionamiento

Ronda columnas

Quarter round $(X_0,\ X_4,\ X_8, X_{12})$

Quarter round $(X_1,\ X_5,\ X_9, X_{13})$

Quarter round $(X_2,\ X_6,\ X_{10}, X_{14})$

Quarter round $(X_3,\ X_7,\ X_{11}, X_{15})$

Rondas diagonales

Quarter round $(X_0,\ X_5,\ X_{10}, X_{15})$

Quarter round $(X_1,\ X_6,\ X_{11}, X_{12})$

Quarter round $(X_2,\ X_7,\ X_8, X_{13})$

Quarter round $(X_3,\ X_4,\ X_9, X_{14})$

Teniendo en cuenta que se ha asignado un número del 0 al 15 a cada uno de los elementos del bloque.

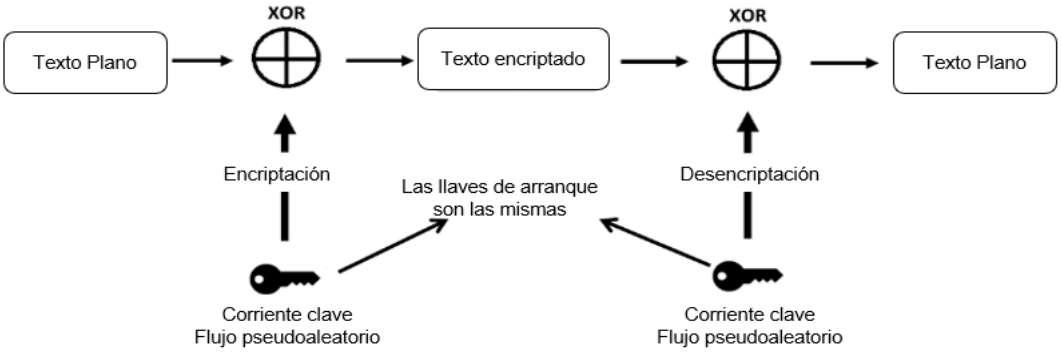


Fig. 7. Representación del algoritmo ChaCha20.

Nota. Fuente: [51].

Uso del algoritmo de encriptación ChaCha20

a. Seguridad y Resistencia Criptográfica: Fue diseñado para proporcionar un alto nivel de seguridad.

b. Eficiencia en Software: Es conocido por ser eficiente en software, su implementación es relativamente rápida en diversas plataformas, lo que puede ser beneficioso en dispositivos con recursos limitados, como dispositivos móviles.

c. Flexibilidad y Parametrización: Permite cierta flexibilidad en la elección de parámetros, especifica el número de rondas de operaciones, lo que permite adaptar la velocidad de cifrado a tus necesidades de seguridad y rendimiento.

d. Resistencia a Ataques Lado Canal: Es considerado resistente a ciertos tipos de ataques de canal lateral, como los basados en el consumo de energía o el tiempo de ejecución.

1.6.4. Blowfish

En 1993, Bruce Schneier desarrolló el algoritmo Blowfish como una alternativa al estándar de cifrado DES. Blowfish es conocido por su capacidad de ofrecer un alto nivel de seguridad, hasta el momento, no se conoce ningún criptoanálisis efectivo que pueda vulnerarlo. La clave de longitud variable del algoritmo lo hace extremadamente versátil y le confiere una mayor velocidad de cifrado en comparación con otros algoritmos de cifrado, como DES e IDEA [52].

Funcionamiento

a. El mensaje original, que consta de 64 bits, se divide en dos partes de 32 bits cada una.

b. La primera mitad, también conocida como "izquierda", se somete a una operación XOR con el primer elemento de una matriz llamada P, produciendo un valor llamado Pi. A continuación, se aplica una función de transformación llamada F.

c. La segunda mitad, o "derecha", del mensaje se somete a una operación XOR con Pi para generar un nuevo valor de F.

d. El valor de F reemplaza la mitad "izquierda" del mensaje, mientras que P reemplaza la mitad "derecha".

e. Este proceso se repite 15 veces más utilizando elementos sucesivos de la matriz P.

f. Los valores obtenidos de P y F se someten a una operación XOR con las dos últimas entradas de la matriz P (entradas 17 y 18) y se combinan para generar el mensaje cifrado de 64 bits.

g. Este algoritmo toma un mensaje de 64 bits, lo divide en dos partes, realiza varias operaciones XOR y transformaciones, y luego lo combina nuevamente para producir el mensaje cifrado de 64 bits.

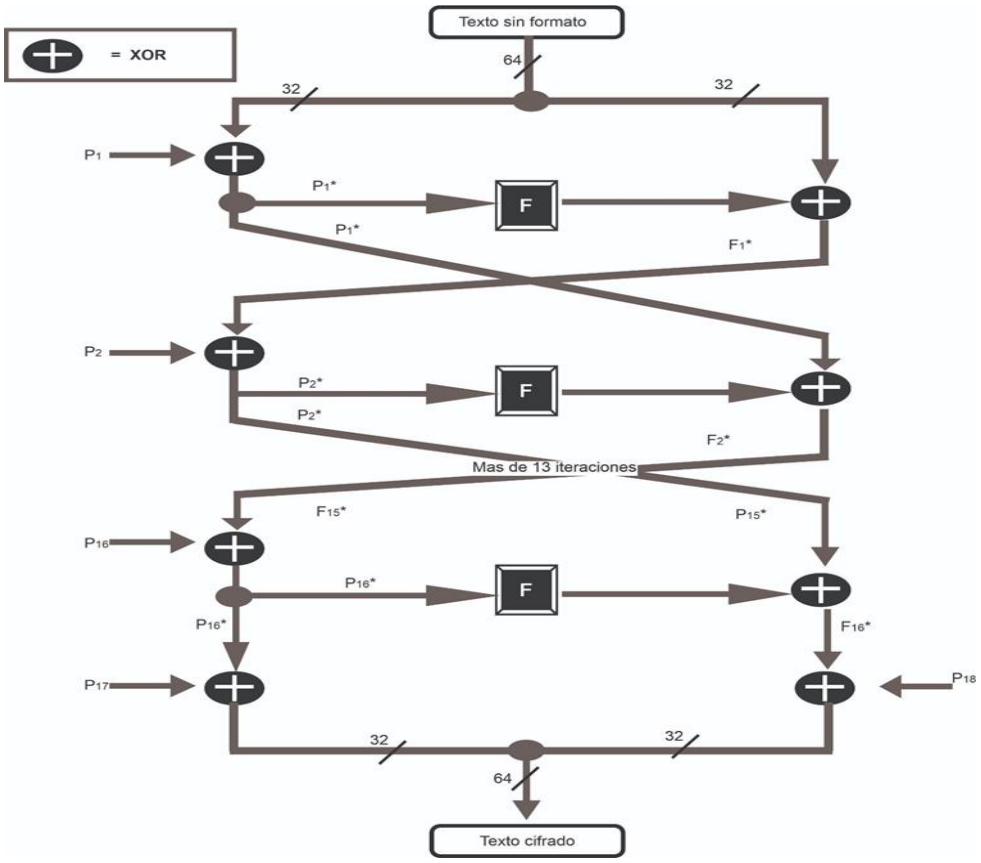


Fig. 8. Representación del algoritmo Blowfish.

*Nota***.** Fuente: [53].

Uso del algoritmo de encriptación Blowfish

a. Velocidad de Cifrado y Descifrado: Es conocido por su rapidez en términos de cifrado y descifrado, esta característica puede ser ventajosa en situaciones donde el rendimiento es crítico y se requiere un cifrado eficiente.

b. Simplicidad del Algoritmo: Es un algoritmo relativamente simple y fácil de entender en comparación con algunos de los algoritmos más complejos.

c. Implementación en Hardware y Software: Puede ser implementado tanto en hardware como en software, su versatilidad permite su uso en una variedad de plataformas y dispositivos.

d. Licencia de Uso: Está disponible para su uso sin restricciones de licencia y es de dominio público, lo que significa que no hay restricciones significativas para su implementación y uso.

e. Historial de Seguridad: A lo largo de los años, Blowfish ha demostrado resistencia contra muchos tipos de ataques criptográficos.

1.6.5. One Time Pad (OTP)

En 1917, Gilbert Sandford Burnham obtuvo una patente para un dispositivo de cifrado que podría alcanzar el nivel perfecto de seguridad. Fue unos 25 años más tarde que Claude Shannon introdujo el concepto y que el algoritmo de cifrado OTP (también llamado cifrado de Vernam) podría alcanzar este nivel de seguridad [54].

Funcionamiento

a. Donde $\alpha \oplus b$ la operación bit a bit or-exclusiva (XOR) de dos secuencias de bits $a\ y\ b$ (como se presenta, si $a = a_1, \dots, a_p\ y\ b = b_1, \dots, b_p, \dots$, entonces $a \oplus b = a_1 \oplus b_1, \dots, a_p \oplus b_p$). Entonces, el esquema de cifrado se define como:

b. Si se tiene un número entero $p > 0$, los espacios de mensaje M, claves K y textos cifrados C son iguales al conjunto $\{0,1\}^p$, es decir, todo el conjunto que contiene

las posibles cadenas binarias de longitud P.

c. El proceso de generación de claves del algoritmo Gen consiste en seleccionar una cadena de $K = \{0,1\}^p$ al azar, utilizando una distribución normal, donde cada una de las 2^p cadenas en el espacio de claves tiene la misma probabilidad de ser elegida, específicamente, una probabilidad exactamente igual a 2^{-p}.

d. El funcionamiento de encriptación Enc se describe de la siguiente manera: tomando como entrada una clave $k \in \{0,1\}^p$ y un mensaje $m \in \{0,1\}^p$, la salida generada es el valor $c := k \oplus m$. Es importante destacar que este algoritmo es determinista, es decir, siempre produce el mismo resultado con la misma entrada, a diferencia del caso probabilístico.

e. El proceso de descifrado Des se ejecuta de la siguiente manera: tomando como entrada una clave $k \in \{0,1\}^p$ y un texto encriptado $c \in \{0,1\}^p$, la salida generada es el valor $m := k \oplus c$.

f. OTP es considerado un esquema de cifrado debido a que $\forall\, k, m$ se tiene $Des_k\big(Enc_k(m)\big) = k \oplus k \oplus m$.

g. De manera intuitiva, OTP satisface las condiciones del secreto perfecto, debido a que dada una cadena cifrada c, no hay forma de que un atacante pueda determinar cuál es el mensaje original m que la generó.

Para comprobar la veracidad de esta afirmación, es importante destacar que para cada posible mensaje existe una clave tal que el resultado de encriptar el mensaje con dicha clave es igual a la cadena cifrada $c = Enc\,(k, m)$, es decir $k = m \oplus c$. Además, cada clave es seleccionada con una probabilidad uniforme, sin que exista alguna clave que tenga una mayor probabilidad que otra. Al considerar esto, se puede afirmar que el proceso de encriptación no revela información alguna acerca del mensaje original que fue cifrado, ya que cada mensaje es igualmente probable de haber sido encriptado.

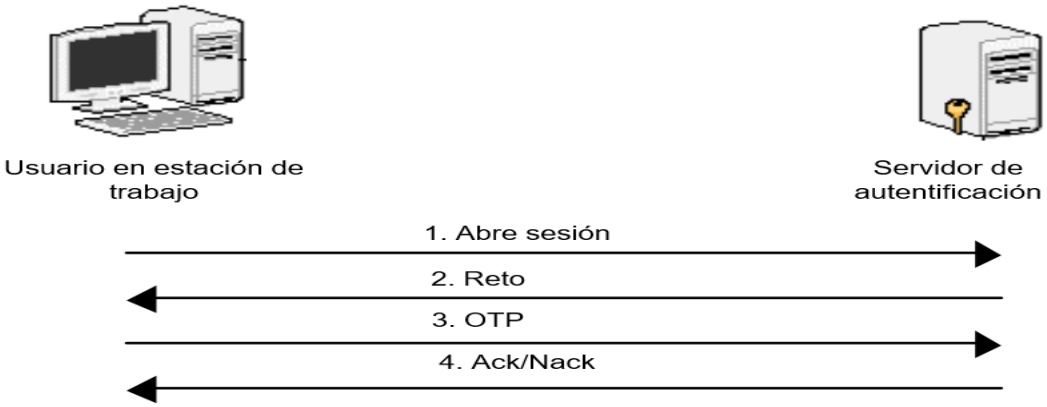


Fig. 9. Representación del algoritmo OTP.

Nota. Fuente: [55].

1.6.6. CAST-128

CAST-128 es un algoritmo de cifrado simétrico que utiliza una longitud variable de hasta 128 bits. Desarrollado en 1996 por Carlisle Adams y Stafford Tavares como una alternativa más eficiente y segura para otros algoritmos de cifrado en ese momento, como DES e Ideas, Cast-128 era una red de Feistel que constaba de 12 o 16 rondas. El tamaño del bloque es de 64 bits y la longitud de la clave varía entre 40 y 128 bits, pero siempre en aumentos de 8 bits. Es importante tener en cuenta que solo se usan 16 rondas completas cuando el tamaño de la clave excede los 80 bits [56].

Este es el formato de entrada y salida del algoritmo de cifrado CAST-128. La entrada consta de un texto plano de 64 bits y una clave de longitud variable entre 40 y 128 bits, la salida es un texto cifrado de 64 bits [57]. se detalla de la siguiente manera.

Funcionamiento

a. Generar 16 pares de subclaves $\{km_i, kr_i\}$ a partir de la clave K.

b. Se divide el texto plano en dos mitades, L0 y R0, de 32 bits cada una, donde L0 es la mitad izquierda que contiene los primeros 32 bits del texto sin formato y R0 es la mitad derecha que contiene los siguientes 32 bits del texto sin formato.

c. Durante las 16 rondas del algoritmo, se deben calcular los valores de $L_i \; y \; R_i$ para

cada iteración. Esto se logra mediante las siguientes fórmulas: L_i toma el valor de $R_i - 1$, mientras que Ri toma el valor de $L_i\ XOR\ f(R_i - 1\ Km_i, Kr_i$, donde f es una función. Es importante destacar que la función f puede ser de $Tipo_1$, $Tipo_2$, $Tipo_3$, dependiendo de la iteración i en la que se encuentre el algoritmo.

d. El texto cifrado se obtiene intercambiando los bloques finales L16 y R16 obtenidos en la última ronda del algoritmo, y luego concatenándolos en ese orden para formar el texto cifrado $c1 \ldots c64$.

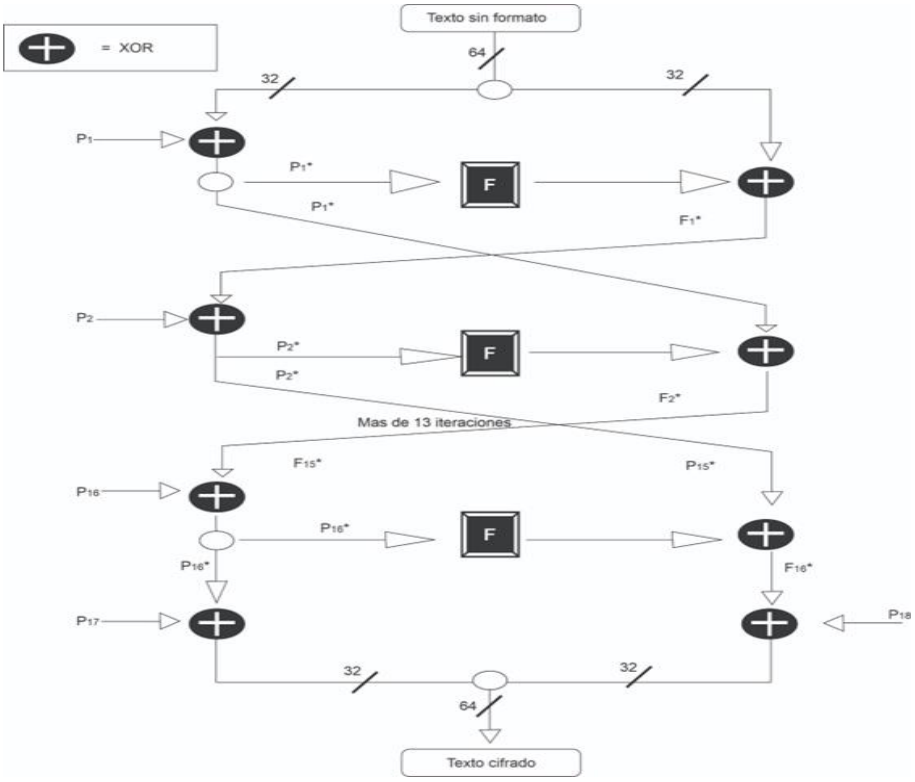


Fig. 10. Representación del algoritmo CAST-128.

Nota. Fuente: [58].

1.6.7. MISTY-1

MISTY-1 es un algoritmo de encriptación de bloques que emplea una clave de 128 bits y un número variable de rondas para cifrar datos de 64 bits. Fue desarrollado por Mitsuru Matsui de la corporación Mitsubishi Electrical, y se ha demostrado que es resistente al criptoanálisis diferencial y lineal, lo que lo convierte en un cifrado seguro [59], este algoritmo sigue los siguientes pasos.

Funcionamiento

a. **Expansión de Clave**: Genera una serie de subclaves a partir de la clave original utilizando una operación de mezcla no lineal.

b. **Rondas de Encriptación**: se aplica un conjunto de rondas de encriptación, que pueden variar en número, a los datos de entrada utilizando las subclaves generadas en la fase anterior. Cada ronda consiste en cuatro operaciones: una operación de sustitución no lineal, una operación de permutación, una operación de mezcla lineal y una operación de mezcla no lineal.

c. **Ronda Final**: se aplica una última ronda de encriptación que consta de una operación de sustitución no lineal y una operación de mezcla lineal.

d. **Salida**: el resultado final de la encriptación es el bloque cifrado.

1.6.8. HIGHT

El algoritmo HIGHT se emplea principalmente en aplicaciones que requieren un alto nivel de seguridad y un bajo consumo de energía, como la tecnología de identificación por radiofrecuencia (RFID). Se trata de un algoritmo de cifrado de bloques que combina una seguridad sólida con una estructura ligera, utilizando bloques de 64 bits y claves de 128 bits. Resulta una opción adecuada para aplicaciones con restricciones de coste y consumo energético reducido, como las tecnologías RFID. HIGHT lleva a cabo operaciones simples en un total de 32 rondas, lo que lo convierte en una opción idónea para su implementación en hardware en lugar de software [60].

1.6.9. CAMELLIA

Camellia es un algoritmo de encriptación simétrica, creado en 2000 gracias a una colaboración entre la empresa japonesa de telecomunicaciones y telegrafía NTT (Nippon Telegraph and Telephone Corporation) y la compañía de electrónica Mitsubishi Electric Corporation. Este algoritmo es altamente versátil ya que se puede implementar tanto en hardware como en software [61].

Se considera a Camellia como un estándar de seguridad criptográfica en varios países europeos, y es particularmente utilizado en sistemas de gobierno electrónico en Japón, además de su implementación en aplicaciones y protocolos de seguridad como IPsec, entre otros.

Funcionamiento

a. Camellia es un algoritmo de cifrado simétrico que utiliza bloques de 128 bits y permite claves de 128, 192 o 256 bits. Se basa en rondas de tipo Feistel, con 18 rondas para claves de 128 bits y 24 rondas para claves de 192 y 256 bits.

b. El mensaje que se desea cifrar se divide en bloques de longitud fija de 128 bits.

c. Cada bloque de entrada de 128 bits se divide en dos bloques de 64 bits, llamados Li y Ri. Se aplica una operación XOR a cada bloque con la subllave correspondiente, donde w varía de 1 a 4. Luego, ambos bloques se someten a las rondas de tipo Feistel.

d. En cada ronda, el valor de Li se obtiene mediante una operación XOR entre el bloque Ri-1 y la salida de la función F.

e. En cada ronda, el bloque Ri se iguala al bloque Li-1 de la ronda anterior.

f. En cada ronda, los bloques Li y Ri intercambian su posición.

g. En cada ciclo de 6 rondas (excepto en la última), los bloques Li y Ri se someten a las funciones FL y FL-1, respectivamente, y luego se utilizan nuevamente en las rondas siguientes.

h. Después de completar todas las rondas, se aplica una operación XOR a cada bloque con su respectiva subllave kw.

i. Las salidas resultantes de la operación anterior se concatenan en secuencia para formar el primer bloque del texto cifrado.

j. Este proceso se repite para cifrar todos los bloques que componen el texto original.

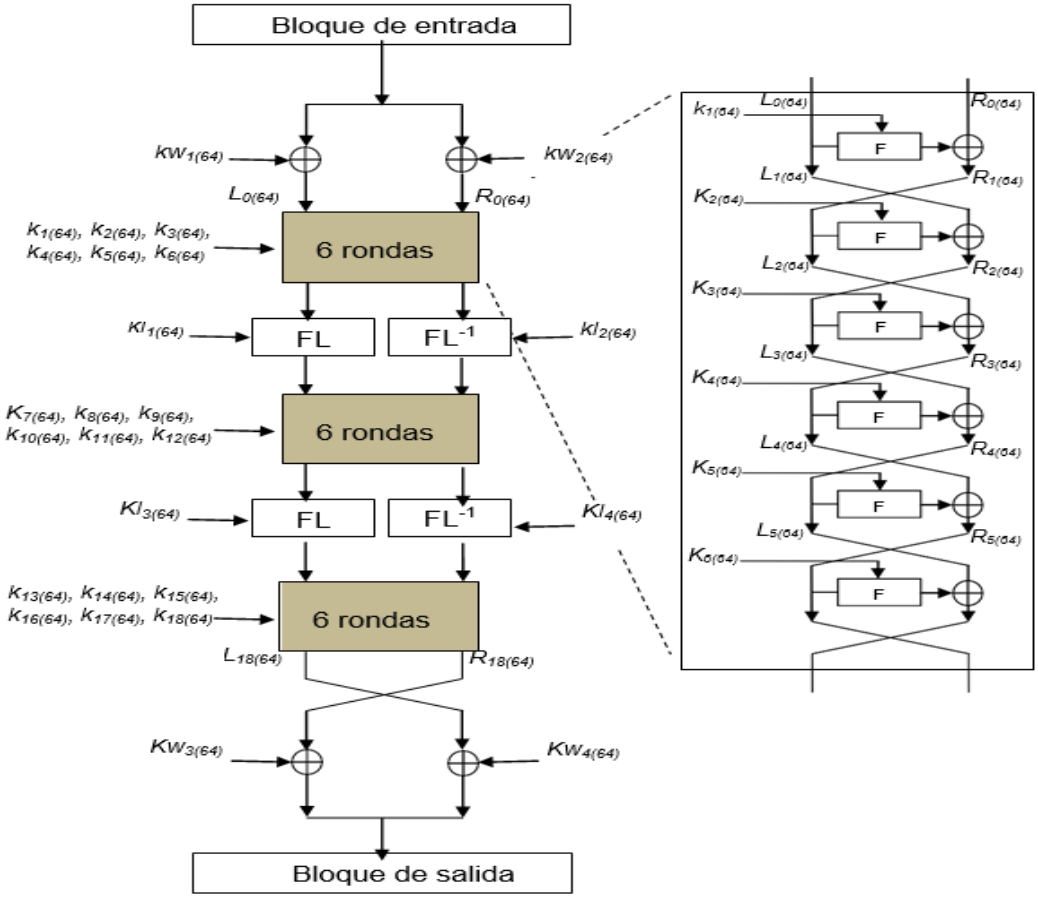


Fig. 11. Representación del algoritmo Camelia.

Nota. Fuente: [61].

1.6.10. SEED

SEED es un algoritmo de cifrado de bloques que utiliza una clave de 128 bits y bloques de 128 bits. Se basa en la estructura de red Feistel y realiza 16 rondas de cifrado. Durante cada ronda, las entradas se dividen en dos mitades de 64 bits para aplicar las operaciones de encriptado.

La primera parte es la entrada de la función F, que produce una salida que es exclusiva de la siguiente mitad y realiza un intercambio de las dos mitades de la entrada. La función F divide su entrada en dos partes de 32 bits cada una. Cada parte se aplica una subclave de 32 bits, luego la segunda parte derecha se combina exclusivamente con la parte izquierda. Después de esto, los bloques pasan a una mini-cifradora de 3 etapas que utiliza una función G y una suma modular módulo 2^{232} [62].

1.6.11. Encadenamiento de bloques de cifrado - Cipher-block chaining (CBC)

Este tipo de encriptación se utiliza ampliamente junto a la función hash con el propósito de comprobar la autenticidad de la información, En este enfoque, cada bloque de texto plano se combina con el bloque de cifrado previamente mencionado utilizando la operación XOR [63].

$$C_0 = IV$$

Fórmula de cifrado representado matemáticamente.

$$C_i = E_{k(P_i \oplus C_i-1),\ C_0} = IV$$

Fórmula de desencriptado representado matemáticamente.

$$P_i = D_{k(C_i)\oplus C_i-1,\ C_0} = IV$$

Donde:

C_0 = Vector de inicialización (IV).

C_i = Bloque cifrado i-ésimo.

E = Función de cifrado con clave K.

M_i = Bloque de texto plano i-ésimo.

XOR = Operación lógica "o exclusiva".

El proceso de cifrado CBC consiste en realizar una operación XOR entre el bloque de texto plano M_i y el bloque cifrado anterior $C_i - 1$. Posteriormente, se aplica la función de cifrado E con la clave K a este resultado para obtener el bloque cifrado actual C_i. Este proceso se repite para cada bloque de texto plano hasta que se hayan cifrado todos los bloques, con estás fórmulas se realiza para ECB, CBC, OFB y CFB [64].

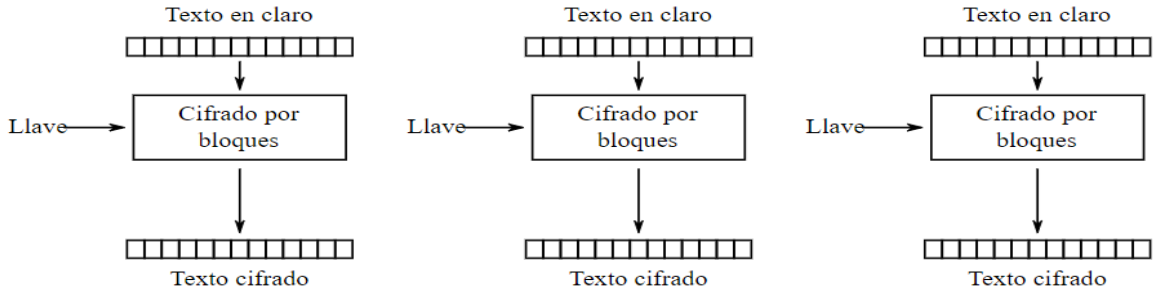


Fig. 12. Representación del algoritmo CBC.

Nota. Fuente: [65].

1.6.12. Estándar de cifrado de datos - Data Encryption Standard (DES)

El algoritmo de cifrado de bloque Data Encryption Standard (DES) es un tipo de encriptación de clave simétrica. Fue desarrollado por IBM en 1970 y posteriormente estandarizado por el Instituto Nacional de Estándares y Tecnología (NIST) de Estados Unidos en la década de 1977. En su momento, fue considerado el primer método seguro y confiable de encriptación en cumplimiento con los estándares de la época. Sin embargo, en la actualidad se considera desactualizado debido a la reducción de la longitud de las claves en bits, en el año 1994 el sistema criptográfico fue expuesto por terceros, es tan sensible que mediante un ataque de fuerza bruta se logra descifrar en poco tiempo e incluso en segundos [66].

El gobierno de USA dependía del algoritmo DES como criterio de los algoritmos de encriptación, en primer lugar, contaba con tamaño de clave de 64 bits, luego una institución de inteligencia y seguridad cibernética de los Estados Unidos, encargada de proteger los sistemas de información y la seguridad nacional del país National Security Agency (NSA), impuso una limitación para usar el Algoritmo DES con una extensión de clave de 56 bits, para esto, DES renuncia a 8 bits de la clave de 64 bits y luego usa la clave de 56 bits compactada, que derivada de la clave de 64 bits puede cifrar datos de dimensión de bloque de 64 bits [67].

La fórmula $Ri = Li - 1 + f\ (Ri - 1, Ki)$ es parte de la descripción del desarrollo de cifrado de bloques del algoritmo de encriptación DES (Data Encryption Standard). Se utiliza en cada una de las 16 rondas de transformación del algoritmo.

Donde

$Li - 1\ (Ri - 1)$ = Son los bloques de texto plano de 32 bits que se están cifrando en la ronda actual.

ki = Es la subclave de 48 bits proporcionada a esa ronda.

$Li - 1$ = El bloque se copia directamente en el siguiente bloque de texto cifrado.

Ri = El bloque se calcula mediante una función $f(Ri - 1,\ Ki)$ que toma como entrada el bloque $Ri - 1$ y la subclave ki correspondiente [68].

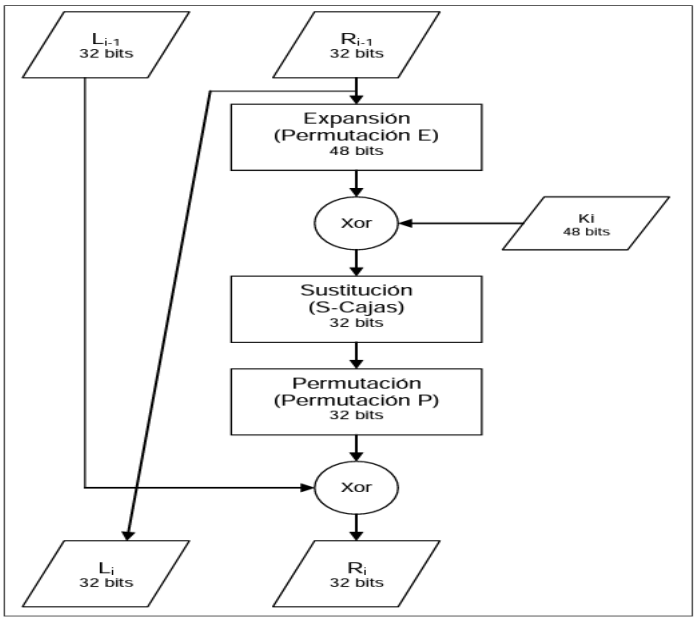


Fig. 13. Representación del algoritmo DES.

Nota. Fuente: [68].

1.6.13. TDEA ó Triple DES(3DES)

Está relacionado con el algoritmo DES, conformado por diferentes operaciones básicas con la finalidad de convertir un texto normal a uno cifrado, 3DES ejecuta un triple cifrado del algoritmo DES, asume la función de encriptar tres veces más a los datos, después que DES presentó deficiencia ante la seguridad de la información, lo utilizaron las principales plataformas como la oficina de Microsoft, Firefox, sistemas de pago Europay, MasterCard y Visa (EMV), con el transcurso del tiempo se implementaron nuevos algoritmos que presentan mayor seguridad y empezaron a reemplazarlo [69].

El algoritmo Triple-DES se denomina así porque utiliza el algoritmo DES de manera repetida con diferentes claves para cifrar el mensaje original. Esto es factible debido a que el algoritmo DES no posee una estructura de grupo. El algoritmo Triple-DES se implementa

siguiendo la siguiente estructura:

Donde

M = Mensaje a encriptar.

C = Texto encriptado.

E_K = Encriptado utilizando la clave k.

D_K = Desencriptado utilizando la clave k.

La fórmula utilizada para llevar a cabo el encriptado es la siguiente:

$$C = E_{K1}\left(D_{K2}\left(D_{K1}(M)\right)\right)$$

La fórmula para desencriptar consiste en aplicar la expresión anterior en orden inverso:

$$NI = D_{K1}\left(E_{K2}\left(D_{K1}(C)\right)\right)$$

Para llevar a cabo el encriptado se aplica la codificación utilizando la llave $K1$, luego se desencripta utilizando la llave $K2$ y, finalmente, se retorna a encriptar con la llave $K1$ [70].

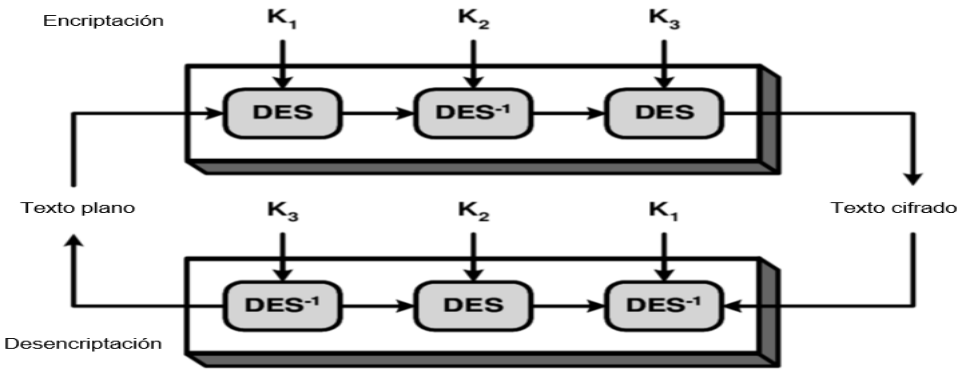


Fig. 14. Representación del algoritmo TDEA o 3DES.

Nota. Fuente: [71].

1.6.14. Rivest Cipher (RC2)

Es un algoritmo de cifrado simétrico de bloque, también conocido como ARC2, creado por Ron Rivest en 1987. Utiliza bloques de datos de 64 bits y claves de cifrado que varían de 40 a 128 bits. La clave de 40 bits es relativamente pequeña y débil, se utilizó ampliamente para fines de exportación debido a las restricciones gubernamentales.

RC2 genera una clave de encriptado y un Vector de Inicialización (IV). La clave se

compone de 12 caracteres (96 bits) y el IV de 8 caracteres (64 bits), que se combinan para formar la clave sinóptica. A pesar de su popularidad en el pasado, la seguridad de la clave de 40 bits de RC2 se considera actualmente insuficiente para proteger datos sensibles. Por lo tanto, es aconsejable utilizar claves más robustas y algoritmos de cifrado más sofisticados con el fin de asegurar la seguridad de los datos durante su transmisión [72].

1.7. Algoritmos criptográficos de clave asimétrica

1.7.1. Rivest, Shamir y Adleman (RSA)

Un algoritmo que fue implementado en las primeras técnicas de encriptación, su modo es de tipo asimétrica para los datos que se transportan por la red, el cifrado principal se hace con la clave pública y para descifrar se emplea una clave privada, si por algún motivo se pierde la clave privada es imposible verificar el contenido del archivo.

Este algoritmo se originó en el año 1997, su modo de funcionar es utilizando una clave pública relacionado con dos números primos muy altos para la encriptación de datos, se utiliza en la vida cotidiana como es el protocolo de HTTPS, correos electrónicos, mensajes que se envían mediante la red, documentos, archivos de imágenes y discos extraíbles [73].

Para llevar a cabo la ejecución de las claves, se sigue un procedimiento específico que se detalla a continuación [74].

$$C = M^e \bmod n \quad (Encriptar)$$

$$M = C^d \bmod n \quad (Desencriptar)$$

Donde.

M = Mensaje original.

C = Texto cifrado resultante (también llamado criptograma).

e = Clave pública del destinatario.

d = Clave privada.

n = Módulo público del destinatario.

mod = Operación común en la aritmética modular que devuelve el resto de la división

de los dos valores que se le proporcionan.

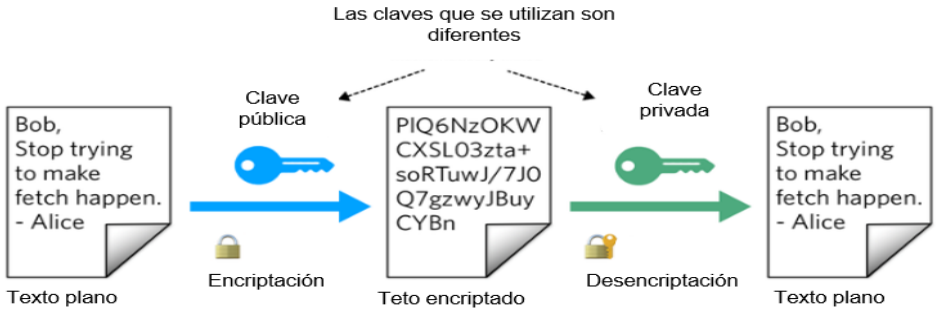


Fig. 15. Representación del algoritmo RSA.

Nota. Fuente: [75].

1.7.2. Digital Signature Algorithm (DSA)

Está enfocado en la generación de firmas digitales y fue propuesto por el Instituto Nacional de Estándares y Tecnología en 1990. Utiliza la criptografía de clave pública y se basa en la combinación de dos algoritmos: Schnorr y ElGamal. DSA está protegido por la patente estadounidense, DSA usa la noción de una función matemática única para la elaboración de la firma digital con dos valores que lo conforman de 160 bits [22].

El algoritmo DSA es completamente asimétrico, pero una desventaja que tiene frente a RSA es que requiere un mayor tiempo de cómputo con el mismo hardware. A pesar de ello, DSA se utiliza comúnmente como algoritmo de firma digital y es un estándar actualmente. Sin embargo, es importante tener en cuenta que DSA no se utiliza para cifrar datos, sino solamente como una forma de generar firmas digitales [76].

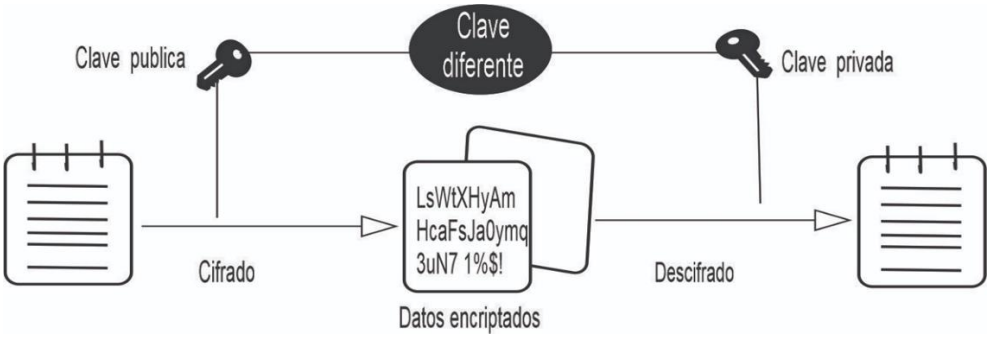


Fig. 16. Representación del algoritmo DSA.

Nota. Fuente: [77].

II. MATERIAL Y MÉTODO

2.1. Tipo y Diseño de Investigación

2.1.1. Tipo de investigación

La investigación realizada fue de tipo tecnológica aplicada, utilizando un software basado en el lenguaje de programación Python. Se aplicaron los conocimientos adquiridos previamente en la carrera y se seleccionaron algoritmos para mejorar la seguridad de la información en el envío del texto plano por internet.

2.1.2. Diseño de investigación

El diseño de investigación utilizado fue cuasiexperimental, donde no se realizaron modificaciones en los algoritmos criptográficos, se midieron los niveles de confidencialidad, integridad y disponibilidad asociados a dichos algoritmos.

Además, este estudio adopta un enfoque cuantitativo, donde los resultados de las mediciones con cada uno de los algoritmos arrojaron datos cuantitativos, posteriormente, fueron evaluados de modo que se mejore la seguridad de la información en este tipo de envíos.

2.2. Variables, Operacionalización

2.2.1. Variable Independiente

Algoritmos criptográficos

2.2.2. Variable Dependiente

Seguridad de la información en el envío de texto plano por internet

TABLA I.

OPERACIÓNALIZACIÓN DE VARIABLES

Variable de estudio	**Definición conceptual**	**Definición operacional**	**Dimensiones**	**Indicadores**	**Ítems**	**Instrumento**	**Valores finales**	**Tipo de variable**	**Escala de medición**
Variable independiente Algoritmos criptográficos	Los algoritmos se utilizan en la criptografía con el propósito de garantizar la seguridad de los datos confidenciales durante su transmisión o almacenamiento	Procesos matemáticos esenciales para cifrar y descifrar los datos, esto implica la claridad y precisión de los algoritmos y operaciones matemáticas involucradas en el proceso	Calidad del algoritmo	Rendimiento de cifrado	$R_c = \frac{Tamaño\ del\ archivo\ (bytes)}{Tiempo\ de\ encriptación\ (s)}$	Bitácora de resultados	Kb/s	Numérico	Intervalo
				Rendimiento de descifrado	$R_d = \frac{Tamaño\ del\ archivo\ (bytes)}{Tiempo\ de\ desencriptación\ (s)}$		Kb/s	Numérico	Intervalo
				Fortaleza del algoritmo	$F_a = \frac{\sum K_p}{n}$	Matriz de Kerckhoffs	%	Numérico	Checklist

Variable de estudio	Definición conceptual	Definición operacional	Dimensiones	Indicadores	Ítems	Instrumento	Valores finales	Tipo de variable	Escala de medición
Variable dependiente: seguridad de la información en el envío de texto plano por internet	La seguridad de datos en texto plano se basa en el uso de técnicas y herramientas criptográficas para garantizar la confidencialidad e integridad de la información	Medidas de gestión de claves y autenticación para asegurar que solo los usuarios autorizados puedan acceder a la información	Seguridad de información	Nivel de seguridad	$S = \frac{(NCP)}{(NCU)} * 100\%$	Bitácora de resultados Guías de observación	Se expresa como un porcentaje al multiplicar el resultado por 100%	Numérico	Intervalo
				Integridad	$I = \frac{(N_{tb})}{(N_{bnm})} * 100\%$			Numérico	Intervalo

2.3. Población de estudio, muestra, muestreo y criterios de selección

2.3.1. Población

La población seleccionada para esta investigación fueron aquellos algoritmos criptográficos empleados en el envío en texto plano por internet, y que han sido identificados en la literatura científica, los cuales son quince (15) y los que se listan en la siguiente tabla:

TABLA II

POBLACIÓN DE ESTUDIO

N°	Algoritmo	Tamaño de clave (bits)	Autor
1	TDEA - 3DES	64-bit	[16], [17]
2	MISTY1	64-bit	[78]
3	CAST-128	64-bit	[16], [56]
4	HIGHT	64-bit	[79]
5	DES	64-bit	[15], [56]
6	AES-128	128-bit	[16], [17], [56]
7	ARC2 - RC2	Hasta-128	[16]
8	SEED	128-bit	[80]
9	Camellia	128-bit	[61]
10	One-time-pad	160-bit	[18]
11	AES-256	256-bit	[15], [17]
12	Chacha20	256-bit	[17]
13	Blowfish	Hasta 448-bit	[16], [56]
14	DSA	1024-bit	[81]
15	RSA	Hasta 2048-bit	[15]
	TOTAL	15	

2.3.2. Muestra.

La muestra seleccionada para esta investigación fue seleccionada mediante muestreo no probabilístico, y fueron aquellos algoritmos criptográficos empleados en el envío en texto plano por internet que, según la literatura científica, han obtenido mejores resultados respecto a rendimiento de cifrado y rendimiento de descifrado, los cuales son tres (03) y los que se listan en la siguiente tabla:

TABLA III

MUESTRA DE ESTUDIO

N°	Algoritmo	Tamaño de clave (bits)	Autor
1	AES-256	256-bit	[15], [17]
2	Chacha20	256-bit	[17]
3	Blowfish	Hasta 448-bit	[16], [56]
	TOTAL	03	

2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad

En esta investigación se utilizó un instrumento para la recolección de datos, el cual fue el registro electrónico, se empleó una ficha de registro la cual se encuentra especificada en los anexos de esta investigación. Este instrumento fue diseñado específicamente para la ejecución del estudio y permitió registrar de manera sistemática y precisa los datos necesarios para el análisis y la interpretación de los resultados, con base en los indicadores especificados en la **¡Error! No se encuentra el origen de la referencia.**tabla I y tabla II.

Análisis documental en este sentido, se seleccionaron los artículos pertinentes para este estudio mediante búsquedas en las bases de datos científicas designadas por la universidad; Para esto, se utilizó una cadena de búsqueda y palabras clave para facilitar la selección de la información relevante. Para esta búsqueda, se selecciaron cuatro (04) bases de datos científicas específicas, a saber; Scopus, IOP Science, Science Direct y IEEE Xplore.

2.5. Procedimiento de análisis de datos

La información recolectada fue sometida a un análisis estadístico utilizando fórmulas y técnicas adecuadas para el procesamiento de los datos. De esta manera, se logró obtener resultados precisos y confiables que permitieron responder a la formulación del problema planteado.

En el caso de la variable independiente se utilizó las siguientes fórmulas estadísticas teniendo en cuenta los indicadores:

a. Rendimiento de cifrado.

La velocidad de encriptación de un algoritmo R_c está asociada con el rendimiento de cifrado, que indica la rapidez con la que el algoritmo realiza la tarea de encriptación.

$$R_c = \frac{Tamaño\ del\ archivo\ (bytes)}{Tiempo\ de\ encriptación\ (s)}$$

R_c = Rendimiento de cifrado

b. Rendimiento de descifrado.

La velocidad de desencriptación de un algoritmo (R_d) está vinculada al rendimiento de descifrado, lo cual indica la rapidez con la que el algoritmo realiza la tarea de desencriptación.

$$R_d = \frac{Tamaño\ del\ archivo\ (bytes)}{Tiempo\ de\ desencriptación\ (s)}$$

R_d = Rendimiento de descifrado

c. Fortaleza del algoritmo

La fortaleza del algoritmo F_a se relaciona con el cumplimiento que tiene el algoritmo evaluado en concordancia con los seis (06) Principios Criptográficos de Kerckhoffs. Para este caso se utilizó un "checklist" para la verificación del cumplimiento de los principios, que consiste en un conjunto de ítems o elementos a revisar, verificar o completar. Se utilizó con el objetivo de asegurarse de que todos estos principios específicos se cumplan con los 3 algoritmos como son Blowfish, AES-256 y ChaCha20.

$$F_a = \frac{\sum K_p}{n}$$

F_a = Fortaleza del algoritmo

K_p = Algoritmos que sí cumplen con los Principios de Kerckhoffs que evalúan la seguridad de la información del envío en texto plano

n = Número total de los principios de Kerckhoffs

Variable independiente

d. Nivel de seguridad

Este indicador muestra que tan seguro es la contraseña al momento de crear un usuario, así como al encriptar un documento confidencial, el nivel de seguridad aumenta a medida que mayor sea el número de claves posibles. Por lo tanto, para garantizar una alta seguridad es esencial utilizar claves seguras y robustas, en este indicador se mide tres niveles de seguridad según la clave digitada (débil, medio y fuerte).

$$S = \frac{(NCP)}{(NCU)} * 100\%$$

S = Seguridad

NCP = Número de claves posibles que se pueden generar.

NCU = Número de claves utilizadas que se han utilizado para encriptar los datos.

e. Integridad

Se expresa como un porcentaje al multiplicar el resultado por 100%. Este porcentaje indica que la integridad aumenta a medida que mayor sea el número de bits modificados (NBM) y disminuye a medida que se modifican menos bits durante la transmisión o el almacenamiento, teniendo la siguiente fórmula:

$$I = \frac{(N_{tb})}{(N_{bnm})} * 100\%$$

I = Integridad

N_{tb} = Número total de bits en los datos originales sin encriptar.

N_{bnm} = Número de bits no modificados.

2.6. Criterios éticos

Como criterio ético se tiene la protección al proceso de la investigación, de acorde a las normas establecidas se rige el Código de Ética en investigación de la Universidad Señor de Sipán S.A.C. Los investigadores en esta investigación somos responsables de los resultados que se obtengan, el tema de investigación está orientado a la actitud que se tiene que poner y el tiempo dedicado para lograr obtener un buen trabajo y cumplir con el objetivo.

Esta investigación está alineado a las tecnologías de la información y comunicación, donde se relaciona con la seguridad informática, los datos de información de la mayor parte de las personas están propensos a la inseguridad que se viene dando a diario con la conexión a internet, lo cual puede ser víctima de suplantación de información, donde el usuario muestre desconfianza en las nuevas tecnologías.

III. RESULTADOS Y DISCUSIÓN

3.1. Resultados

3.1.1. Variable Independiente

3.1.1.1. Indicador “Rendimiento de cifrado”

a. Perfil de usuarios

Con la finalidad de evaluar los indicadores de las variables de operacionalización, el usuario tiene que realizar el registro con el propósito de crear una cuenta y poder acceder al programa, los datos que se solicitan son (nombre de usuario, correo electrónico, usuario y contraseña) como se muestra en la Fig. 42, con esta información registrada se le habilita una cuenta para posteriormente ingresar con usuario y contraseña y poder usar el programa.

Después de iniciar sesión, el usuario puede guardar sus documentos confidenciales seleccionando el algoritmo de encriptación que desee, Blowfish, AES-256 ó ChaCha20, digita una contraseña para guardar su documento y así mantenerlo de manera segura Fig. 43, luego, muestra el listado de todo los documentos que fueron registrados Fig. 44 y puede desencriptar el documento encriptado usando la misma clave de encriptación Fig. 49 y así hacer legible el contenido del documento Fig. 50.

b. Evaluación de indicadores

Para evaluar el indicador de rendimiento de cifrado (R_c) se realizó la evaluación de los tres algoritmos criptográficos utilizando archivos administrativos de diferentes tamaños. En la primera prueba se utilizó un archivo de 15.4 kb, seguidamente de los archivos de 21 kb, 45,1 kb, 58.3 kb, 66.3 kb, 104 kb, 142 kb, 215 kb y por último un documento de 272 kb, teniendo un total de nueve documentos confidenciales para las pruebas respectivas.

TABLA IV

INDICADOR RENDIMIENTO DE CIFRADO DOC1 - DOC3

	Doc1				Doc2				Doc3			
Algoritmo	KB	Bytes	Tiempo Cifrado	(b/s)	KB	Bytes	Tiempo Cifrado	b/s)	KB	Bytes	Tiempo Cifrado	(b/s)
Blowfish	15.4	15,840	0,184	86,09	21	21,578	0,194	111,23	45.1	46,194	0,23	200,843478
AES-256	15.4	15,840	0,162	97,777 7778	21	21,578	0,163	132,38	45.1	46,194	0,165	279,96
ChaCha20	15.4	15,840	0,202	78,415 8416	21	21,578	0,206	104,75	45.1	46,194	0,216	213,86 1111

Nota. Elaboración propia.

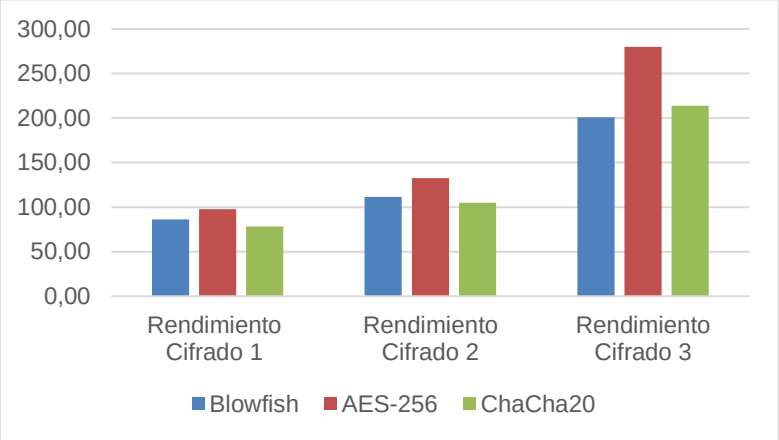


Fig. 17. Indicador rendimiento de cifrado Doc1 - Doc3.

En la primera prueba realizada se demuestra que el algoritmo AES-256 presenta mejores resultados en rendimiento de cifrado, este algoritmo al momento de hacer el proceso de encriptación muestra el tiempo en segundos más bajos en comparación con los algoritmos de encriptación Blowfish y ChaCha20. En el primer documento (Doc1) con un tamaño de 15.4 kb, el algoritmo AES-256 presenta una mayor seguridad frente al algoritmo Blowfish obteniendo una diferencia del 11,6 b/s, mientras que, en ChaCha20 con una mayor diferencia de 19,3 b/s. En el segundo documento (Doc2) con un tamaño de 21 kb, AES-256 obtuvo una diferencia de 21,1 b/s frente a Blowfish y 27,6 b/s en comparación con ChaCha20. En el tercer documento (Doc3) con un tamaño de 45.1 kb, AES-256 muestra una diferencia con blowfish de 79,1 b/s y de 66,1 b/s con ChaCha20 como se detalla en la TABLA IV.

TABLA V

INDICADOR RENDIMIENTO DE CIFRADO DOC4 - DOC6

	Doc4				Doc5				Doc6			
Algoritmo	KB	Bytes	Tiempo Cifrado	(b/s)	KB	Bytes	Tiempo Cifrado	(b/s)	KB	Bytes	Tiempo Cifrado	(b/s)
Blowfish	58.3	59,72	0,33	180,97	66.3	67,924	0,350	194,07	104	107,014	0,371	288,45
AES-256	58.3	59,72	0,175	341,26	66.3	67,924	0,186	365,18	104	107,014	0,199	537,76
ChaCha20	58.3	59,72	0,226	264,25	66.3	67,924	0,238	285,39	104	107,014	0,248	431,51

Nota. Elaboración propia.

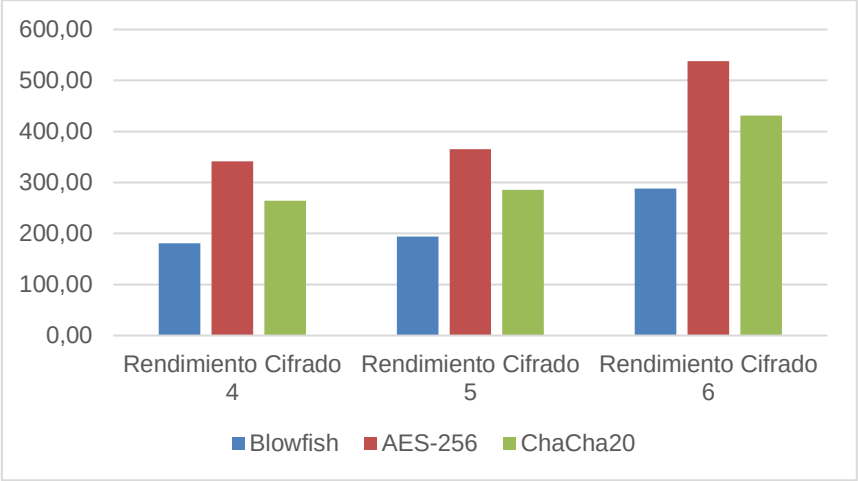


Fig. 18. Indicador rendimiento de cifrado Doc4 - Doc6.

En el cuarto documento (Doc4) con un tamaño de 58.3 kb, el algoritmo de encriptación AES-256 presenta mejores resultados en comparación con el algoritmo Blowfish obteniendo una diferencia del 160 b/s mientras que, en ChaCha20 con una diferencia de 77 b/s. En el quinto documento (Doc5) con un tamaño de 66.3 kb, AES-256 notó una diferencia de 171,1 b/s frente a Blowfish y 79,9 frente a ChaCha20. En el sexto documento (Doc6) con un tamaño de 104 kb, AES-256 muestra una diferencia con blowfish de 249 b/s y de 106,2 b/s con ChaCha20 como se detalla en la TABLA V.

TABLA VI

INDICADOR RENDIMIENTO DE CIFRADO DOC7 – DOC9

	Doc7				Doc8				Doc9			
Algoritmo	KB	Bytes	Tiempo Cifrado	(b/s)	KB	Bytes	Tiempo Cifrado	(b/s)	KB	Bytes	Tiempo Cifrado	(b/s)
Blowfish	142	145,58	0,375	388,21	215	220,561	0,386	571,40	272	279,22	0,396	705,10
AES-256	142	145,58	0,225	647,022222	215	220,561	0,237	930,64	272	279,22	0,257	1086,45914
ChaCha20	142	145,58	0,252	577,70	215	220,561	0,265	832,31	272	279,22	0,275	1015,35

Nota. Elaboración propia.

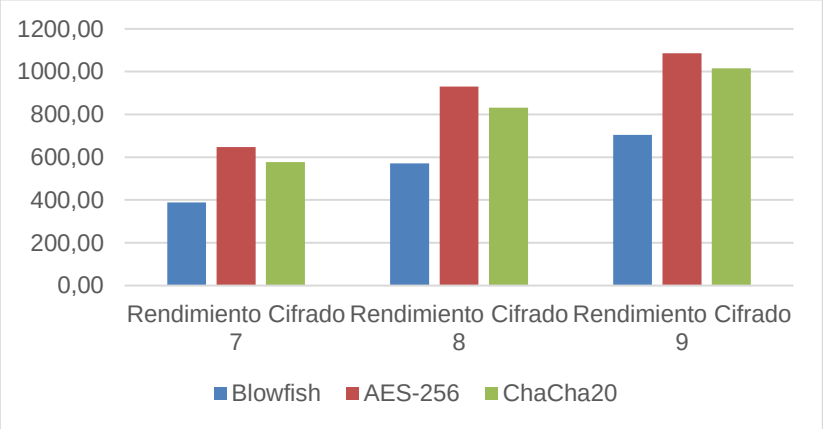


Fig. 19. Indicador rendimiento de cifrado Doc7 – Doc9.

En el séptimo documento (Doc7) con un tamaño de 142 kb, el algoritmo criptográfico AES-256 obtiene resultados más relevantes en comparación con Blowfish y ChaCha20, AES-256 con una diferencia de 258,8 b/s en comparación con Blowfish y con ChaCha20 con una diferencia de 69,3 b/s. En el octavo documento (Doc8) con un tamaño de 215 kb, AES-256 marcó una diferencia de 359,2 b/s frente a Blowfish y 98,3 frente a ChaCha20. En el noveno documento (Doc9) con un tamaño de 272 kb, AES-256 presentó una diferencia con blowfish de 381,3 b/s y de 71,1 b/s con ChaCha20 como se detalla en la TABLA VI.

El uso del algoritmo AES-256 proporciona una sólida base para proteger datos confidenciales, se considera una de las alternativas más seguras y de confianza dentro del

campo de la criptografía. Este algoritmo criptográfico ha sido aprobado por instituciones de estándares criptográficos de renombre, como el Instituto Nacional de Estándares y Tecnología (NIST) de Estados Unidos, debido a la alta seguridad que brinda. El tamaño de clave de 256 bits en AES-256 lo hace extremadamente difícil de descifrar mediante el intento de probar todas las combinaciones posibles de claves.

3.1.1.2. Indicador "Rendimiento de descifrado"

Para evaluar el indicador de rendimiento de descifrado (R_d) se realizó la evaluación de los tres algoritmos criptográficos, para este proceso se tuvo en cuenta los siguientes documentos: En la primera prueba se utilizó un archivo de 15.4 kb, seguidamente de los archivos de 21 kb, 45,1 kb, 58.3 kb, 66.3 kb, 104 kb, 142 kb, 215 kb y por último un documento de 272 kb.

TABLA VII

INDICADOR RENDIMIENTO DE DESCIFRADO DOC1 – DOC3

	Doc1				Doc2				Doc3			
Algoritmo	KB	Bytes	Tiempo Descifrado	(b/s)	KB	Bytes	Tiempo Descifrado	(b/s)	KB	Bytes	Tiempo Descifrado	(b/s)
Blowfish	15.4	15,840	0,172	92,09	21	21,578	0,176	122,60	45.1	46,194	0,185	249,70
AES-256	15.4	15,840	0,163	97,18	21	21,578	0,158	136,57	45.1	46,194	0,164	281,67
ChaCha20	15.4	15,840	0,197	80,41	21	21,578	0,2	107,89	45.1	46,194	0,21	219,97

Nota. Elaboración propia.

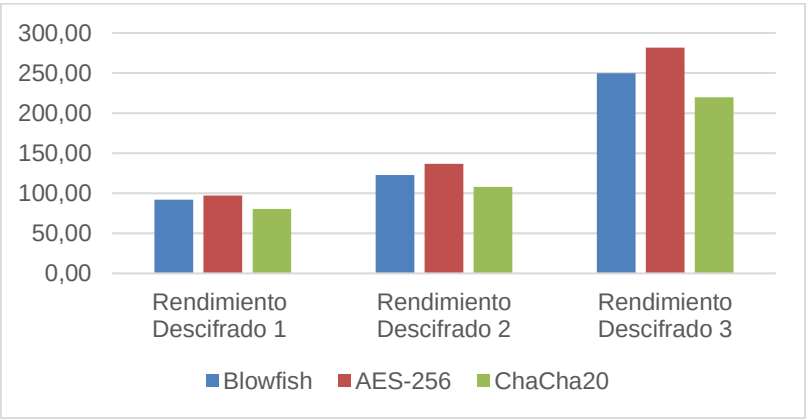


Fig. 20. Indicador rendimiento de descifrado Doc1 – Doc3.

Para el proceso de desencriptación, se evaluaron tres documentos diferentes. En el primer documento (Doc1), con un tamaño de 15.4 kb, se observó que el algoritmo AES-256 ofrece una mayor seguridad en comparación con el algoritmo Blowfish. La diferencia de rendimiento fue de 5,1 b/s frente a Blowfish y de 16,7 b/s frente a ChaCha20.En el segundo documento (Doc2), con un tamaño de 21 kb, los resultados indicaron que AES-256 superó a Blowfish con una diferencia de 13,9 b/s y a ChaCha20 con una diferencia de 28,6 b/s. En el tercer documento (Doc3), con un tamaño de 45.1 kb, AES-256 destacó nuevamente al mostrar una diferencia de rendimiento de 31,9 b/s frente a Blowfish y de 61,7 b/s frente a ChaCha20, como se detalla en la TABLA VII.

TABLA VIII

INDICADOR RENDIMIENTO DE DESCIFRADO DOC4 – DOC6

	Doc4				Doc5				Doc6			
Algoritmo	KB	Bytes	Tiempo Descifrado	(b/s)	KB	Bytes	Tiempo Descifrado	(b/s)	KB	Bytes	Tiempo Descifrado	(b/s)
Blowfish	58.3	59,720	0,222	269,01	66.3	67,924	0,232	292,78	104	107,014	0,241	444,041494
AES-256	58.3	59,720	0,171	349,24	66.3	67,924	0,181	375,27	104	107,014	0,193	554,48
ChaCha20	58.3	59,720	0,217	275,21	66.3	67,924	0,222	305,96	104	107,014	0,232	461,27

Nota. Elaboración propia.

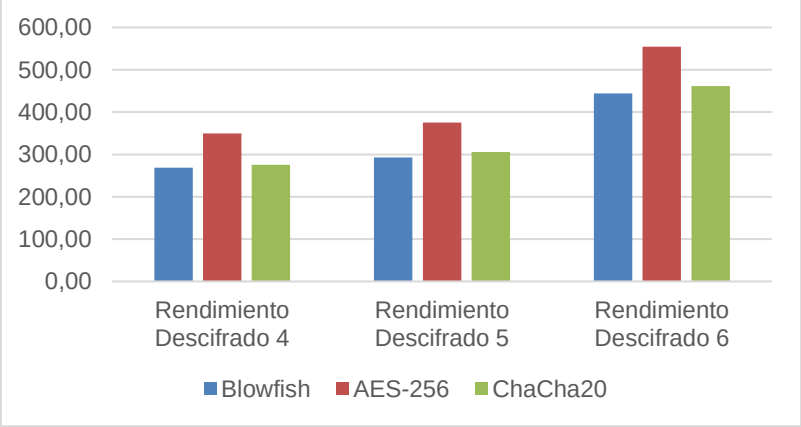


Fig. 21. Indicador rendimiento de descifrado Doc4 – Doc6.

En el análisis del cuarto documento (Doc4) con un tamaño de 58.3 kb, se evidenció que el algoritmo de encriptación AES-256 supera significativamente al algoritmo Blowfish. La diferencia de rendimiento fue de 80,2 b/s en comparación con Blowfish y de 74,0 b/s en comparación con ChaCha20. En el quinto documento (Doc5) de tamaño 66,3 kb, AES-256 también demostró su superioridad con una diferencia de 82,4 b/s frente a Blowfish y 69,3 b/s en comparación con ChaCha20. En el sexto documento (Doc6) con un tamaño de 104 kb, AES-256 continuó destacando con una diferencia de rendimiento de 110,4 b/s frente a Blowfish y 93,2 b/s frente a ChaCha20, según se detalla en la TABLA VIII.

TABLA IX

INDICADOR RENDIMIENTO DE DESCIFRADO DOC7 – DOC9

	Doc7				Doc8				Doc9			
Algoritmo	KB	Bytes	Tiempo Cifrado	(b/s)	KB	Bytes	Tiempo Cifrado	(b/s)	KB	Bytes	Tiempo Cifrado	(b/s)
Blowfish	142	145,58	0,364	399,945055	215	220,561	0,375	588,162667	272	279,22	0,383	729,033943
AES-256	142	145,58	0,217	670,875576	215	220,561	0,249	885,787149	272	279,22	0,244	1144,34426
ChaCha20	142	145,58	0,241	604,06639	215	220,561	0,253	871,782609	272	279,22	0,261	1069,80843

Nota. Elaboración propia.

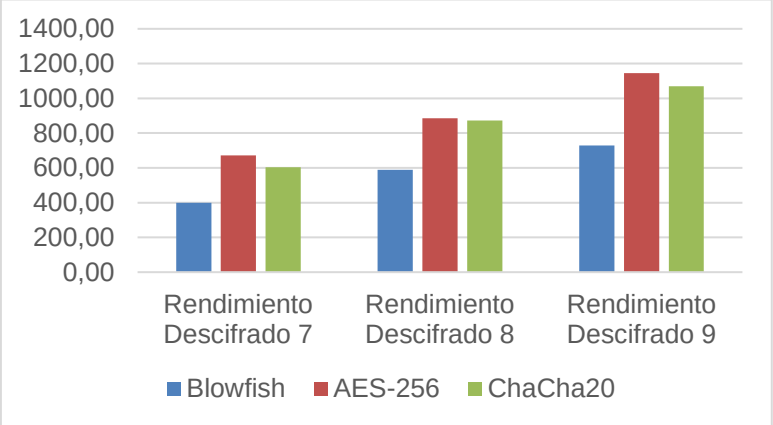


Fig. 22. Indicador rendimiento de descifrado Doc7 – Doc9.

En el séptimo documento (Doc7) de 142 kb, el algoritmo criptográfico AES-256 destacó con resultados significativamente superiores en comparación con Blowfish y ChaCha20. La diferencia de rendimiento fue de 270,9 b/s frente a Blowfish y de 66,8 b/s frente a ChaCha20. El octavo documento (Doc8), con un tamaño de 215 kb, también evidenció la superioridad de AES-256, con una diferencia de 297,6 b/s frente a Blowfish y 14,0 b/s frente a ChaCha20. En el noveno documento (Doc9) de 272 kb, AES-256 exhibió un rendimiento excepcional con una diferencia de 415,3 b/s frente a Blowfish y de 74,5 b/s frente a ChaCha20, como se detalla en la TABLA IX.

Los resultados obtenidos en el rendimiento de descifrado indican consistentemente que el algoritmo de encriptación AES-256 supera de manera significativa a Blowfish y ChaCha20 en términos de seguridad y eficiencia. A lo largo de diversos documentos con diferentes tamaños, AES-256 demostró ser la opción más destacada, evidenciando diferencias de rendimiento sustanciales. Este desempeño superior respalda la preferencia y la confianza en AES-256 como una elección robusta para la protección de datos confidenciales en comparación con los algoritmos criptográficos Blowfish y ChaCha20.

3.1.1.1. Indicador “Fortaleza del algoritmo”

La fortaleza del algoritmo está relacionada con el cumplimiento que tiene el algoritmo evaluado en concordancia con los seis (06) Principios Criptográficos de Kerckhoffs, se realiza el respectivo análisis a los algoritmos de encriptación Blowfish, AES-256 y Chacha20.

TABLA X

PRINCIPIOS CRIPTOGRÁFICOS DE KERCKHOFFS

N°	Principio	Blowfish	AES-256	ChaCha20
1	Si el sistema no es teóricamente irrompible, al menos debe serlo en la práctica	X	X	X
2	La efectividad del sistema no debe depender de que su diseño permanezca en secreto.	X	X	X
3	La clave debe ser fácilmente memorizable de manera que no haya que recurrir a notas escritas	X	X	X

4	Los criptogramas deberán dar resultados alfanuméricos.		X	
5	El sistema debe ser operable por una única persona	X	X	X
6	El sistema debe ser fácil de utilizar	X	X	X
	Ponderado de cumplimiento	5	6	5
	F_a	83.3%	100%	83.3%

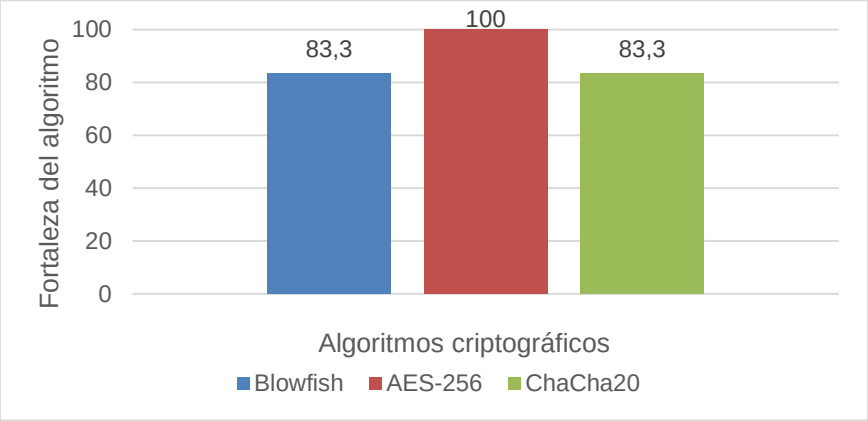


Fig. 23. Indicador fortaleza del algoritmo.

Teniendo en cuenta los principios de Kerckhoffs en la implementación del programa criptográfico, esto aporta una serie de beneficios y garantías importantes. Al aplicar los principios de Kerckhoffs se obtiene lo siguiente:

Para verificar si el programa resiste a posibles ataques, para ello se hace uso de claves largas y robustas como se ejecutó anteriormente, el sistema se vuelve más resistente frente a ataques. De los tres algoritmos implementados AES-256 garantizan el 100% de en cumplimiento a los principios de Kerckhoffs, mientras que, Blowfish y ChaCha20 no cumple con el cuarto principio criptográfico que es mostrar los datos alfanuméricos, en este caso se muestran resultado binarios y hexadecimales.

En el segundo principio de Kerckhoffs se cumple con los tres algoritmos implementados, en cuanto al diseño del sistema se garantiza el 100% de efectividad.

Al garantizar que la seguridad del sistema pertenezca únicamente de la clave utilizada

para la encriptación, se tiene una mayor fortaleza en la protección de los datos. Sin la clave correcta, sería extremadamente difícil o prácticamente imposible desencriptar los datos encriptados, en este caso se utilizó una clave de 37 dígitos para le ejecución de los tres algoritmos Blowfish, AES-256 y ChaCha20, lo cual brinda una seguridad del 100%.

Para verificar los resultados de los tres algoritmos implementados (Blowfish, AES-256 y Chacha20), se determinó que ChaCha20 como AES-256 garantizan un criptograma alfanumérico del 100%. Sin embargo, Blowfish no cumple con el cuarto principio de Kerckhoffs.

En el quinto principio de Kerckhoffs se cumple con los tres algoritmos implementados, en cuanto a la operatividad del sistema siendo usado por una púnica persona garantizando el 100% de operatividad.

Otro principio que se refleja en la implementación de los algoritmos criptográficos es que el sistema sea fácil de usar y administrar, se promueve su adopción y uso correcto. Esto evita errores por parte de los usuarios y reduce la probabilidad de vulnerabilidades causadas por una mala configuración o uso incorrecto del sistema, con la implementación de los algoritmos criptográficos se garantiza el 100% de facilidad de uso, el sistema que se muestra es sencillo y entendible al momento de usar.

TABLA XI

CUMPLIMIENTO DE LOS PRINCIPIOS DE KERCKHOFFS

Principio	AES-256	ChaCha20	Blowfish
Cumplimiento de los principios de kerckhoffs en la implementación de los tres algoritmos criptográficos	100%	83.3%	83.3%

Tras llevar a cabo la evaluación de los principios de Kerckhoffs, se presenta una tabla resumen que ilustra el grado de cumplimiento en la implementación de los tres algoritmos

criptográficos. Se observa que AES-256 alcanza un cumplimiento del 100% en todos los principios de Kerckhoffs, destacando su integridad en su totalidad, mientras que, Blowfish como ChaCha20 muestran un cumplimiento del 83.3%, indicando cierta variabilidad en su adhesión a estos principios.

3.1.2. Variable dependiente

3.1.2.1. Indicador “Nivel de seguridad”

En este nivel de seguridad, se establece que la contraseña debe estar compuesta por un mínimo de cinco caracteres. Esta especificación busca reforzar la robustez de las contraseñas, asegurando que sean lo suficientemente extensas para resistir posibles intentos de acceso no autorizado. Un requisito de longitud mínima contribuye a aumentar la complejidad y la fortaleza de las contraseñas, haciendo más difícil su descifrado mediante ataques de fuerza bruta o técnicas similares. Para evaluar este indicador se mide la cantidad de caracteres de la contraseña y se clasifica en tres niveles (Débil, Medio y Alto).

Nivel Débil:

En este nivel la contraseña está constituida por cinco caracteres como mínimo según se menciona en el siguiente estudio [82].

a. Secuencias numéricas o alfabéticas: Contraseñas que consisten en secuencias obvias, como "qwerty" o "987654321", son débiles y fáciles de descifrar.

b. Información personal fácil de obtener: Contraseñas que incluyen información personal fácilmente accesible, como el nombre del usuario, la fecha de nacimiento o el nombre de la empresa, son vulnerables.

c. Contraseñas comunes: Contraseñas que se utilizan con frecuencia, como "password", "admin", "123456", son débiles porque son las primeras que probarían los atacantes.

d. Palabras del diccionario: Contraseñas que son palabras comunes en el diccionario

son susceptibles a ataques de diccionario. Por ejemplo, "house", "sunshine" o "football".

e. Falta de combinación de caracteres: Contraseñas que solo contienen letras o solo números son más débiles que aquellas que combinan letras, números y caracteres especiales.

f. Patrones simples en el teclado: Contraseñas que siguen patrones simples en el teclado, como "asdfgh" o "zxcvbn", son predecibles y, por lo tanto, débiles.

g. Contraseñas no actualizadas: Contraseñas que no se cambian regularmente son más propensas a ser descubiertas con el tiempo.

h. Contraseñas cortas: Contraseñas que son demasiado cortas, como "12345" o "abcde", son fáciles de adivinar mediante métodos de fuerza bruta.

Nivel Medio:

Este nivel consta con un minino de seis caracteres y tener en cuenta los siguientes requisitos, letras minúsculas, mayúsculas, números caracteres especiales como se muestra en el siguiente estudio [83].

a. Combinación de caracteres: Una contraseña que combina letras tanto en mayúsculas como en minúsculas, números y caracteres especiales, como "P@ssw0rd".

b. Longitud adecuada: Una contraseña que tiene una longitud razonable, por ejemplo, de al menos 10 caracteres, como “Secure123t".

c. No utiliza información personal fácilmente accesible: La contraseña evita el uso de información personal fácilmente accesible, como nombres propios, fechas de nacimiento o palabras comunes.

d. No sigue patrones obvios en el teclado: Evita patrones sencillos en el teclado, como "asdf" o "12345".

e. No contiene palabras del diccionario: Evita el uso de palabras comunes del diccionario y utiliza combinaciones de letras que no forman palabras conocidas.

f. Se actualiza regularmente: La contraseña se cambia periódicamente para mantener la seguridad a lo largo del tiempo.

Nivel Alto:

La contraseña tiene que estar conformada de seis a más caracteres, donde debe contener letras mayúsculas y minúsculas, números, caracteres como se especifica en la siguiente investigación [84].

a. Debe tener por lo menos 12 caracteres.

b. Se permiten cinco intentos antes de imponer un bloqueo temporal de 10 minutos.

c. Se deben incluir letras mayúsculas y minúsculas (a-z y A-Z).

d. Se debe incluir un número (del 0 al 9).

e. Se debe incluir un carácter especial (!, @, #,%, etc.).

f. No puede incluir la palabra "Zendesk".

g. No puede parecerse a una dirección de correo electrónico.

h. Debe verificarse en una lista de contraseñas vulneradas conocidas para garantizar que no sea una de ellas.

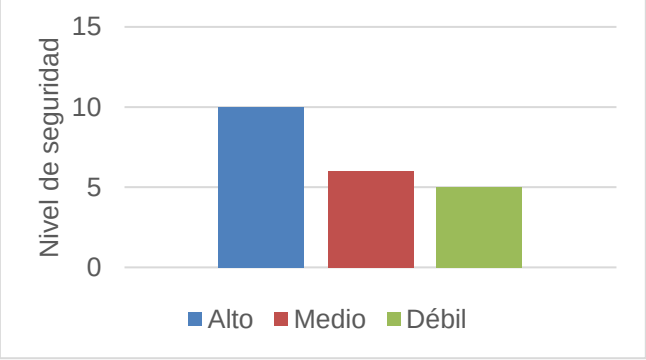


Fig. 24. Nivel de seguridad.

3.1.2.2. Indicador “Integridad”

La integridad es esencial en la criptografía, sin ella, un atacante podría modificar los datos en tránsito o en reposo, lo que podría comprometer la confidencialidad y la autenticidad de la información. Para ello, se realizó diferentes evaluaciones para comprobar la integridad en los tres algoritmos criptográficos implementados, donde N_{bnm} demuestra el número de bits no modificados y N_{tb} el número total de bits en los datos originales sin encriptar, Para comprobar la integridad se realizó el estudio de nueve documentos empresariales de diferentes tamaños que se detalló en el proceso de encriptación.

TABLA XII

INDICADOR INTEGRIDAD DOC1 – DOC3

Algoritmo	Doc1			Doc2			Doc3		
	Archivo original (B)	Archivo desencriptado (B)	Integridad (%)	Archivo original (B)	Archivo desencriptado (B)	Integridad (%)	Archivo original (B)	Archivo desencriptado (B)	Integridad (%)
Blowfish	15,840	15,840	100%	21,578	21,576	99,99%	46,194	46,192	99,99%
AES-256	15,840	15,840	100%	21,578	21,578	100%	46,194	46,194	100%
ChaCha20	15,840	15,840	100%	21,578	21,578	100%	46.194	46,194	100%

Nota. Elaboración propia.

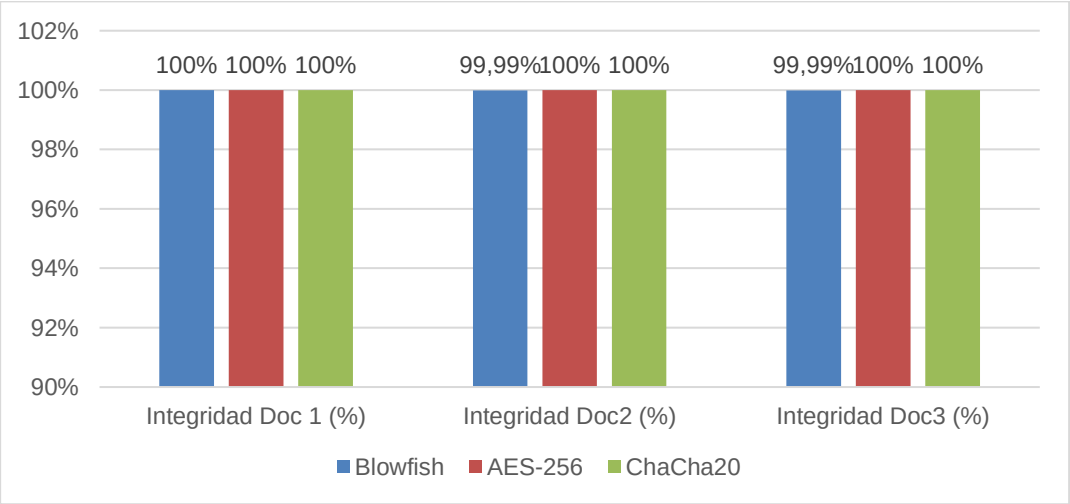


Fig. 25. Indicador de integridad Doc1 – Doc3.

La integridad de los datos es crucial para asegurar que no sean modificados de manera no autorizada, accidental o malintencionada, garantizando así la consistencia y exactitud de la información. En la primera evaluación (Doc1), se utilizó un archivo de 13,480 Bytes para probar tres algoritmos criptográficos: blowfish, AES-256 y ChaCha20. En este análisis, se observó que los tres algoritmos demostraron una integridad del 100%. En la segunda evaluación, se empleó un archivo de 25,578 Bytes. En este caso, tanto AES-256 como ChaCha20 conservaron una integridad del 100%, mientras que blowfish registró un 99,99%. En la tercera evaluación (Doc3) con un archivo de 46,194 Bytes, se obtuvieron resultados similares al estudio anterior. AES-256 y ChaCha20 exhibieron una integridad del 100%, mientras que blowfish alcanzó un 99,99% como se detalla en la TABLA XII.

TABLA XIII

INDICADOR INTEGRIDAD DOC4 – DOC6

Algoritmo	Doc4			Doc5			Doc6		
	Archivo original (B)	Archivo desencriptado (B)	Integridad (%)	Archivo original (B)	Archivo desencriptado (B)	Integridad (%)	Archivo original (B)	Archivo desencriptado (B)	Integridad (%)
Blowfish	59,720	59,720	100%	67,924	67,920	99,99%	107,014	107,008	99,99%
AES-256	59,720	59,720	100%	67,924	67,924	100%	107,014	107,014	100%
ChaCha20	59,720	59,720	100%	67,924	67,924	100%	107,014	107,014	100%

Nota. Elaboración propia.

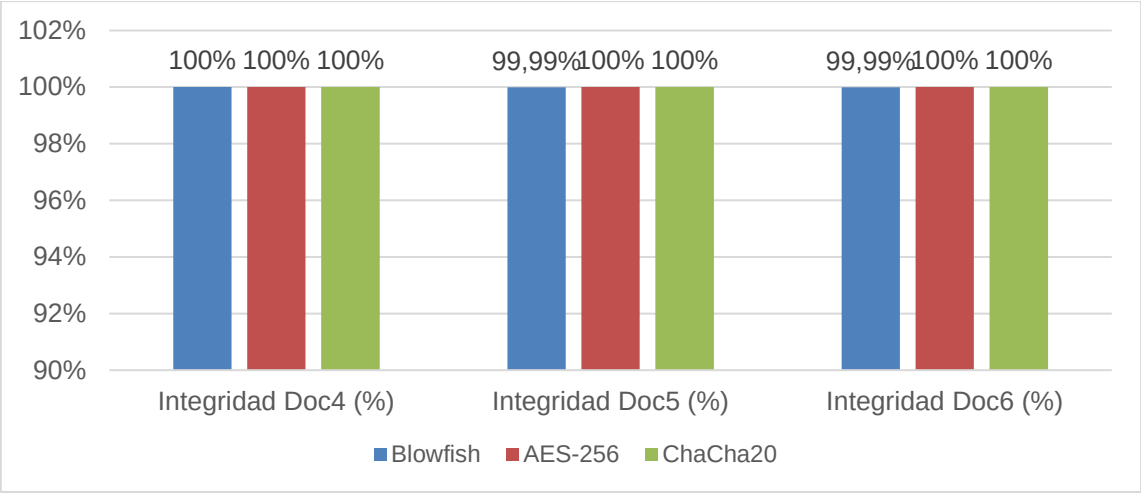


Fig. 26. Indicador de integridad Doc4 – Doc6.

En la cuarta evaluación (Doc4), se procedió a analizar un archivo de mayor tamaño, concretamente de 59,720 Bytes. En esta instancia, se constató que los tres algoritmos criptográficos, Blowfish, AES-256 y ChaCha20, demostraron una integridad impecable del 100%, destacando su robustez incluso frente a conjuntos de datos más extensos. En la quinta evaluación (Doc5), el enfoque se centró en un archivo de 67,924 Bytes, en este contexto, tanto AES-256 como ChaCha20 mantuvieron un nivel de integridad intachable del 100%, subrayando su capacidad para preservar la integridad de los datos en condiciones diversas. Por su parte, Blowfish, aunque continuó exhibiendo un rendimiento muy alto con un 99,99%, evidenció una ligera variación en comparación con los resultados anteriores. En la sexta evaluación (Doc6), se extendió a un archivo de dimensiones considerables, alcanzando los 107,014 Bytes. Los resultados obtenidos fueron congruentes con las observaciones previas, donde AES-256 y ChaCha20 mantuvieron una integridad del 100%, consolidando su confiabilidad en diferentes contextos. Por otro lado, Blowfish, aunque destacó con un 99,99%, persistió con una mínima variabilidad según lo detallado en la TABLA XIII, reafirmando su sólido desempeño en escenarios de evaluación más amplios.

TABLA XIV

INDICADOR INTEGRIDAD DOC7 – DOC9

Algoritmo	Doc7			Doc8			Doc9		
	Archivo original (B)	Archivo desencriptado (B)	Integridad (%)	Archivo original (B)	Archivo desencriptado (B)	Integridad (%)	Archivo original (B)	Archivo desencriptado (B)	Integridad (%)
Blowfish	145,580	145,576	99,99%	220,561	220,560	99,99%	279,220	279,216	99,99%
AES-256	145,580	145,580	100%	220,561	220,561	100%	279,220	279,220	100%
ChaCha20	145,580	145,580	100%	220,561	220,561	100%	279,220	279,220	100%

Nota. Elaboración propia.

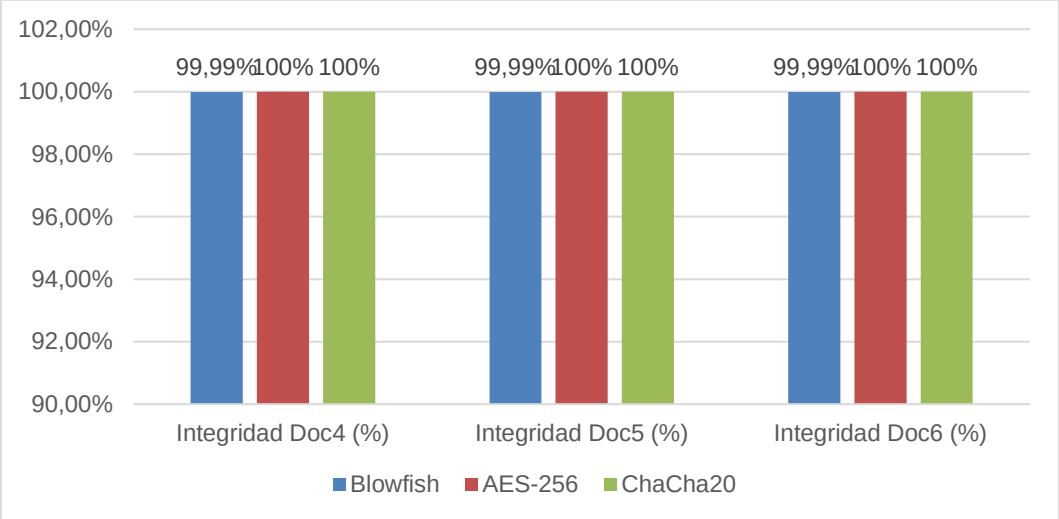


Fig. 27. Indicador de integridad Doc7 – Doc9.

En el marco de la séptima evaluación (Doc7), se llevó a cabo un análisis exhaustivo de un archivo sustancialmente más grande, con una dimensión específica de 145,580 Bytes. En esta coyuntura, se verificó con certeza que los dos algoritmos criptográficos principales, AES-256 y ChaCha20, demostraron una integridad impecable del 100%, enfatizando su robustez incluso cuando se enfrentan a conjuntos de datos de mayor envergadura. Al mismo tiempo, Blowfish, aunque mostró un sólido desempeño con un 99,99%, evidenció su capacidad para mantener la seguridad de los datos. En la octava evaluación (Doc8), se hizo

especial hincapié en un archivo de dimensiones aún mayores, alcanzando un total de 220,561 Bytes. En este escenario, tanto AES-256 como ChaCha20 mantuvieron un nivel de integridad intachable del 100%, destacando su capacidad para preservar la integridad de los datos en condiciones diversas y desafiantes. Aunque Blowfish continuó exhibiendo un rendimiento elevado con un 99,99%, se notó una leve variación en comparación con los resultados anteriores, lo que subraya la importancia de evaluar el rendimiento de los algoritmos en diferentes contextos y condiciones. En la novena evaluación (Doc9), el análisis se amplió a un archivo de dimensiones aún más considerables, alcanzando la cifra de 279,220 Bytes. Los resultados obtenidos fueron consistentes con las observaciones previas, confirmando que tanto AES-256 y ChaCha20 mantuvieron una integridad del 100%, consolidando su confiabilidad en diversos contextos y ante conjuntos de datos significativos. Por otro lado, Blowfish, a pesar de destacar con un 99,99%, mostró una mínima variabilidad según se detalla en la TABLA XIV, reafirmando su sólido desempeño en escenarios de evaluación más amplios y desafiantes. Estos hallazgos subrayan la importancia de considerar la adaptabilidad de los algoritmos criptográficos a condiciones diversas para garantizar una seguridad efectiva de los datos.

Se concluye que, para garantizar una mayor integridad en la seguridad de los archivos, es recomendable utilizar el algoritmo de encriptación AES-256 y ChaCha20, ya que ambos han demostrado consistentemente un nivel óptimo de integridad en diversas evaluaciones. Estos resultados respaldan la elección de estos algoritmos como una medida efectiva para preservar la integridad de los datos en entornos de seguridad.

3.2. Discusión

Respecto a seleccionar los algoritmos criptográficos más relevantes que garanticen la seguridad de los datos, se seleccionaron dichos algoritmos criptográficos mediante la adopción de las directrices para la revisión sistemática que propusieron Kitchenham & Charters, buscando información de bases de datos de revistas indizadas, identificándose un total de quince algoritmos criptográficos existentes para dicho fin, de los cuales, blowfish, AES-256 y Chacha20 fueron aquellos que tenían las mejores valoraciones en cuanto a indicadores tales como tiempo de cifrado, tiempo de descifrado, rendimiento, entre otros. Estos resultados concuerdan parcialmente con los de otros estudios tales como, el realizado en Claros; Coca, Blanco y Sandoval, [17] en donde se seleccionaron a AES, 3DES y ChaCha20 como los algoritmos de mejores prestaciones en cuanto a seguridad de la información y los cuales fueron evaluados en cuanto a tiempo de ejecución, consumo de energía y el porcentaje de la Unidad de Procesamiento Gráfico (GPU), para cifrar y descifrar un conjunto de 150 archivos con distintos formatos y tamaños. También concuerda parcialmente con el estudio realizado por Kubadia; Idnani; Jain, [16] en donde se seleccionaron a AES, ARC2, Blowfish, CAST y 3DES, evaluándolos con base a aspectos como los tiempos de cifrado y descifrado, el rendimiento y el tamaño del texto cifrado en comparación con el texto plano. Estos resultados son parecidos ya que algoritmos como AES-256, Chacha20 y Blowfish a lo largo de la literatura científica han obtenido buenos resultados, por lo que los consolida como los de mejores prestaciones.

Respecto a identificar los principales ataques conocidos contra los algoritmos criptográficos AES-256, Chacha20 y Blowfish, se identificaron los principales ataques conocidos contra estos algoritmos criptográficos, siguiendo la misma línea metodológica de Kitchenham & Charters, recogiendo un total de once ataques, destacando entre ellos los ataques de fuerza bruta, los ataques Man-in-the-Middle, los ataques Meet-in-the-middle, entre otros, los cuales atentan contra la seguridad de la información vulnerando la gestión de contraseñas, la interceptación de mensajes, la filtración de información, entre otros. Estos

resultados se contraponen contra otros resultados obtenidos en otros estudios ya que muchos de ellos no han realizado un análisis de las vulnerabilidades y amenazas a las que están expuestos estos algoritmos criptográficos. Se tiene, por ejemplo, el estudio realizado por admavathi; Kumari y Ranjitha, [15] centrándose directamente a conceptualizar términos asociados y empleados en la criptografía, para luego directamente caracterizar los algoritmos DES, AES and RSA. Del mismo modo, también se compara con los resultados obtenidos por Mawengkang; Sitepu y Efendi, [18] en donde no se hace un análisis previo de los ataques que atentan contra los algoritmos criptográficos, sino que, directamente desarrollan un método de combinación Super Cifrado entre los algoritmos One Time Pad y Vigenere consiguiendo un cifrado más fuerte haciéndolo muy difícil de resolver. Se han buscado investigaciones acerca de estos autores en los que, posiblemente han realizado estudios acerca de los ataques que buscan vulnerar los algoritmos criptográficos, pero, sin embargo, no se han encontrado por lo que existe una brecha en cuanto a ello, haciendo que los investigadores en etapas tempranas de investigación queden aún poco claros en cuanto a este análisis previo.

Respecto a desarrollar en lenguaje de programación los algoritmos criptográficos para cifrar los datos de un texto plano, se desarrollaron dichos algoritmos en lenguaje de programación Phyton, empleando un entorno de desarrollo integrado PyCharm, utilizando el entorno virtual Virtualenv y considerando para ello un contexto de archivos de texto pertenecientes a documentos empresariales con extensión “.docx”, el cual permitió observar cómo variaban las mediciones de los indicadores en los algoritmos blowfish, AES-256 y ChaCha20. Estos resultados se muestran mediante la implementación de los algoritmos en Phyton.

Respecto a validar el rendimiento de cada algoritmo criptográfico implementado en el contexto establecido previamente, se validaron dichos rendimientos de manera individual para cada uno de ellos, considerando para la calidad de los indicadores tales como rendimiento de cifrado, rendimiento de descifrado y fortaleza del algoritmo, mientras que por el lado de la

seguridad de la información indicadores tales como la confidencialidad y la integridad, reflejando que AES-256 es el mejor en dichos aspectos obteniendo, por ejemplo, un 100% de cumplimiento en cuanto a los seis Principios Criptográficos de Kerckhoffs, en comparación de los otros dos algoritmos. Estos resultados son parecidos a los estudios obtenidos realizados de Padmavathi; Kumari y Ranjitha donde realizaron una evaluación del desempeño de los algoritmos criptográficos; en la siguiente investigación de Kubadia; Idnani y Jain también realizaron una evaluación de algoritmos, finalmente, Claros, Centellas; Coca y Alcocer en un estudio comparativo de los algoritmos simétricos [15], [16], [17] en donde los autores validaron el rendimiento de cada algoritmo criptográfico implementado. Cada uno de los algoritmos que fueron evaluados tienen fortalezas, por ejemplo, en el primer estudio, ChaCha20, tuvo un consumo de CPU del 10-15% al momento de compararlo con otros algoritmos como AES, 3DES los cuales tuvieron un consumo del 15-30%. Por una parte, en el segundo estudio, AES presentó un tiempo de encriptado de 2,0 segundos para paquetes de 868 (KB) mientras que, los algoritmos DES y RSA demostraron requerir más tiempo para el cifrado, mientras que, para el descifrado AES presentó un tiempo de 1,2 segundos, siendo más efectivo que DES y RSA en la protección de la información durante la transmisión. Finalmente, en el tercer estudio, los algoritmos AES, ARC2, Blowfish, CAST y 3DES se cifraron con un tamaño de 8 MB, y se observó que el algoritmo Blowfish presentó un tiempo de cifrado de 0,137474 segundos, mientras que los demás algoritmos tuvieron tiempos de cifrado más largos, asimismo, para el descifrado, Blowfish también fue el más eficiente, con un tiempo de 0,137718 segundos. Los resultados demostraron que el algoritmo Blowfish es el más efectivo en comparación con los otros algoritmos evaluados.

3.3. Aporte de la investigación

Objetivo 01. Seleccionar los algoritmos criptográficos más relevantes que garanticen la seguridad de los datos

Con el propósito de seleccionar los algoritmos criptográficos más relevantes para asegurar la protección de los datos, de acuerdo a la literatura científica se aplicaron las fases

propuestas por Kitchenham & Charters [82]. Según los autores mencionados, las directrices propuestas tienen como objetivo principal conseguir un enfoque general del área de investigación y complementar mediante la revisión del estado actual de las evidencias científicas en temas específicos, como se busca lograr en este informe. En este contexto, los resultados obtenidos en esta revisión permitieron identificar y mapear los algoritmos criptográficos utilizados para salvaguardar la información transmitida en texto plano a través de internet, así como comprender su grado de implementación. Además, se examinaron los resultados que fueron obtenidos por diferentes autores en este campo.

A continuación, se procede con la ejecución del proceso de revisión sistemática, el cual se presenta en la figura siguiente:

Método de Kitchenham & Charters	Planificar la revisión sistemática
	Realizar la revisión sistemática
	Reportar la revisión sistemática

Fig. 28. Método de Sistemático de Revisión de Literatura.

Nota. Fuente: [82].

Una vez establecido el enfoque de la revisión de la literatura que tiene relación con los algoritmos criptográficos utilizados para garantizar la seguridad de la información en el envío del texto plano por Internet, se procedió a su implementación. En este sentido, se seleccionó los artículos pertinentes para este estudio mediante búsquedas en las bases de datos científicas designadas por la universidad; Para esto, se utilizó una cadena de búsqueda y palabras clave para facilitar la selección de la información relevante. Para esta búsqueda, se seleccionó cuatro (04) bases de datos científicas específicas, a saber; Scopus, IOP Science, Science Direct y IEEE Xplore.

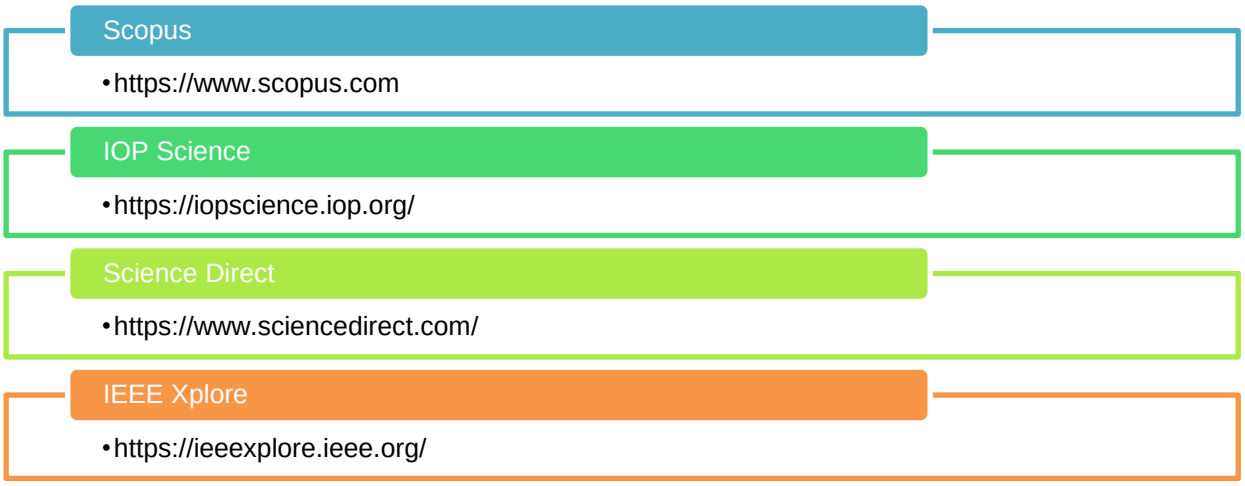


Fig. 29. Bases de datos científicas empleadas.

Al momento de seleccionar las cuatro bases de datos, se realizó con la intención de enfocarse exclusivamente artículos que han sido revisados por pares y que son publicados en revistas, conferencias, talleres, libros o simposios reconocidos. Para realizar una búsqueda efectiva en estas bases de datos, se empleó la siguiente cadena de búsqueda:

("evaluation of cryptographic algorithms" OR "comparison of cryptographic algorithms")

Fig. 30. Cadena de búsqueda para la revisión de artículos.

La elección de esta cadena de búsqueda se fundamenta en pruebas iniciales donde se exploraron temas que tengan relación con algoritmos criptográficos utilizados para asegurar la confidencialidad de la seguridad de la información en el envío del texto plano por Internet. Es fundamental destacar que para esta búsqueda bibliográfica se llevó a cabo dentro de un marco de tiempo limitado, abarcando la literatura publicada desde 2018 hasta 2023. Una vez obtenido los artículos de las bases de datos utilizando la cadena de búsqueda mencionada, se procedió a realizar un análisis para determinar su relevancia.

En una primera fase, se evaluó la relevancia de los artículos teniendo en cuenta los títulos. De esta manera, se descartaron aquellos artículos que según el título indicaba con claridad que no eran pertinentes para este estudio. Algunos de los artículos recuperados a

través del protocolo de búsqueda no guardaban relación con los algoritmos criptográficos utilizados para asegurar la seguridad de la información en el envío del texto plano por Internet. En cambio, abordaban temas como algoritmos criptográficos para audio, imagen o video, entre otros. En el caso donde la relevancia no se determinaba con facilidad mediante el título, los artículos pasaban a la siguiente fase para continuar con su posterior selección.

En una segunda fase, se realizó el análisis de los resúmenes de los artículos que pasaron la primera etapa. Durante este análisis, se aplicaron criterios de exclusión con el objetivo de descartar lo siguiente: (1) artículos que no han sido sometidos a revisión por pares, como entrevistas, anuncios de prensa o artículos de opinión; (2) artículos que no estaban disponibles en su versión completa; (3) artículos donde el enfoque principal no estaba relacionado con los algoritmos criptográficos para asegurar la seguridad de la información; (4) artículos científicos duplicados; (5) trabajos publicados y escritos en un idioma diferente al inglés; (6) y, por último, artículos retractados. Donde, los artículos que cumplieron estos criterios de exclusión se consideró que se centraban en los algoritmos criptográficos que se utilizan para garantizar la seguridad de la información del envío en texto plano por internet fueron seleccionados para su análisis posterior.

En tercera fase, se procedió con la categorización de los artículos de investigación relevantes en la literatura. Para realizar esta categorización, se siguió el enfoque propuesto por Kitchenham y Charters, tal como se muestra en la figura 82. Este enfoque consiste en extraer palabras clave y conceptos de los resúmenes de los artículos que reflejan las contribuciones relacionadas con los algoritmos criptográficos utilizados para garantizar la seguridad de la información en la transmisión de texto plano por Internet [82]. Además, se recopiló información de los artículos para su posterior análisis, con el propósito de abordar las preguntas de investigación planteadas. Se consideró un total de 7 (siete) elementos principales de cada artículo, que se detallan en la tabla siguiente.

TABLA XV

PRINCIPALES ELEMENTOS EXTRAÍDOS DE LOS ARTÍCULOS CIENTÍFICOS

N°	Elementos de datos	Detalle
1	Autor	Autor(es) del artículo.
2	Año	Año de la publicación del artículo.
3	Título	Nombre de la publicación.
4	País	País de la publicación.
5	Fuente	Revista, Conferencias, Simposios.
6	Base de Datos	Base de Datos donde se encontró el artículo.
7	Procedencia	Universidad, institución, etcétera.

Nota: Elaboración propia.

Los siete elementos permitieron obtener información relevante de cada artículo seleccionado, incluyendo el nombre del autor(es), año de publicación, título, país de origen, fuente de publicación, base de datos a la que pertenece y procedencia del artículo. Después de obtener estos elementos, se asignó un número del 1 al 33 a cada artículo seleccionado como identificador único. La información extraída se recopiló en una hoja de Excel donde permite organizar y analizar la información con facilidad.

Haciendo uso del protocolo de búsqueda, se recuperaron un total de 3581 artículos de las cuatro bases de datos científicas. Después de la primera etapa de selección, que se basó en los títulos de los artículos, se excluyeron 446 artículos con fechas anteriores a 2018, lo que resultó en 3135 artículos para una nueva evaluación. En esta fase se excluyeron aquellos artículos que no estaban relacionados con algoritmos criptográficos existentes para la seguridad de la información del envío en texto plano por internet. Al finalizar el proceso de selección, se eligieron 33 artículos para su inclusión en el estudio, tal como se muestra en la figura siguiente:

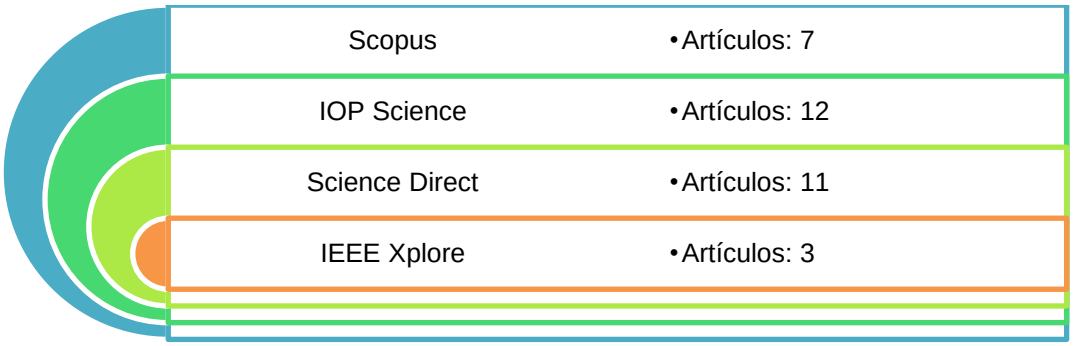


Fig. 31. Selección de artículos por base de datos científica.

Tras analizar la información extraída de los 33 artículos científicos, se identificaron un total de quince (15) algoritmos criptográficos existentes para garantizar seguridad de la información del envío en texto plano por internet. Estos algoritmos fueron seleccionados y se encuentran detallados en la siguiente tabla:

TABLA XVI
ALGORITMOS CRIPTOGRÁFICOS IDENTIFICADOS DE LOS ARTÍCULOS CIENTÍFICOS

N°	Algoritmo	Longitud de clave (bits)	Tipo de cifrado	Cifrado de flujo o bloque	Rondas	Autor
1	TDEA - 3DES	64-bit	Simétrica	Bloque	48	[16], [17]
2	MISTY1	64-bit	Simétrica	Bloque	4	[78]
3	CAST-128	64-bit	Simétrica	Bloque	12	[16], [56]
4	HIGHT	64-bit	Simétrica	Bloque	32	[79]
5	DES	64-bit	Simétrica	Bloque	16	[15], [56]
6	AES-128	128-bit	Simétrica	Bloque	10	[16], [17], [56]
7	ARC2 ó RC2	Hasta-128	Simétrica	Flujo	1 a 255	[16]
8	SEED	128-bit	Simétrica	Bloque	16	[80]
9	Camellia	128-bit	Simétrica	Bloque	18	[61]
10	One-time-pad	160-bit	Simétrica	Flujo	-	[18]
11	AES-256	256-bit	Simétrica	Bloque	14	[15], [17]
12	Chacha20	256-bit	Simétrica	Flujo	20	[17]
13	Blowfish	Hasta 448-bit	Simétrica	Bloque	16 a 18	[16], [56]
14	DSA	1024-bit	Asimétrica	-	-	[81]
15	RSA	Hasta 2048-bit	Asimétrica	-	-	[15]
	TOTAL	15				

Posteriormente a la identificación de los algoritmos criptográficos existentes para la seguridad de la información del envío en texto plano, se buscó seleccionar los de mayor relevancia para ser posteriormente evaluados. Para ello se consideró el análisis de tres (3) artículos científicos donde se realiza la comparación de los algoritmos criptográficos en tiempos de encriptado, desencriptado, y el rendimiento.

En la primera comparación se realizó el estudio de los algoritmos criptográficos AES, ARC2, Blowfish, CAST y DES3, y las métricas de rendimiento para la base del estudio son el tiempo de cifrado, tiempo de descifrado, rendimiento, tamaño del texto sin formato frente al tamaño del texto cifrado [16].

TABLA XVII

COMPARACIÓN DE ALGORITMOS AES, ARC2, BLOWFISH, CAST Y 3DES

Tamaño (MB)	Algoritmo	T-Cifrado	T-Descifrado
64	AES	2.659.563	259.036
64	ARC2	19.852.633	19.577.693
64	Blowfish	1.160.837	1.080.431
64	CAST	9.126.131	9.135.119
64	3DES	3.981.097	3.999.742

Nota: Elaboración propia.

Para respaldar la afirmación de que el algoritmo Blowfish exhibe un rendimiento superior en comparación con AES, ARC2, CAST y 3DES en cuanto a los tiempos de cifrado y descifrado, se llevó a cabo una evaluación exhaustiva. Esta prueba se ejecutó utilizando un mensaje de tamaño considerable, específicamente, con un volumen de datos de 64 MB.

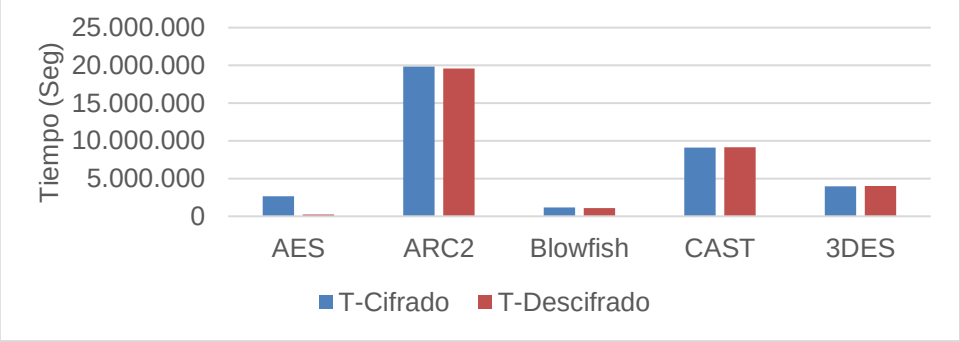


Fig. 32. Comparación en tiempos de cifrado y descifrados de algoritmos AES, ARC2, Blowfish, CAST y 3DES.

En cuanto a rendimiento se demostró que el algoritmo Blowfish tiene mejores resultados en comparación con AES, ARC2, CAST y 3DES, la prueba se realizó en mensajes de diferentes tamaños desde 1 MB, 2 MB, 4 MB, y así sucesivamente duplicando el tamaño de mensaje hasta llegar a 128 MB.

TABLA XVIII

COMPARACIÓN DE ALGORITMOS AES, ARC2, BLOWFISH, CAST Y 3DES

Tamaño (MB)	AES	ARC2	Blowfish	CAST	DES3
1	12.154.591	1.659.062	20.882.616	3.238.763	6.717.529
2	11.522.383	1.672.481	30.226.997	3.433.723	7.645.728
4	12.590.912	1.680.215	26.295.854	3.595.853	7.951.282
8	12.903.042	1.653.599	29.768.307	3.575.147	8.023.302
16	12.630.329	1.611.972	30.215.904	3.571.585	8.133.766
32	12.687.143	1.671.135	28.710.615	3.586.782	8.259.477
64	12.483.230	1.662.070	29.240.590	3.588.801	8.211.667
128	12.655.691	1.669.441	27.951.549	3.575.272	8.209.673

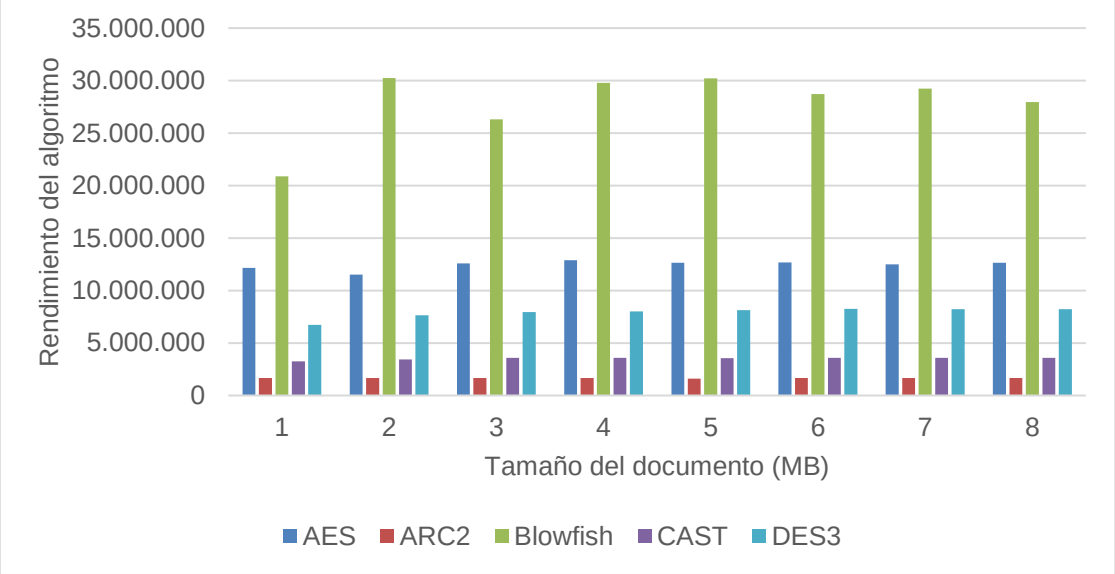


Fig. 33. Comparación en rendimiento de algoritmos criptográficos AES, ARC2, Blowfish, CAST y 3DES.

Blowfish algoritmo seleccionado.

Es importante tener en cuenta que el desempeño de los algoritmos de encriptación puede variar según el hardware y la implementación específicos. Además, la seguridad de un algoritmo no se basa únicamente en su velocidad de encriptación y desencriptación, sino en

su resistencia a ataques criptográficos y su capacidad para proteger la información sensible de manera efectiva, en este estudio se puede deducir que el algoritmo Blowfish presenta mejores resultados en tiempo para cifrar y descifrar, además mostró mejores resultados en cuanto a rendimiento junto con el algoritmo AES.

En el segundo estudio se realizó la comparación de los algoritmos criptográficos AES (256), DES (56), 3DES (128), RC2 (128) y RSA (2048) en milisegundos (ms), El estudio evaluó el tiempo de ejecución, el tamaño de memoria del archivo y el rendimiento del algoritmo para diferentes tamaños de archivos de texto experimentales [67].

TABLA XIX

COMPARACIÓN DE ALGORITMOS AES (256), DES (56), 3DES (128), RC2 (128) Y RSA (2048)

Archivo de texto	AES (256)	DES (56)	3DES (128)	RC2 (128)	RSA (2048)
915 (KB)	2050	2133	2235	2064	4915
5.384 (MB)	3606	3255	3094	3867	7805
11.804 (MB)	4882	6481	7062	7847	15440
35.350 (MB)	13371	16735	18337	21505	39107
59.809 (MB)	25038	32411	38483	39958	65527
106 (MB)	51249	65721	78041	83432	109237

Nota. Elaboración propia.

El algoritmo AES es el más rápido tanto en los procesos de cifrado como de descifrado en comparación con otros algoritmos en términos de tiempo de procesamiento. DES tuvo el segundo nivel de rendimiento después de AES porque consume menos tiempo evaluado que otros algoritmos. 3DES y RC2 consumieron aproximadamente los mismos valores de tiempo durante los procesos de cifrado y descifrado, mientras que RSA fue el más lento.

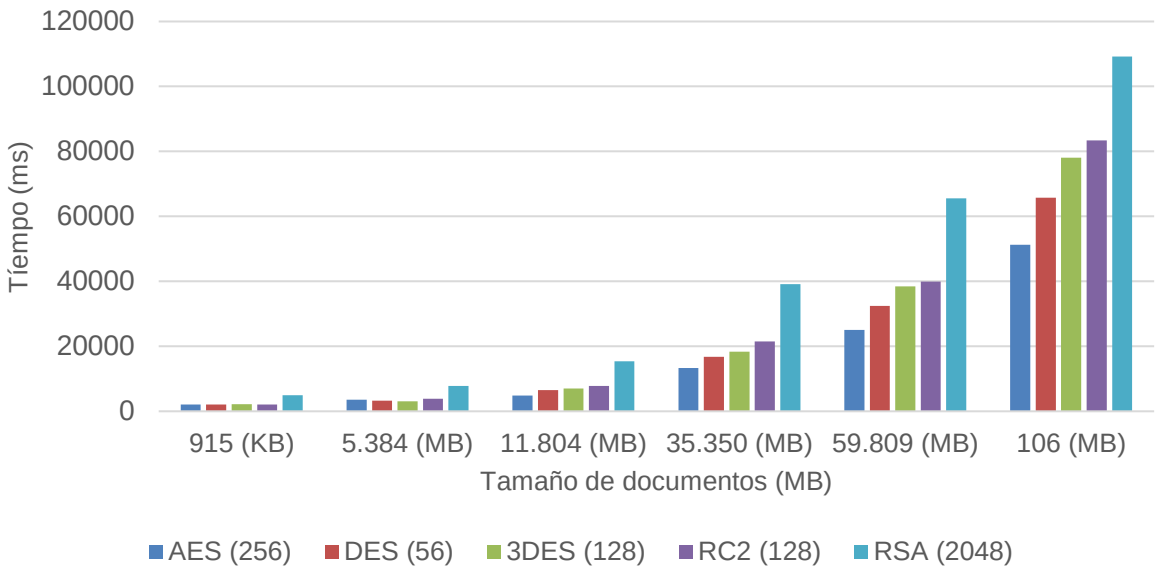


Fig. 34. Comparación en tiempo de cifrado y descifrado de algoritmos AES (256), DES (56), 3DES (128), RC2 (128) y RSA (2048).

El algoritmo AES-256 presenta los mejores resultados en cuanto al tiempo promedio total y el rendimiento en comparación con otros algoritmos, lo convierte en el algoritmo más rápido y eficiente evaluado en este estudio.

TABLA XX

COMPARACIÓN DE ALGORITMOS AES (256), DES (56), 3DES (128), RC2 (128) Y RSA (2048)

Tiempo promedio y rendimiento	AES (256)	DES (56)	3DES (128)	RC2 (128)	RSA (2048)
Tiempo promedio total	16699.33	21122.67	24542	26445.5	40338.5
Rendimiento (MByte/seg)	13,12999	10,38041	8,934154	8,291089	5,435552

Los resultados obtenidos respaldan claramente la superioridad del algoritmo AES-256 en cuanto a su velocidad y rendimiento en los procesos de encriptado y desencriptado, en comparación con los demás algoritmos evaluados.

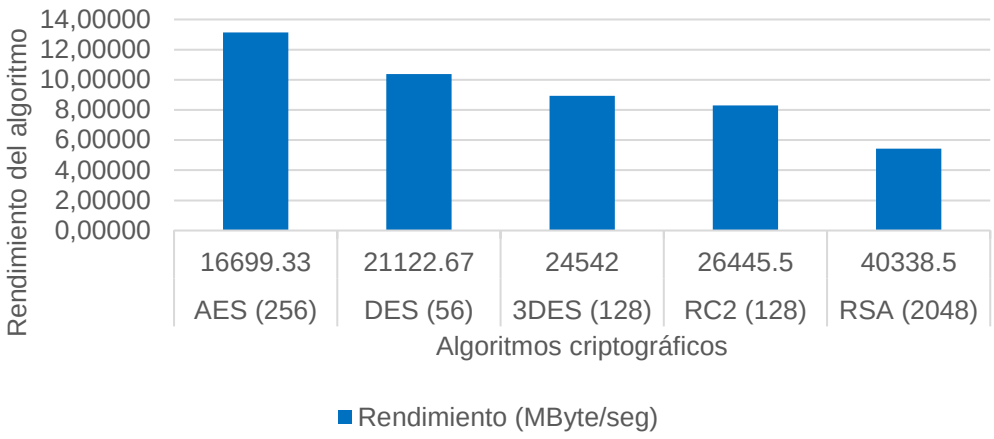


Fig. 35. Comparación del rendimiento en MByte/seg de los algoritmos AES (256), DES (56), 3DES (128), RC2 (128) y RSA (2048).

AES-256 algoritmo seleccionado.

El estudio muestra que el algoritmo AES-256 es el más rápido y eficiente en términos de tiempo de procesamiento y rendimiento en comparación con otros algoritmos. Estos resultados son útiles para aplicaciones prácticas que requieren un alto nivel de seguridad y eficiencia en la transmisión y almacenamiento de datos cifrados, en este segundo estudio se selecciona el algoritmo de encriptación AES-256.

En este artículo, se presentó una comparación del rendimiento entre ChaCha20 y otros dos algoritmos de encriptación, AES y 3DES. El enfoque principal se centra en analizar el tiempo de ejecución, el consumo de CPU y el consumo de energía requeridos por cada algoritmo tanto para encriptar como para desencriptar información. Para llevar a cabo este estudio, se diseñó un experimento en el que se utilizó un paquete de 150 (KB) de diversos formatos y tamaños para evaluar el desempeño de cada algoritmo [17].

TABLA XXI

COMPARACIÓN DE ALGORITMOS CHACHA20, AES-128, AES-192, AES-256 Y 3DES

N°	Algoritmo	Paquete (KB)	Encriptado (Seg)	Desencriptado (Seg)
1	Chacha20	150	14,1	12,5
2	AES-128	150	18,3	17,8
3	AES-192	150	17,2	17,1
4	AES-256	150	17	16,5
5	3DES	150	24,1	25,2

En ambos casos, el algoritmo 3DES se destaca como el que demanda más recursos computacionales. En contraste, se observa que las diferentes versiones del algoritmo AES no presentan diferencias significativas, ya que todas exhiben un consumo muy similar al algoritmo ChaCha20.

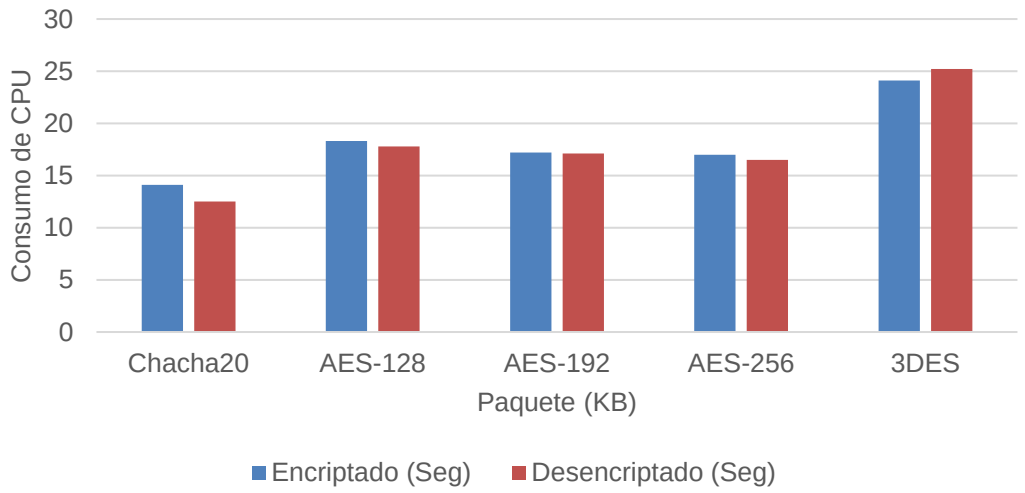


Fig. 36. Consumo de CPU de los algoritmos Chacha20, AES-128, AES-192, AES-256 y 3DES.

En lo que respecta al consumo de energía, se observa que el algoritmo ChaCha20 registra valores más bajos. Sin embargo, no se observa una diferencia significativa entre todos los algoritmos, ya que las discrepancias son inferiores a 5 Watts en la mayoría de los casos.

TABLA XXII

COMPARACIÓN DE ALGORITMOS CHACHA20, AES-128, AES-192, AES-256 Y 3DES

N°	Algoritmo	Paquete (KB)	Encriptado (Seg)	Desencriptado (Seg)
1	Chacha20	150	19,2	19,3
2	AES-128	150	21,3	20,5
3	AES-192	150	22,1	22,1
4	AES-256	150	21	21,6
5	3DES	150	23	24,2

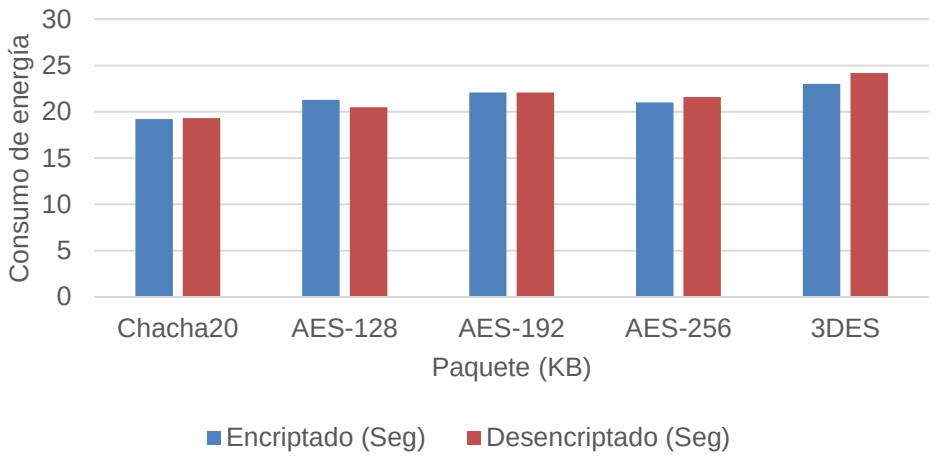


Fig. 37. Consumo de energía de los algoritmos Chacha20, AES-128, AES-192, AES-256 y 3DES.

Chacha20 algoritmo seleccionado.

Tras el análisis realizado, se llegó a la conclusión de que el algoritmo ChaCha20 es el más veloz, el algoritmo AES consume la menor cantidad de recursos de CPU, y no se encontraron diferencias significativas en cuanto al consumo de energía entre ninguno de los algoritmos evaluados.

Después de realizar una comparación de diferentes algoritmos de encriptación, se muestran los algoritmos seleccionados mediante una revisión de los artículos científicos, donde se evaluó diferentes indicadores para verificar si es viable en este aporte a la investigación, tras evaluar los diferentes criterios se seleccionó los siguientes.

TABLA XXIII

COMPARACIÓN DE LOS ALGORITMOS SELECCIONADOS AES-256, CHACHA20 Y BLOWFISH

Variables	AES-256	Chacha20	Blowfish
Contexto	Es ampliamente reconocido como uno de los algoritmos de encriptación más seguros y eficientes.	Es ampliamente utilizado en aplicaciones de cifrado de datos y protocolos de seguridad, como TLS.	Es conocido por ser rápido y eficiente, lo que hace adecuado para aplicaciones donde se requiere un procesamiento ágil de datos.
Indicador	-Tiempo de cifrado -Tiempo de descifrado -Rendimiento en cifrado. -Rendimiento en descifrado.	-Consumo de CPU. -Consumo de energía.	-Tiempo de cifrado. -Tiempo de descifrado. -Rendimiento en cifrado. -Rendimiento en descifrado.

Objetivo 02. Identificar y analizar los principales ataques criptográficos dirigidos a documentos encriptados.

Para identificar los principales ataques que atentan contra los algoritmos criptográficos que velan por la seguridad de la información, se ejecutó una revisión de artículos que pusieran de manifiesto dichos ataques para cada uno de los tres algoritmos seleccionados en el paso previo: AES-256, Chacha20 y Blowfish. Para ello se realizó una secuencia similar a la que siguió para buscar en la literatura los algoritmos criptográficos. A continuación, se muestran los ataques según algoritmos de manera específica para cada uno de ellos:

TABLA XXIV

RESUMEN DE ATAQUES CRIPTOGRÁFICOS SEGÚN LITERATURA

N°	Ataque		Descripción
	Nombre original	Traducción al castellano	
1	Ciphertext-only attack	Ataque con sólo texto cifrado	“Es un tipo de criptoanálisis donde el atacante sólo tiene acceso al mensaje cifrado e intenta recuperar el texto plano original o la clave utilizada para cifrarlo. Se trata de uno de los escenarios más desafiantes y comunes en criptografía, ya que requiere que el atacante explote cualquier debilidad o patrón en el algoritmo de cifrado o en el propio texto cifrado” [83].
2	Known-plaintext attack	Ataque de texto sin formato conocido	“El criptoanalista tiene texto sin formato y el texto cifrado correspondiente. Este tipo de ataque es más poderoso que los ataques de texto cifrado conocidos porque se conoce más información sobre el criptosistema” [84].
3	Chosen-plaintext attack	Ataque de texto sin formato elegido	“Se refiere a una situación en la que el atacante tiene la facultad de seleccionar los textos sin formato y observar los textos cifrados correspondientes. Este tipo de ataque se considera menos factible en comparación con el conocido ataque de texto sin formato” [85].
4	Chosen-cipher text attack	Ataque de texto con cifrado elegido	“La conjunción de inyecciones de fallas junto con ataques de texto cifrado elegido representa un riesgo importante para las implementaciones y puede evadir varias medidas de seguridad” [86].
5	Brute force attack	Ataque de fuerza bruta	“Un ataque de fuerza bruta en SSH es un intento por parte de los atacantes de obtener las credenciales de un usuario utilizando el protocolo SSH. Por lo general, los atacantes tienen éxito en este tipo de ataques debido a que las contraseñas utilizadas por los usuarios suelen ser débiles y fáciles de adivinar” [87].
6	Birthday attack	Ataque de cumpleaños	“Este tipo de ataque puede ser aprovechado para abusar del intercambio de información entre múltiples partes. Los ataques de cumpleaños se realizan utilizando algoritmos hash para verificar la integridad de mensajes, software o datos digitales” [88].
7	Dictionary attack	Ataque de diccionario	“El ataque de diccionario implica el uso de conjuntos de caracteres predefinidos que se organizan en forma de diccionario con el fin de intentar adivinar el texto sin formato de un texto cifrado específico” [89].

8	Man-in-the-middle attack	Ataque de intermediario	“El ataque de intermediario (MitM) es una forma de ataque perpetrada por un usuario interno malicioso que se hace pasar por uno de los equipos en una comunicación entre dos equipos. El ataque puede clasificarse en dos categorías: espionaje y manipulación” [90].
9	Meet-in-the-middle attack	Ataque por encuentro a medio camino	“El ataque de encuentro en el medio resulta especialmente efectivo contra los algoritmos de encriptación que cuentan con un espacio de claves relativamente pequeño y son vulnerables a la búsqueda exhaustiva de claves” [91].
10	Related-key attack	Ataque de clave relacionado	“Este tipo de criptoanálisis se basa en la observación del funcionamiento de un cifrado con diferentes claves para intentar descifrarlo. Son ataques criptoanalíticos de gran potencia en los que el atacante tiene la capacidad de manipular la clave secreta de un sistema criptográfico” [92].
11	Side-channel attack	Ataque de canal lateral	“En el ámbito de la ciberseguridad, es cada vez más común escuchar acerca de los ataques de canal lateral, los cuales se centran en aprovechar las debilidades de la implementación de un sistema informático en lugar de atacar directamente sus algoritmos o software. El punto de inflexión se produjo cuando se descubrieron vulnerabilidades en procesadores como Meltdown y Spectre” [93].

Nota: estos ataques criptográficos han sido recogidos de la revisión de la literatura científica.

La tabla anterior muestra el listado de ataques que se logró identificar en la revisión de la literatura, sin embargo, a continuación, se muestran los ataques por cada uno de los algoritmos que se seleccionaron en pasos anteriores.

Respecto al algoritmo AES-256 se identificaron los siguientes ataques criptográficos:

TABLA XXV

ATAQUES IDENTIFICADOS RESPECTO AL ALGORITMO AES-256

N°	Ataque criptográfico	Vulnerabilidad	Autor
1	Fuerza bruta	Gestión de contraseñas	[94]
2	Ataque de clave relacionado	Configuración incorrecta	[95]
3	Ataque de canal lateral	Filtración de información del sistema	[96]
4	Ataque por encuentro a medio camino	Recuperar la llave de cifrado	[97]
5	Ataque de texto sin formato elegido	Debilidad en la generación de claves	[98]
6	Ataque de texto sin formato conocido	Debilidad en la generación de claves	[99]

Nota: estos ataques han sido recogidos de la revisión de la literatura científica.

Respecto al algoritmo ChaCha20 se identificaron los siguientes ataques criptográficos:

TABLA XXVI

ATAQUES IDENTIFICADOS RESPECTO AL ALGORITMO CHACHA20

N°	Ataque criptográfico	Vulnerabilidad	Autor
1	Ataque de fuerza bruta	Gestión de contraseñas	[100]
2	Ataque de cumpleaños	Busca encontrar dos mensajes aleatorios que generan el mismo resumen de mensaje	[101]
3	Ataque de diccionario	Gestión de contraseñas	[100]
4	Ataque por encuentro a medio camino	Recuperar la llave de cifrado	[102]
5	Ataque de canal lateral	Filtración de información del sistema	[96]

Nota: estos ataques han sido recogidos de la revisión de la literatura científica.

Respecto al algoritmo Blowfish se identificaron los siguientes ataques criptográficos:

TABLA XXVII

ATAQUES IDENTIFICADOS RESPECTO AL ALGORITMO BLOWFISH

N°	Ataque criptográfico	Vulnerabilidad	Autor
1	Ataque de fuerza bruta	Gestión de contraseñas	[103]
2	Ataque de cumpleaños	Busca encontrar dos mensajes aleatorios que generan el mismo resumen de mensaje	[104]
3	Ataque de diccionario	Gestión de contraseñas	[105]
4	Ataque de intermediario	Interceptación de mensajes	[106]
5	Ataque por encuentro a medio camino	Recuperar la llave de cifrado	[107]

Nota: estos ataques han sido recogidos de la revisión de la literatura científica.

Asimismo, luego de identificar los ataques criptográficos para los algoritmos AES-256, Chacha20 y Blowfish, se procedió a realizar una comparativa acerca de las características básicas de dichos algoritmos, de modo que sirviese de preámbulo a la implementación de cada uno de ellos. La comparativa se muestra a continuación:

TABLA XXVIII

COMPARATIVA DE ALGORITMOS CRIPTOGRÁFICOS A IMPLEMENTAR

Algoritmo	Autor	Inventado	Tamaño de clave	Tamaño de bloque	Tipo de cifrado	Método de cifrado	Rondas	Flexibilidad	Escalabilidad	Velocidad de procesamiento	Seguridad
AES-256	Joan Daemen y Vincent Rijmen	1998	256-bit	128 bits	Simétrico	Cifrado de bloque	14	Sí	No escalable	Rápido	Excelente
Chacha20	Daniel J. Bernstein	2008	256-bit	128 bits	Simétrico	Cifrado de flujo	20	Sí	Escalable	Muy rápido	Seguro y robusto
Blowfish	Bruce Schneier	1993	Hasta 448-bit	64 bits	Simétrico	Cifrado de bloque	16-18	Sí	Escalable	Muy rápido	Seguro y resistente

Nota: estos algoritmos han sido recogidos de la revisión de la literatura científica.

Objetivo 03. Desarrollar en lenguaje de programación los algoritmos criptográficos para cifrar los datos de un texto plano.

Para el desarrollo de los algoritmos criptográficos, se necesitó previamente de establecer un contexto:

a. Archivos de texto plano, documentos empresariales (informes, memorandos y oficios).

El experimento se llevó a cabo utilizando exclusivamente un tipo de archivo de extensión (.Docx, este archivo fue seleccionado de manera que tuviera una distribución de tamaño uniforme, lo cual permitió observar cómo variaban las mediciones de los indicadores en los algoritmos Blowfish, AES-256 y Chacha20 a medida que se modificaba el tamaño del archivo utilizado.

Se llevaron a cabo las implementaciones de los algoritmos Blowfish, AES-256 y Chacha20 de acuerdo a sus respectivas especificaciones. Para ello, se utilizó el lenguaje de programación Python. Cada algoritmo fue implementado siguiendo las instrucciones teóricas descritas en la investigación. A continuación, se detallan los pasos seguidos durante la implementación de estos algoritmos:

Entorno de desarrollo integrado (IDE) PyCharm

Para la ejecución de los algoritmos se utilizó PyCharm, es un entorno de desarrollo integrado (IDE) diseñado específicamente para el desarrollo en Python. Es desarrollado por JetBrains y es ampliamente utilizado por programadores y desarrolladores de Python, este IDE fue utilizado para ejecutar los tres algoritmos previamente seleccionados en el primer objetivo: Blowfish, AES-256 y Chacha20.

Entorno virtual Virtualenv

Para crear y gestionar el entorno virtual se utilizó Virtualenv en PyCharm, es una valiosa herramienta que permite crear y gestionar entornos virtuales separados en proyectos

de Python. Esto simplifica la administración de dependencias y versiones de bibliotecas, asegurando un entorno de desarrollo coherente y proporcionando la capacidad de trabajar de manera independiente y controlada en varios proyectos de Python.

AWS “Amazon Web Services”

AWS es una plataforma de servicios en la nube ofrecida por Amazon, que proporciona una amplia variedad de servicios, como almacenamiento, cómputo, bases de datos, análisis, inteligencia artificial, Internet de las cosas (IoT), seguridad y más, que permiten a las empresas escalar y crecer de manera más eficiente.

Amazon Lightsail es un servicio de computación en la nube ofrecido por Amazon Web Services (AWS) que facilita la implementación, administración y escalabilidad de aplicaciones web y sitios web. Está diseñado para ser una solución sencilla y fácil de usar. Se crea el servidor con el nombre de tesis-server como se muestra en la Fig. 41. Además, la base de datos se creó en MySQL database.

TABLA XXIX

REQUERIMIENTOS UTILIZADOS EN EL IDE PYCHARM

Requerimiento	Versión	Descripción
click	7.1.2	Simplifica la creación de interfaces de línea de comandos en Python
Flask	1.1.2	Framework web ligero y flexible para Python que se utiliza para desarrollar aplicaciones web
gunicorn	20.0.4	Servidor web compatible con Python que se utiliza para implementar y ejecutar aplicaciones web en entornos de producción
itsdangerous	1.1.0	Biblioteca que proporciona herramientas para la generación y verificación de tokens de seguridad en aplicaciones web
Jinja2	2.11.3	Potente motor de plantillas utilizado para generar y renderizar contenido dinámico en aplicaciones web

MarkupSafe	1.1.1	Ayuda a prevenir ataques de XSS, garantiza la seguridad en la generación de contenido dinámico y mejora la integridad del marcado en general.
Werkzeug	1.0.1	Proporciona funcionalidades para el enrutamiento de URL, el manejo de solicitudes y respuestas HTTP
numpy		Biblioteca fundamental para el procesamiento numérico y científico en Python
pandas		Ofrece una amplia gama de funcionalidades para el manejo de datos, preprocesamiento, análisis estadístico y más
scikit-learn		biblioteca de aprendizaje automático que ofrece algoritmos, herramientas de preprocesamiento de datos, evaluación de modelos y validación cruzada.
Flask-Cors	3.0.10	Simplifica la configuración y el manejo de CORS, permitiendo el acceso controlado a recursos desde diferentes dominios y asegurando la seguridad de la aplicación.
cryptography		Biblioteca que proporciona funciones y herramientas para realizar operaciones criptográficas en Python

Para la implementación de los algoritmos se trabajó en un select, este formulario permite verificar las opciones que fueron almacenadas, esta función se utiliza tanto para el encriptado y desencriptado, posteriormente se implementa otro select para seleccionar el tipo de archivo.

TABLA XXX

FUNCIONES BÁSICAS DE LOS ALGORITMOS CRIPTOGRÁFICOS

Algoritmos a seleccionar	Descripción	Función
<select name="encryptionType"> <option value="AES">AES-256</option> <option value="Blowfish">Blowfish</option> <option value="ChaCha20">ChaCha20</option> </select>	Se utiliza el elemento select para crear una lista desplegable en un formulario web, permitiendo que el usuario seleccione una opción que crea por conveniente para	Encriptar y desencriptar

	encriptar su documento.	
<select id="dataType" class="form-select" name="dataType"> <option value="text">Texto</option> <option value="file">Archivo</option> </select>	En este formulario se puede seleccionar el tipo de datos	Tipo de datos

Especificaciones del contenido para la ejecución de los tres algoritmos implementado, donde se trabaja con la misma clave tanto para el encriptado y desencriptado.

TABLA XXXI

ESPECIFICACIONES DEL CONTENIDO PARA LA EJECUCION DEL PROGRAMA

Código	Descripción	Función
password = "2u-tuLnyApEBk1nhSzqYQ3Qvj_gEYb8ckM3zV7PsmI"	generación de clave	Clave para encriptar y desencriptar
def encrypt_file(file): data = file.read() encrypted_data = cipher_suite.encrypt(data) encrypted_filename = 'encrypted_file.bin' encrypted_filepath = os.path.join(app.root_path, 'uploads', encrypted_filename)	Se ejecuta el método de encriptado, posteriormente se guarda el archivo con el nombre de 'encrypted_file.bin'	Encriptar
def decrypt_file(file): data = file.read() decrypted_data = cipher_suite.decrypt(data) decrypted_filename = file_path_output file_path_output = 'archivo.docx'	Se hace el llamado a la función de desencriptar, luego se guarda con el nombre de 'archivo.docx'	Desencriptar

TABLA XXXII

INDICADORES A EVALUAR CON LOS ALGORITMOS IMPLEMENTADOS

Código	Descripción	Función
end_time = time.time() decryption_time = end_time - start_time	Registra el tiempo final de la encriptación	Encriptación en segundos
end_time = time.time() decryption_time = end_time - start_time	Registrar el tiempo de finalización de la encriptación	Desencriptación en segundos
rendimiento = file_size/decryption_time return jsonify({'result': ciphertext,'decryption_time':round(decryption_time, 4),'rendimiento':rou nd(rendimiento, 4),'state':True,'file_size':file_size})	Está asociada con el rendimiento de cifrado, que indica la rapidez con la que el algoritmo realiza la tarea de encriptación	Rendimiento de encriptado

if decryption_time != -1:
rendimiento = file_size/decryption_time
return jsonify({'result': plaintext,'decryption_time':round(decryption_time, 4),'rendimiento':round(rendimiento, 4),'state':True,'file_size':file_size})
else:
return jsonify({'result': plaintext,'decryption_time':round(decryption_time, 4),'state':False,'file_size':0})
	Está vinculada al rendimiento de descifrado, lo cual indica la rapidez con la que el algoritmo realiza la tarea de desencriptación	Rendimiento de desencriptado

file_size = len(blob) print("Tamaño archivo byte",file_size)	Utiliza la función file_size para recuperar el tamaño del archivo en bytes	Tamaño del archivo original (bytes)
rendimiento = file_size/decryption_time return jsonify({'result': ciphertext,'decryption_time':round(decryption_time, 4),'rendimiento':round(rendimiento, 4),'state':True,'file_size':file_size})	Muestra el archivo encriptado en bytes	Tamaño del archivo encriptado (bytes)

Implementacion del algoritmo Blowfish

Algoritmo criptográfico Blowfish		
Código	Descripción	Función
def encrypt_file_blowfish(key, input_file, output_file): #key = pad_key(key)	Crea un objeto de cifrado utilizando la	Encriptar

iv = b'\x00' * 8 # Initialization Vector
start_time = time.time()
	clave proporcionada, cifra los datos rellenados y vuelve a escribir los datos cifrados en el archivo.	

def decrypt_file_blowfish(key, input_file, output_file):
#key_data = pad_key(key)
iv = b'\x00' * 8 # Initialization Vector
start_time = time.time()
print("KEY_SECURITY",key)
file_size = ''
	Blowfish utiliza la clave proporcionada, descifra los datos cifrados, elimina cualquier relleno de los datos descifrados y vuelve a escribir los datos descifrados en el archivo.	Desencripta

Implementacion del algoritmo AES-256

Algoritmo criptográfico AES-256		
Código	Descripción	Función
def encrypt_file_2_aes (data, output_filename): start_time = time.time() encrypted_data = cipher_suite.encrypt(data) encrypted_filepath = os.path.join(app.root_path, 'uploads', output_filename)	Lee el contenido del archivo, genera un vector de inicialización aleatorio (IV), crea un objeto de cifrado AES utilizando la clave y el IV proporcionados	Encriptar

def decrypt_file_example(encryption_type, file):
if encryption_type == 'AES':
data = file.read()
start_time = time.time()
try:
decrypted_data = cipher_suite.decrypt(data)
print("Datos desencriptados:", decrypted_data)
decrypted_filepath = os.path.join(app.root_path, 'uploads', file_path_output)
	crea un objeto de cifrado AES utilizando la clave y el IV proporcionados,
	descifra los datos cifrados, desbloquea los datos descifrados y escribe los datos descifrados.
		Desencripta

Implementacion del algoritmo ChaCha20

Algoritmo criptográfico ChaCha20		
Código	Descripción	Función
def encrypt_file_chacha20(key, input_file, output_file): nonce = b'\x00' * 16 start_time = time.time() file_size = '' encrypted_filepath = os.path.join(app.root_path, 'uploads', output_file)	Lee el contenido del archivo, genera un valor aleatorio llamado "nonce", lo utiliza más la clave proporcionada para crear un objeto de cifrado, cifra los datos del archivo utilizando dicho objeto de cifrado.	Encriptar
def decrypt_file_chacha20(key, input_file, output_file): nonce = b'\x00' * 16 start_time = time.time() file_size = '' encrypted_filepath = os.path.join(app.root_path, 'uploads', output_file) with open(encrypted_filepath, 'wb') as file_out:	Lee el contenido del archivo cifrado, extrae el nonce desde el principio de los datos del archivo, crea un objeto de cifrado utilizando la clave y el nonce proporcionados descifra los datos	Desencripta

Objetivo 04. Proponer recomendaciones precisas que promuevan la ejecución segura y medir la satisfacción del usuario.

Con el propósito de mantener la seguridad de la información cuando se transmite por internet es necesario tener en cuenta lo siguiente:

- Selección de algoritmos de encriptación: Seleccionar el algoritmo de encriptación reconocidos y actualizados, como son: Blowfish, AES-256 y ChaCha20, que demostraron garantizar la información mediante las pruebas.
- Gestión segura de claves: Haciendo uso de un sistema generados de claves aleatorias, teniendo en cuenta letras minúsculas, mayúsculas, símbolos y caracteres, asegurando que las claves de encriptación sean almacenadas y

gestionadas de manera segura. Considera el uso de servicios especializados de gestión de claves si estás manejando grandes volúmenes de datos sensibles, por ejemplo, el generador de contraseñas aleatorias dashlane.

- Contraseñas robustas: Si se utiliza una contraseña para proteger la clave de encriptación, asegúrate de que sea robusta y única. Evita el uso de contraseñas predecibles y fomenta prácticas de contraseña segura.
- Almacenamiento seguro de archivos encriptados: Protege adecuadamente los archivos encriptados, asegúrate de que los archivos y las claves estén almacenados en ubicaciones seguras y accesibles solo para personal autorizado haciendo el uso de su clave y contraseña.
- Las contraseñas tienen que se empleadas con responsabilidad y no facilitar a terceros usuarios.
- Evitar realizar secuencias lógicas, como nombres, teléfonos o fechas de cumpleaños.
- La contraseña cambiar una vez al mes y no usar contraseñas anteriores.

Al usuario se muestra una opción donde puede realizar una encuesta y así poder medir la satisfacción de usuario haciendo uso del cuestionario SUS.

b. System Usability Scale (SUS), Escala de usabilidad del sistema

Las pruebas de experiencia de usuario tienen como objetivo fundamental garantizar que el producto o servicio cumpla con las necesidades de los usuarios, mientras se analiza su facilidad de uso y se verifica su utilidad y la satisfacción que brinda a los usuarios. La evaluación de la usabilidad de un sistema, conforme a la definición de la norma ISO 9241 Parte 11, está intrínsecamente ligada al contexto de uso del sistema. Esto implica considerar quiénes son los usuarios, sus metas al utilizar el sistema y el entorno en el que tienen lugar

las interacciones.

Este cuestionario incluye un total de 10 preguntas, donde se solicita al usuario que proporcione una calificación para cada una de ellas. Cada pregunta está diseñada con un sistema de puntuación con un rango que va desde 1 (No satisfecho), 2 (Poco satisfecho), 3 (Satisfecho), 4 (Muy satisfecho). El cuestionario se detalla en los anexos Fig. 39 y Fig. 40.

Respuestas Individuales: Es la suma total de las respuestas obtenidas que han marcado cada usuario en las 10 preguntas planteadas en el cuestionario, cada pregunta consta con un rango del 1 al 4, donde, 1 (No satisfecho), 2 (Poco satisfecho), 3 (Satisfecho), 4 (Muy satisfecho), variando la puntuación total entre 0 y 40 como se muestra en la TABLA XXXIII.

TABLA XXXIII

CALIFICACION INDIVIDUAL DEL CUESTIONARIO SUS

ID	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	TOTAL
USER1	4	3	3	4	4	3	3	3	3	4	34
USER2	3	3	3	4	4	3	2	3	3	3	31
USER3	3	4	4	3	4	3	3	4	3	3	34
USER4	3	4	2	4	2	4	2	4	3	3	31
USER5	4	3	4	4	3	1	3	3	4	3	32
USER6	3	3	4	3	3	4	3	3	3	4	33
USER7	4	3	3	4	3	4	3	4	3	3	34
USER8	4	3	4	3	4	2	3	4	4	4	35
USER9	2	3	3	3	3	4	2	3	3	3	29

Nota. Elaboración propia.

Revisar y ajustar la calificación total en porcentaje: Con el propósito de obtener el puntaje final del cuestionario de Escala de Usabilidad del Sistema, se multiplica la suma de las calificaciones individuales por 2.5, ajustando a una escala como mínimo 0 y máximo 100.

TABLA XXXIV

CALIFICACION TOTAL EN PORCENTAJE DEL CUESTIONARIO SUS

ID	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	TOTAL
USER1	10	8	8	10	10	8	8	8	8	10	85%
USER2	8	8	8	10	10	8	5	8	8	7,5	77,5%
USER3	8	10	10	8	10	8	8	10	8	7,5	85%
USER4	8	10	5	10	5	10	5	10	8	7,5	77,5%

USER5	10	8	10	10	8	3	8	8	10	7,5	80%
USER6	8	8	10	8	8	10	8	8	8	10	82,5%
USER7	10	8	8	10	8	10	8	10	8	7,5	85%
USER8	10	8	10	8	10	5	8	10	10	10	87,5%
USER9	5	8	8	8	8	10	5	8	8	7,5	72,5%

Nota. Elaboración propia.

Análisis de calificación final: Este análisis, se examinan los factores que componen la calificación total, con el propósito de identificar la usabilidad del programa. La calificación total se evalúa de la siguiente manera:

0 a 50: No satisfecho, se necesita hacer mejoras en el programa.

51 a 66,67: Poco satisfecho, el programa cumple con requisitos básicos.

67 - 83,34: Satisfecho, el programa es de fácil uso.

84 - 100: Muy satisfecho, el programa está bien diseñado, eficiente y satisfactorio.

En la calificación final del cuestionario de Escala de Usabilidad del Sistema, si el porcentaje presenta mayor del 67% se interpreta como buena aceptación por los usuarios, mientras que, si presenta menor del 67%, quiere decir que no es aceptable.

TABLA XXXV

CALIFICACION FINAL DEL CUESTIONARIO SUS

ID	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	TOTAL	CONDICION
USER1	10	8	8	10	10	8	8	8	8	10	85,0%	Muy satisfecho
USER2	8	8	8	10	10	8	5	8	8	7,5	77,5%	Satisfecho
USER3	8	10	10	8	10	8	8	10	8	7,5	85,0%	Muy satisfecho
USER4	8	10	5	10	5	10	5	10	8	7,5	77,5%	Satisfecho
USER5	10	8	10	10	8	3	8	8	10	7,5	80,0%	Satisfecho
USER6	8	8	10	8	8	10	8	8	8	10	82,5%	Satisfecho
USER7	10	8	8	10	8	10	8	10	8	7,5	85,0%	Muy satisfecho
USER8	10	8	10	8	10	5	8	10	10	10	87,5%	Muy satisfecho
USER9	5	8	8	8	8	10	5	8	8	7,5	72,5%	Satisfecho

Nota. Elaboración propia.

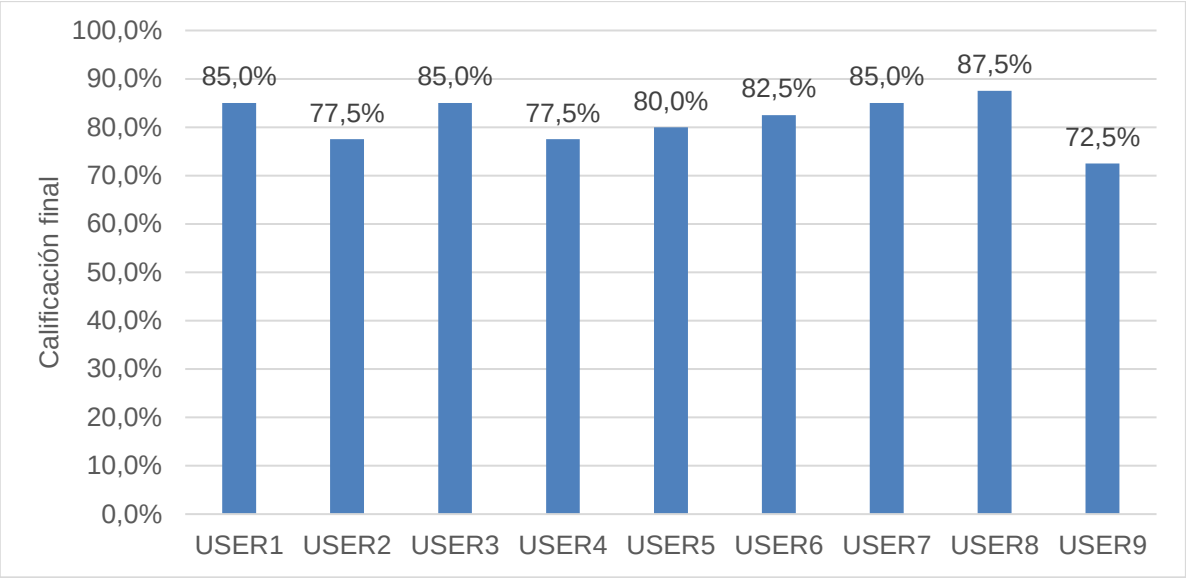


Fig. 38. Calificación final del cuestionario SUS.

Al realizar el análisis de datos de los usuarios, se logró entender que cinco de ellos quedaron satisfechos con el uso del programa y cuatro usuarios marcaron una puntuación de muy satisfecho, Según las condiciones planteadas anteriormente.

IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

Se seleccionaron los algoritmos criptográficos más relevantes que garantizan la seguridad de los datos mediante la adopción de las directrices para la revisión sistemática que propusieron Kitchenham & Charters, buscando información de bases de datos de revistas indizadas, identificándose un total de quince algoritmos criptográficos existentes para dicho fin, de los cuales, Blowfish, AES-256 y Chacha20 y fueron aquellos que tenían las mejores valoraciones en cuanto a indicadores tales como tiempo de cifrado, tiempo de descifrado, rendimiento, entre otros.

Se identificaron los principales ataques conocidos contra los algoritmos criptográficos Blowfish, AES-256 y Chacha20, siguiendo la misma línea metodológica de Kitchenham & Charters, recogiendo un total de once ataques, destacando entre ellos los ataques de fuerza bruta, los ataques Dictionary attack, entre otros, los cuales atentan contra la seguridad de la información vulnerando la gestión de contraseñas, la interceptación de mensajes, la filtración de información, entre otros.

Se desarrollaron los algoritmos criptográficos para cifrar los datos de un texto plano en lenguaje de programación Phyton, empleando un entorno de desarrollo integrado PyCharm y considerando para ello un contexto de archivos de texto pertenecientes a documentos empresariales con extensión “.docx”, el cual permitió observar cómo variaban las mediciones de los indicadores en los algoritmos Blowfish, AES-256 y Chacha20.

Al usuario se le especificó que detalles tener en cuenta al momento de hacer uso del programa FortiDoc, algunas de ellas la generación de claves seguras, claves robustas, almacenamiento de los archivos encriptados, entre otros. Ademas, al usuario se evaluó mediante un cuestionario que incluye un total de 10 preguntas, donde se solicita que proporcione una calificación para cada una de ellas. Cada pregunta está diseñada con un sistema de puntuación con un rango que va desde 1 (No satisfecho), 2 (Poco satisfecho), 3 (Satisfecho), 4 (Muy satisfecho).

4.2. Recomendaciones

A los investigadores de pregrado, postgrado y en general, adoptar las directrices para la revisión sistemática que propusieron Kitchenham & Charters, donde muestran una hoja de ruta que tiene como propósito la búsqueda de información en bases de datos de revistas indizadas las cuales permiten obtener artículos de cuantiosa calidad académica, útil y fidedigna al momento de investigar.

A los estudiosos que pretenden investigar acerca de algoritmos criptográficos, hacer uso de los indicadores empleados en la operacionalización de variables de este estudio ya que reúnen métricas apropiadas para la valoración individual de cada uno de los algoritmos a evaluar en cuanto a la confidencialidad, integridad y disponibilidad de la información y que cuentan además con el soporte académico de una buena revisión sistemática de la literatura.

A la comunidad universitaria peruana, el aprendizaje de Python, en etapas tempranas de la carrera profesional, como lenguaje de programación ya que les permitirá en la criptografía, proteger los datos de cifrado y las comunicaciones, haciendo empleo de la codificación, permitiendo que solo público objetivo sea capaz de descifrar y procesar dichos datos, permitiendo limitar el acceso no autorizado a la información.

A la comunidad académica internacional, hacer uso de los Principios Criptográficos de Kerckhoffs estos garantizan la seguridad de un sistema criptográfico, por la robustez de cada uno de los seis principios en términos de seguridad, verificación, simplicidad, administración de claves, entre otros factores y que a día de hoy no han podido ser derrotados por otra teoría, sino que, por el contrario, siguen erigiéndose como los de mayor adopción en criptografía.

REFERENCIAS

[1] Y. Sadqi y Y. Maleh, «A systematic review and taxonomy of web applications threats,» Information Security Journal A Global Perspective, p. 2, 2020.

[2] Netcraft, «Encuesta de servidores web de enero de 2022,» 17 Enero 2022. [En línea]. Available: https://news.netcraft.com/archives/2022/01/17/january-2022-web-server-survey.html.

[3] Dihuni, «Estadísticas diarias de Big Data: 2,5 quintillones de bytes de datos creados diariamente,» 10 Abril 2020. [En línea]. Available: https://www.dihuni.com/2020/04/10/every-day-big-data-statistics-2-5-quintillion-bytes-of-data-created-daily/.

[4] B. Miranda, «Julian Assange: así fue la gran filtración de documentos clasificados en 2010 por la que EE.UU. pide la extradición del fundador de WikiLeaks,» BBC News Mundo, 12 Abril 2019.

[5] B. H. Tang, S. Y. Deng, C. P. Wu, Z. FengWang, J. P. An y Z. F. Zhao, «Investigación sobre métodos de detección de tráfico malicioso basados en aprendizaje profundo,» 2021 IV Congreso Internacional de Reconocimiento de Patrones e Inteligencia Artificial (PRAI), p. 2, 2021.

[6] Y. F. Chala, «Importancia de la aplicación del mecanismo de cifrado de información en las empresas para la prevención de riesgos como ataques, plagio y pérdida de la confidencialidad.,» Lima, 2019.

[7] R. M. S. Gavino, «Ciberseguridad en la actitividad organizacional de la era tecnológica,» Lima, 2018.

[8] E. Angeles, «Ciberdelitos en el Perú: se elevan denuncias de fraude informático y suplantación de identidad,» El peruano, p. 1, 02 Junio 2021.

[9] C. H. D. Aguirre, «Más de 136 denuncias por incumplir la Ley de Protección

de Datos Personales,» El peruano, p. 1, 29 Diciembre 2021.

[10] J. Tidy, «EE.UU. incauta medio millón de dólares que fueron robados por presuntos hackers norcoreanos,» 21 Julio 2022. [En línea]. Available: https://www.bbc.com/mundo/noticias-internacional-62246974.

[11] S. Lyngaas, «Hackers vulneran un sistema de salud de Florida, exponiendo potencialmente los datos de 1,3 millones de personas,» 4 Enero 2022. [En línea]. Available: https://cnnespanol.cnn.com/2022/01/04/hackers-floridfa-sistema-salud-datos-millones-personas-trax/.

[12] P. D. Ortega, «Los ciberdelitos aumentan un 72% en España,» 8 Febreo 2023. [En línea]. Available: https://elpais.com/espana/2023-02-08/los-ciberdelitos-aumentan-un-72-en-espana.html.

[13] R. Holgado, «Alertan de estafas que suplantan la identidad de 9 entidades bancarias distintas,» 17 Octubre 2022. [En línea]. Available: https://www.20minutos.es/tecnologia/ciberseguridad/alertan-de-estafas-que-suplantan-la-identidad-de-9-entidades-bancarias-distintas-5069177/.

[14] E. irakurri, «Los delitos informáticos han aumentado un 80 % en los últimos cuatro años en Euskadi,» 14 Octubre 2022. [En línea]. Available: https://www.eitb.eus/es/noticias/sociedad/detalle/8984358/los-delitos-informaticos-han-aumentado-80-en-ultimos-cuatro-anos-en-euskadi/.

[15] B. Padmavathi y S. R. Kumari, «A Survey on Performance Analysis of DES, AES and RSA Algorithm along with LSB Substitution Technique,» International Journal of Science and Research (IJSR), India Online ISSN: 2319-7064, vol. 2, pp. 1-4, 2019.

[16] A. Kubadia, D. Idnani y Y. Jain, «Performance Evaluation of AES, ARC2, Blowfish, CAST and DES3 for Standalone Systems,» Proceedings of the Third International Conference on Computing Methodologies and Communication

(ICCMC 2019) IEEE Xplore Part Number: CFP19K25-ART; ISBN: 978-1-5386-7808-4, pp. 1-5, 2019.

[17] C. L. S. Claros, B. L. Coca y S. J. P. Alcocer, «Comparative study of the symmetric cryptography algorithms AES, 3DES and ChaCha20,» ACTA NOVA; ISSN 1683-0768 Artículos Científicos, vol. 10, nº 3, p. 15, 2022.

[18] H. Mawengkang, I. L. Sitepu y S. Efendi, «Security analysis in file with combinations One Time Pad Algorithm and Vigenere Algorithm,» IOP Conf. Series: Materials Science and Engineering 420 (2018) 012129 doi:10.1088/1757-899X/420/1/012129, pp. 2-3, 2018.

[19] K. m. Meghna, «Block Cipher modes of Operation,» 24 Febrero 2022. [En línea]. Available: https://www.geeksforgeeks.org/block-cipher-modes-of-operation/.

[20] R. N. Wright, «Cipher Feedback Mode (CFB),» de Encyclopedia of Physical Science and Technology, Tarzana, California, USA, Tercera Edición, 2003, pp. 61-77.

[21] Mkyong, «ChaCha20-Poly1305 encryption,» 13 Mayo 2020. [En línea]. Available: https://mkyong.com/java/java-11-chacha20-poly1305-encryption-examples/.

[22] E. A. Hernández, «Algoritmo de Firma Digital DSA,» 10 Enero 2018. [En línea]. Available: https://repository.uaeh.edu.mx/bitstream/handle/123456789/18970.

[23] B. H. M. Fernandes, «¿Qué es Base64, para qué sirve y cómo funciona?,» 24 Febrero 2020. [En línea]. Available: https://marquesfernandes.com/es/tecnologia-es/que-y-base64-para-que-serve-y-como-funciona/.

[24] S. P. A. Sánchez, G. J. R. González, C. A. Triana y L. Perez, «Medida del nivel de seguridad informática de las pequeñas y medianas empresas (PYMEs) en

Colombia,» Corporación Universitaria Latinoamericana, p. 1, 2021.

[25] E. Mifsud, «Introducción a la seguridad informática - Seguridad de la información / Seguridad informática,» 26 Marzo 2012. [En línea]. Available: http://recursostic.educacion.es/observatorio/web/ca/software/software-general/1040-introduccion-a-la-seguridad-informatica?start=1.

[26] G. Westreicher, «Definición técnica de dato,» 1 Diciembre 2020. [En línea]. Available: https://economipedia.com/definiciones/dato.html.

[27] A. Morales, «Información,» 9 Diciembre 2019. [En línea]. Available: https://www.todamateria.com/informacion/.

[28] E. V. Briceño, «Seguridad de la información,» de Seguridad de la información, Costa Rica, Editorial Área de Innovación y Desarrollo,S.L., 2021, p. 9.

[29] A. Pérez, «Seguridad informática,» 2 Diciembre 2019. [En línea]. Available: https://www.obsbusiness.school/blog/seguridad-informatica-definicion.

[30] C. P. Claudio Armando, «repositorio.utn.edu.ec,» 2011. [En línea]. Available: http://repositorio.utn.edu.ec/bitstream/123456789/593/4/CAPITULO%20IV.pdf.

[31] L. Sanjuan, «Seminario – Seguridad en desarrollo del Software,» 2011. [En línea]. Available: http://manglar.uninorte.edu.co/bitstream/handle/10584/2204/Crip?sequence=1.

[32] F. Vivar, «Algorithms, applications and Big Data, new paradigms in the process of communication and teaching-learning of data journalism,» Scielo, pp. 8 - 9, 2018.

[33] C. H. A. Urbina y N. J. J. V. Ramos, «Análisis de algoritmos de encriptación de datos de texto, una revisión de la literatura científica,» 2019. [En línea]. Available:

https://repositorio.upn.edu.pe/bitstream/handle/11537/31391/Cabanillas%20Urbina%2c%20Henry%20Alexander%20-%20Nizama%20Ramos%2c%20Juan%20Jose%20Victor.pdf?sequence=1&isAllowed=y.

[34] H. Patricio, «Tipos de algoritmos criptográficos: Cifrados de bloque,» 12 Marzo 2020. [En línea]. Available: https://blog.thedojo.mx/2020/12/03/tipos-de-algoritmos-criptograficos.html.

[35] C. E. López Rojas, «Cifrador de bloques, utilizando la transformación caótica unidimensional de la tent,» Mayo 2011. [En línea]. Available: https://www.repositoriodigital.ipn.mx/bitstream/123456789/12679/1/Tesis%20TENT%20MAP%20Cesar%20Rojas.pdf.

[36] Alegsa, «Definición de texto plano (Sin formato),» 15 Junio 2018. [En línea]. Available: https://www.alegsa.com.ar/Dic/texto_plano.php#gsc.tab=0.

[37] M. Muñoz, «¿Qué es el número Nonce?,» 27 Junio 2021. [En línea]. Available: https://blog.bitnovo.com/que-es-el-numero-nonce/.

[38] D. Vargas, «¿Qué es TLS? Definición, funcionamiento y diferencias con SSL y HTTPS,» 24 Noviembre 2022. [En línea]. Available: https://www.hostinger.es/tutoriales/que-es-tls.

[39] D. O. L. Ramírez y C. C. M. Espinosa, «El Cifrado Web (SSL/TLS),» 2018. [En línea]. Available: https://revista.seguridad.unam.mx/numero-10/el-cifrado-web-ssltls#:~:text=SSL%20(Secure%20Sockets%20Layer)%20traducido,seguras%20a%20trav%C3%A9s%20de%20Internet..

[40] Á. P. Guijarro, «Protocolo HTTP,» 1 Enero 2012. [En línea]. Available: https://alvaroprimoguijarro.files.wordpress.com/2012/01/ud04_http_alvaroprimoguijarro.pdf.

[41] A. López, «Todo sobre criptografía: Algoritmos de clave Simétrica y Asimétrica,» 8 Octubre 2022. [En línea]. Available: https://www.redeszone.net/tutoriales/seguridad/criptografia-algoritmos-clave-simetrica-asimetrica/.

[42] Y. T. Medina Vargas y H. A. Miranda Mnedez, «Comparison of Algorithms Based Cryptography Symmetric DES, AES and 3DES,» Mundo Fesc, Junio 2015.

[43] T. R. Cardona, «Criptografía Simétrica y Asimétrica y su aplicación en medios digitales como las imagenes, video y audio,» Lima, 2020, pp. 14-15.

[44] J. C. Mendoza, «Demostración de cifrado simétrcio y asimétrico,» 2004. [En línea]. Available: https://dialnet.unirioja.es/servlet/articulo?codigo=5972695.

[45] G. M. Díaz y E. E. V. Rivas, «Algoritmo para mostrar el proceso de encriptación y desencriptación de datos en PHP. Una aplicación de seguridad informática,» p. 4, 2001.

[46] N. Gutiérrez, «Encriptacion de datos: Una guía para buenas prácticas de seguridad,» 05 Marzo 2020. [En línea]. Available: https://preyproject.com/blog/es/la-encriptacion-de-datos-una-guia-para-buenas-practicas-de-seguridad/.

[47] A. Pousa, «Algoritmo de cifrado simétrico AES. aceleración de tiempo de cómputo sobre arquitecturas multicore,» Diciembre 2011. [En línea]. Available: http://sedici.unlp.edu.ar/bitstream/handle/10915/4210/Documento_completo.pdf?sequence=1&isAllowed=y.

[48] B. Gómes, «AES-256 ¿Qué es? ¿Cómo funciona? (Mejor explicación),» 18 Abril 2021. [En línea]. Available: https://www.profesionalreview.com/2021/04/18/aes-256/.

[49] L. Javier, «Sistema de cifrado AES-256 bits, ¿es realmente tan seguro?,» 24

Enero 2023. [En línea]. Available: https://hardzone.es/tutoriales/rendimiento/cifrado-aes-256-bits-como-funciona/.

[50] J. A. Flores Grajales, «Protocolo para el cifrado autenticado de datos aplicado a un Centro de Mando y Control Aerotransportado,» Diciembre 2020. [En línea]. Available: https://inaoe.repositorioinstitucional.mx/jspui/bitstream/1009/2159/1/Julio%20Alberto%20Grajales%20Flores_Tesis%20Protocolo%20Criptogr%C3%A1fico%20%28No%20Firmada%29.pdf.

[51] R. Smedinga y M. Biehl, «Ejemplo de cifrado y descifrado de Java ChaCha20 | Cifrado simétrico,» 2020. [En línea]. Available: https://research.rug.nl/files/124739867/proceedings_2020.pdf.

[52] C. C. Mendoza y B. P. Vásquez, «Blowfish,» 2015 Junio 2015. [En línea]. Available: https://es.scribd.com/doc/271494982/Blowfish.

[53] D. D. M. Ruano, S. B. D. Valiente, T. L. F. Guevara y P. M. S. Pérez, «Performance analysis between Blowfish and AES symmetric algorithms,» 21 Diciembre 2021. [En línea]. Available: https://dialnet.unirioja.es/servlet/articulo?codigo=8901276.

[54] V. R. H. Zapata, «Implementación de OTP usando el protocolo Diffie Hellman basado en curvas elípticas (ECDH),» 2014. [En línea]. Available: http://bibliotecadigital.econ.uba.ar/download/tpos/1502-0551_ZapataValdezRH.pdf.

[55] S. G. Calderón, «Estudio y análisis de la seguridad del algoritmo HOTP como generador de contraseñas de un solo usos,» 2005. [En línea]. Available: https://repositorio.uniandes.edu.co/bitstream/handle/1992/10819/u263271.pdf?sequence=1.

[56] M. k. Youssouf, H. O. Siti, M. S. Maheyzah y N. Herve, «Comparative Study Of AES, Blowfish, CAST-128 And DES Encryption Algorithm,» IOSR Journal of Engineering (IOSRJEN), vol. 06, nº PP 01-07, p. 2, 2016.

[57] C. Adams, «El algoritmo de cifrado CAST-128,» 2 Marzo 2013. [En línea]. Available: https://datatracker-ietf-org.translate.goog/doc/rfc2144/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=sc.

[58] K. Aggarwal, J. K. Saini y H. K. Verma, «Performance Evaluation of RC6, Blowfish, DES, IDEA, CAST-128 Block Ciphers,» International Journal of Computer Applications (0975 – 8887), vol. 68, nº 25, p. 12, 2013.

[59] P. N. S. Valverde y T. I. F. Farez, «Análisis y descripción de la gestión de la seguridad en ambientes UMTS y desarrollo de herramienta didáctica,» 2014. [En línea]. Available: https://www.dspace.espol.edu.ec/bitstream/123456789/41672/1/D-84456.pdf.

[60] B. S. J. Sierra y Q. J. Aguilar, «Diseño e implementación de un módulo de seguridad hardware (HSM) sobre microcontrolador basado en el algoritmo HIGHT,» Diciembre 2017. [En línea]. Available: https://repository.udistrital.edu.co/handle/11349/7199?show=full.

[61] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Morai, J. Nakajima y T. Tokita, «Specification of Camellia – a 128-bit Block Cipher,» 26 Septiembre 2001. [En línea]. Available: https://www.cryptrec.go.jp/en/cryptrec_03_spec_cypherlist_files/PDF/06_01espec.pdf.

[62] R. Chakraborty, «Analysis of SEED,» 24 Marzo 2003. [En línea]. Available: https://www.cryptrec.go.jp/exreport/cryptrec-ex-1041-2001.pdf.

[63] J. Lopez, «Así funciona el sistema de cifrado AES-256 bits, ¿es realmente

seguro?,» 06 Mayo 2022. [En línea]. Available: https://hardzone.es/tutoriales/rendimiento/cifrado-aes-256-bits-como-funciona/.

[64] Domingo, «Modos de cifrado: ECB, CBC, CTR, OFB y CFB.,» 22 Julio 2007. [En línea]. Available: https://dlerch.blogspot.com/2007/07/modos-de-cifrado-ecb-cbc-ctr-ofb-y-cfb.html.

[65] K. M. G. Storvestre, «TRUST NO ONE; Homomorphic Encryption and its Applications,» Febrero 2023. [En línea]. Available: https://scholar.google.es/scholar?hl=es&as_sdt=0%2C5&q=TRUSTNOONE%3B+HomomorphicEncryptionandits+Applications&btnG=.

[66] G. Ionos, «El encriptado informático: así se protege la comunicación,» 02 Agosto 2021. [En línea]. Available: https://www.ionos.es/digitalguide/servidores/seguridad/todo-sobre-los-metodos-de-encriptado/.

[67] I. H. Latif, «Time Evaluation Of Different Cryptography Algorithms Using Labview,» Journal of Physics: Conference Series, pp. 2-4, 2020.

[68] F. E. A. Estrada, «Rompimiento del algoritmo DES utilizando fuerza bruta y programación paralela,» Noviembre 2021. [En línea]. Available: https://www.repositoriodigital.ipn.mx/bitstream/123456789/12275/1/Tesis%20Fabiola%20Aguilar.pdf.

[69] J. Lake, «¿What is 3DES encryption and how does DES work?,» 17 Febrero 2022. [En línea]. Available: https://www.comparitech.com/blog/information-security/3des-encryption/.

[70] H. I. Murillo, «Triple DES-96,» Noviembre 2014. [En línea]. Available: https://tesis.ipn.mx/bitstream/handle/123456789/20143/Triple%20DES-96.pdf?sequence=1&isAllowed=y.

[71] C. Kariuki, «La encriptación simétrica explicada en 5 minutos o menos,» 24 Septiembre 2023. [En línea]. Available: https://geekflare.com/es/symmetric-encryption/.

[72] L. Alegsa, «RC2,» 20 Diciembre 2021. [En línea]. Available: https://es.alegsaonline.com/art/81418.

[73] Ionos, «¿Cómo funcionan las claves RSA?,» 01 Marzo 2022. [En línea]. Available: https://www.ionos.es/digitalguide/servidores/seguridad/claves-rsa/.

[74] J. F. A. Pita, «Fundamentos criptográficos del algoritmo RSA,» 2018. [En línea]. Available: http://ri.uagro.mx/bitstream/handle/uagro/776/OK15158773_maestria.pdf?sequence=1&isAllowed=y.

[75] P. Yuliartanto, «Build an RSA Asymmetric Cryptography Generator in Go,» 28 Enero 2020. [En línea]. Available: https://betterprogramming.pub/build-an-rsa-asymmetric-cryptography-generator-in-go-d202b18bcfd0.

[76] A. Lopez, «Todo sobre criptografía: Algoritmos de clave simétrica y asimétrica,» 16 Marzo 2023. [En línea]. Available: https://www.redeszone.net/tutoriales/seguridad/criptografia-algoritmos-clave-simetrica-asimetrica/#373279-dsa.

[77] Simplilearn, «Algoritmo de firma digital (DSA) en criptografía: cómo funciona y ventajas,» 23 Agosto 2022. [En línea]. Available: https://www.simplilearn.com/tutorials/cryptography-tutorial/digital-signature-algorithm.

[78] F. Haopeng, W. Wenhao y Yongjuan, «Cache attack on recursive structure of MISTY1,» 2022 IEEE 6th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), pp. 4-8, 2022.

[79] R. Mahantesh y M. Sasmita, «Design of Secured Block Ciphers PRESENT and

HIGHT Algorithms and its FPGA Implementation,» 2018 Segunda Conferencia Internacional sobre Computación Inteligente y Sistemas de Control (ICICCS), pp. 1-7, 2018.

[80] C. M. H. Islam, R. Ewetz, A. Awad y F. Yao, «Seeds of SEED: R-SAW: New Side Channels Exploiting Read Asymmetry in MLC Phase Change Memories,» 2021 International Symposium on Secure and Private Execution Environment Design (SEED), pp. 2-5, 2021 .

[81] E. H. Rachmawanto, L. B. Handoko, C. Umam, C. Jatmoko y R. R. Ali, «Triple DES Cryptography Based on Hash Function and DSA for Digital Certificate Authentication,» 2022 International Seminar on Application for Technology of Information and Communication (iSemantic), pp. 5-6, 2022.

[82] B. Kitchenham y u. S. Charters, «Guidelines for performing Systematic Literature Reviews in Software Engineering,» 9 Julio 2007. [En línea]. Available: https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf.

[83] A. Ibrahim, A. Chefranov y R. Hamamreh, «Ciphertext-only attack on RSA using lattice basis reduction,» The International Arab Journal of Information Technology, vol. 18, nº 2, pp. 237-247, 2021.

[84] M. Babenko, N. Chervyakov, A. Tchernykh, N. Kucherov, M. Deryabin, G. Radchenko, P. O. A. Navaux y V. Svyatkin, «Security analysis of homomorphic encryption scheme for cloud computing: Known-plaintext attack,» 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), pp. 270-274, 2018.

[85] J. Rao y Z. Cui, «Chosen Plaintext Combined Attack against SM4 Algorithm,» Applied Sciences, vol. 12, nº 18, p. 9349, 2022.

[86] J. Hermelink, P. Pessl y T. Poppelmann, «Fault-Enabled Chosen-Ciphertext Attacks on Kyber,» 22nd International Conference on Cryptology in India, pp. 311-334, 2021.

[87] M. D. O. H. Hossain, F. Doudou y Y. Kadobayashi, «SSH and FTP brute-force Attacks Detection in Computer Networks: LSTM and Machine Learning Approaches,» 2020 5th International Conference on Computer and Communication Systems (ICCCS), pp. 491-497, 2020.

[88] J. M. Biju, N. Gopal y A. J. Prakash, «Cyber attacks and its different types,» International Research Journal of Engineering and Technology (IRJET), vol. 6, nº 3, pp. 4849-4852, 2019.

[89] Laatansa, R. Saputra y N. B., «Analysis of GPGPU-Based Brute-Force and Dictionary Attack on SHA-1 Password Hash,» 2019 3rd International Conference on Informatics and Computational Sciences (ICICoS), pp. 1-4, 2019.

[90] F. Aliyua, S. T. y S. E. M. , «A Detection and Prevention Technique for Man in the Middle Attack in Fog Computing,» The 9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2018), vol. 141, pp. 24-31, 2018.

[91] S. Datta, «¿Cómo funciona el ataque Meet-in-the-Middle?,» 3 Mayo 2023. [En línea]. Available: https://www.baeldung.com/cs/security-meet-in-the-middle-attack.

[92] S. Faust, J. Kramer, M. Orlt y P. Struck, «On the Related-Key Attack Security of Authenticated Encryption Schemes,» International Conference on Security and Cryptography for Networks, vol. 13409, p. 362–386, 2022.

[93] T. I. Romero, «Side-channel attacks: ¿qué son?,» 11 Diciembre 2022. [En línea]. Available: https://www.profesionalreview.com/2022/12/11/side-channel-

attacks/.

[94] L. Scripcariu, F. Diaconu, P. D. Matasaru y L. Gafencu, «AES Vulnerabilities Study,» 2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), pp. 1-5, 2018.

[95] J. Park, H. Kim y J. Kim, «Improved See-In-The-Middle Attacks on AES,» International Conference on Information Security and Cryptology, vol. 13218, p. 271–279, 2021.

[96] Z. Najm, D. Jap, B. Jungk, S. Picek y S. Bhasin, «On comparing side-channel properties of AES and ChaCha20 on microcontrollers,» In 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), pp. 552-555, 2018.

[97] R. Li y C. Jin, «Meet-in-the-middle attacks on 10-round AES-256,» Designs, Codes and cryptography, vol. 80, nº 3, pp. 459-471, 2016.

[98] Y. Zhong y U. Guin, «Fault-Injection Based Chosen-Plaintext Attacks on Multicycle AES Implementations,» Proceedings of the Great Lakes Symposium on VLSI 2022, pp. 443-448, 2022.

[99] B. Weger y B. Skoric, «Known-plaintext attack on AES, using the HHL algorithm and the Macaulay approach.,» 22 Julio 2022. [En línea]. Available: https://pure.tue.nl/ws/portalfiles/portal/288886527/Stolwijk_Y.pdf.

[100] A. Puthuparambil y J. Thomas, «Freestyle, a randomized version of ChaCha for resisting offline brute-force and dictionary attacks,» Journal of Information Security and Applications, vol. 49, nº 2019, pp. 1-33, 2019.

[101] A. K. Sharma y S. K. Mittal, «Cryptography & Network Security Hash Function Applications, Attacks and Advances: A Review,» 2019 Third International Conference on Inventive Systems and Control (ICISC), pp. 177-188, 2019.

[102] K. Bhuvanagiri, «Evaluation of XChaCha20-Poly1305 for Improved File System Level Encryption in the Cloud,» National College of Ireland, Dublín,

2021.

[103] S. Ezadeen y A. Alwattar, «Survey of Blowfish Algorithm for Cloud,» Technium: Romanian Journal of Applied Sciences and Technology, vol. 4, nº 6, pp. 18-28, 2022.

[104] S. Rao y K. Jyothi, «Secret Key Generation using Genetic Algorithm for the Hybrid Blowfish Encryption and Substitution Ciphers,» de 2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, 2022.

[105] N. Oishi y A. Mahamud, «Enhancing Wi-Fi security using a hybrid algorithm of blowfish and RC6,» de 2016 International Conference on Networking Systems and Security (NSysS), Dhaka, 2016.

[106] M. K. S. Alwazzeh y Shamma, «Man in The Middle Attacks Against SSL/TLS: Mitigation and Defeat,» Journal of Cyber Security and Mobility, vol. 9, pp. 449-468, 2020.

[107] S. Ahmad, M. Rizwan, J. Ahmad y N. Barua, «Meet In The Middle Attack: A Cryptanalysis Approach,» International Journal of Computer Applications, vol. 975, nº 1, pp. 1-7, 2010.

[108] S. G. Gui, «Introducción a la seguridad de la información,» 2018. [En línea]. Available: https://openaccess.uoc.edu/bitstream/10609/142807/1/M%C3%B3dulo%201_Introducci%C3%B3n%20a%20la%20seguridad%20de%20la%20informaci%C3%B3n.pdf.

[109] B. Gómez, «AES-256 ¿Qué es? ¿Cómo funciona? (Mejor explicación),» 18 Abril 2021. [En línea]. Available: https://www.profesionalreview.com/2021/04/18/aes-256/.

ANEXOS

Universidad Señor de Sipán

ACTA DE REVISIÓN DE ASESORÍA

Yo **Minguillo Rubio Cesar Augusto**. quien suscribe como asesor designado mediante Resolución de Facultad N°0337-2023/FIAU-USS, del proyecto de investigación titulado **EVALUACIÓN DE ALGORITMOS CRIPTOGRÁFICOS PARA MEJORAR LA SEGURIDAD DE LA INFORMACIÓN EN EL ENVÍO DE TEXTO PLANO POR INTERNET**., desarrollado por el(los) estudiante(s): **Guerrero Vega Ericks Tito, Lozano Delgado Kelkin Heiminn**, del programa de estudios de **Ingeniería de Sistemas**, acredito haber revisado, realizado observaciones y recomendaciones pertinentes tal como se detalla en el siguiente cuadro:

Fecha de revisión:	Modalidad de Asesoría:	Medio de Asesoría:	Veredicto de Asesoría:
25/05/2023 01/06/2023 08/06/2023	Presencial	Laboratorio 11	Aprobado

En virtud de lo antes mencionado, firman:

(**Asesor**) **Minguillo Rubio Cesar Augusto**	DNI: 16787173	
(**Autor 1**) **Guerrero Vega Ericks Tito**	DNI: 46659948	

(**Autor 2**) **Lozano Delgado Kelkin Heiminn**	DNI: 74751359	

Pimentel, 11 de junio de 2023

FICHA DE REVISIÓN				
N°	**Sección del Informe observado** *(Seleccione una opción)*	**Número de página observado**	**Comentario de la observación**	**Estado de la Observación (Seleccione una opción)**
1.	Objetivo 01	55 - 67	Cambiar al formato a IEEE Las figuras y tablas.	- Corregido
2.	Objetivo 02	68 - 72	Redacción del objetivo en el contenido y cambiar al formato a IEEE Las figuras y tablas.	- Corregido
3.				
4.				
5.				

ANEXO 01: DECLARACIÓN JURADA DE ORIGINALIDAD

Quien(es) suscribe(n) la DECLARACIÓN JURADA, soy(somos) **Guerrero Vega Ericks Tito, Lozano Delgado Kelkin Heiminn.** del Programa de Estudios de **Ingeniería de Sistemas** de la Universidad Señor de Sipán S.A.C, declaro (amos) bajo juramento que soy (somos) autor(es) del trabajo titulado:

EVALUACIÓN DE ALGORITMOS CRIPTOGRÁFICOS PARA MEJORAR LA SEGURIDAD DE LA INFORMACIÓN EN EL ENVÍO DE TEXTO PLANO POR INTERNET

El texto de mi trabajo de investigación responde y respeta lo indicado en el Código de Ética del Comité Institucional de Ética en Investigación de la Universidad Señor de Sipán, conforme a los principios y lineamientos detallados en dicho documento, en relación con las citas y referencias bibliográficas, respetando el derecho de propiedad intelectual, por lo cual informo que la investigación cumple con ser inédito, original y autentico.

En virtud de lo antes mencionado, firman:

Guerrero Vega Ericks Tito	DNI: 46659948	
Lozano Delgado Kelkin Heiminn	DNI: 74751359	

Pimentel, 19 de Julio de 2023.

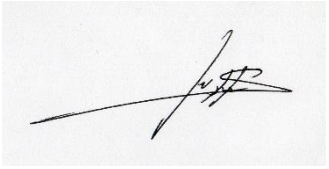
ANEXO 02: ACTA DE REVISIÓN DE SIMILITUD DE LA INVESTIGACIÓN

Yo **Mejia Cabrera Heber Ivan** docente del curso de **Investigación II** del Programa de Estudios de **Ingeniería de Sistemas** y revisor de la investigación del (los) estudiante(s), **Guerrero Vega Ericks Tito, Lozano Delgado Kelkin Heiminn**, titulada:

EVALUACIÓN DE ALGORITMOS CRIPTOGRÁFICOS PARA MEJORAR LA SEGURIDAD DE LA INFORMACIÓN EN EL ENVÍO DE TEXTO PLANO POR INTERNET

Se deja constancia que la investigación antes indicada tiene un índice de similitud del **16%**, verificable en el reporte final del análisis de originalidad mediante el software de similitud TURNITIN. Por lo que se concluye que cada una de las coincidencias detectadas no constituyen plagio y cumple con lo establecido en la Directiva sobre índice de similitud de los productos académicos y de investigación en la Universidad Señor de Sipán S.A.C., aprobada mediante Resolución de Directorio N° 145-2022/PD-USS.

En virtud de lo antes mencionado, firma:

Mejia Cabrera Heber Ivan	DNI: 41639565	

Pimentel, 19 de Julio de 2023.

ANEXO 03: ACTA DE APROBACIÓN DEL ASESOR

Yo **Minguillo Rubio Cesar Augusto**. quien suscribe como asesor designado mediante Resolución de Facultad N° **0366-2023/FIAU-USS**, del proyecto de investigación titulado **Evaluación de algoritmos criptográficos para mejorar la seguridad de la información en el envío de texto plano por internet**., desarrollado por el(los) estudiante(s): **Guerrero Vega Ericks Tito, Lozano Delgado Kelkin Heiminn**., del programa de estudios de **Ingeniería de Sistemas**, acredito haber revisado, realizado observaciones y recomendaciones pertinentes, encontrándose expedito para su revisión por parte del docente del curso.

En virtud de lo antes mencionado, firman:

(**Asesor**) Minguillo Rubio Cesar Augusto	DNI: 16787173	
(**Autor 1**) Guerrero Vega Ericks Tito	DNI: 46659948	
(**Autor 2**) Lozano Delgado Kelkin Heiminn	DNI: 74751359	

Pimentel, 21 de Julio de 2023

CUESTIONARIO SUS (System Usability Scale)

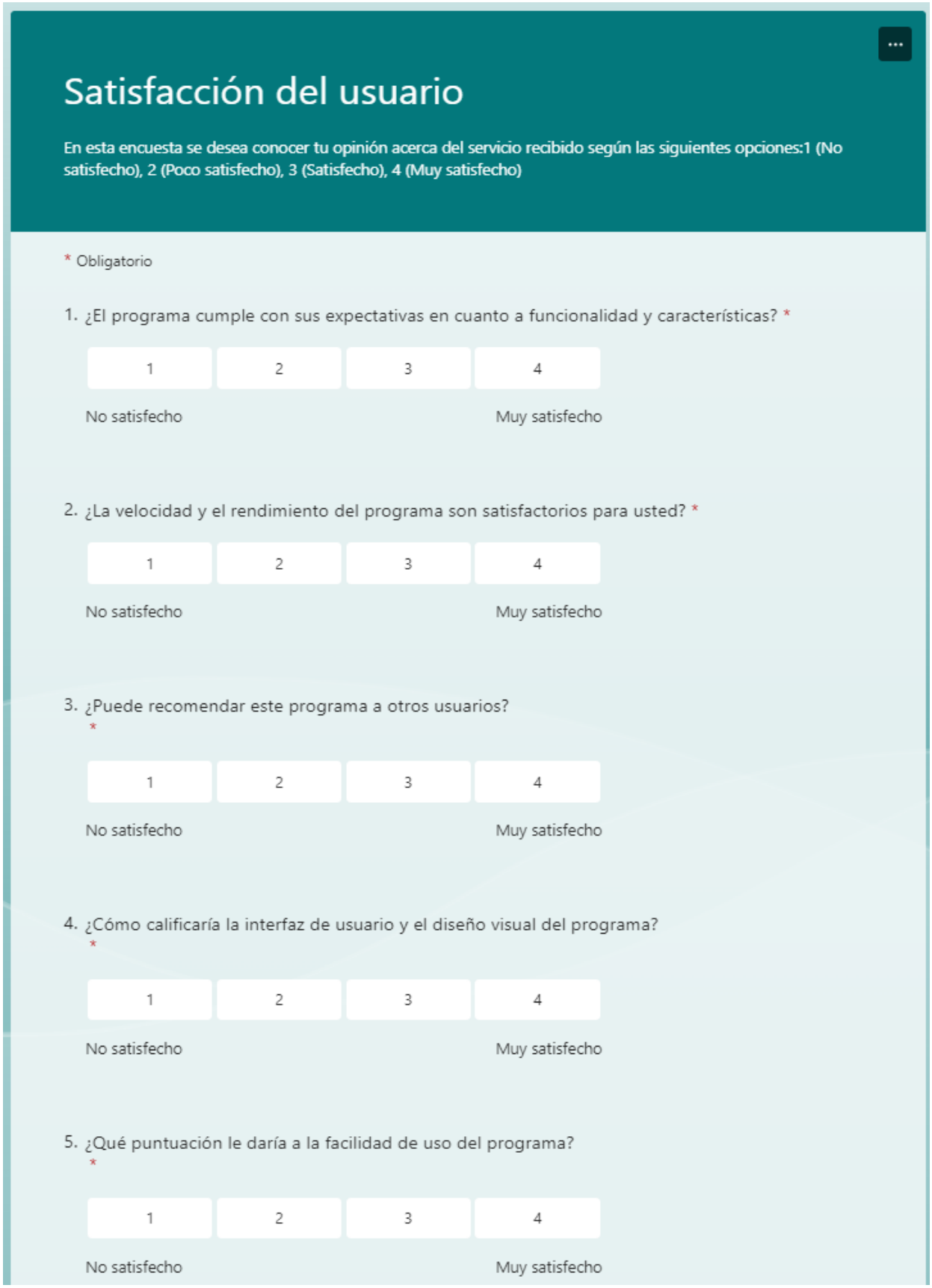


Fig. 39. Cuestionario de Satisfacción de Usuario del 1 al 5.

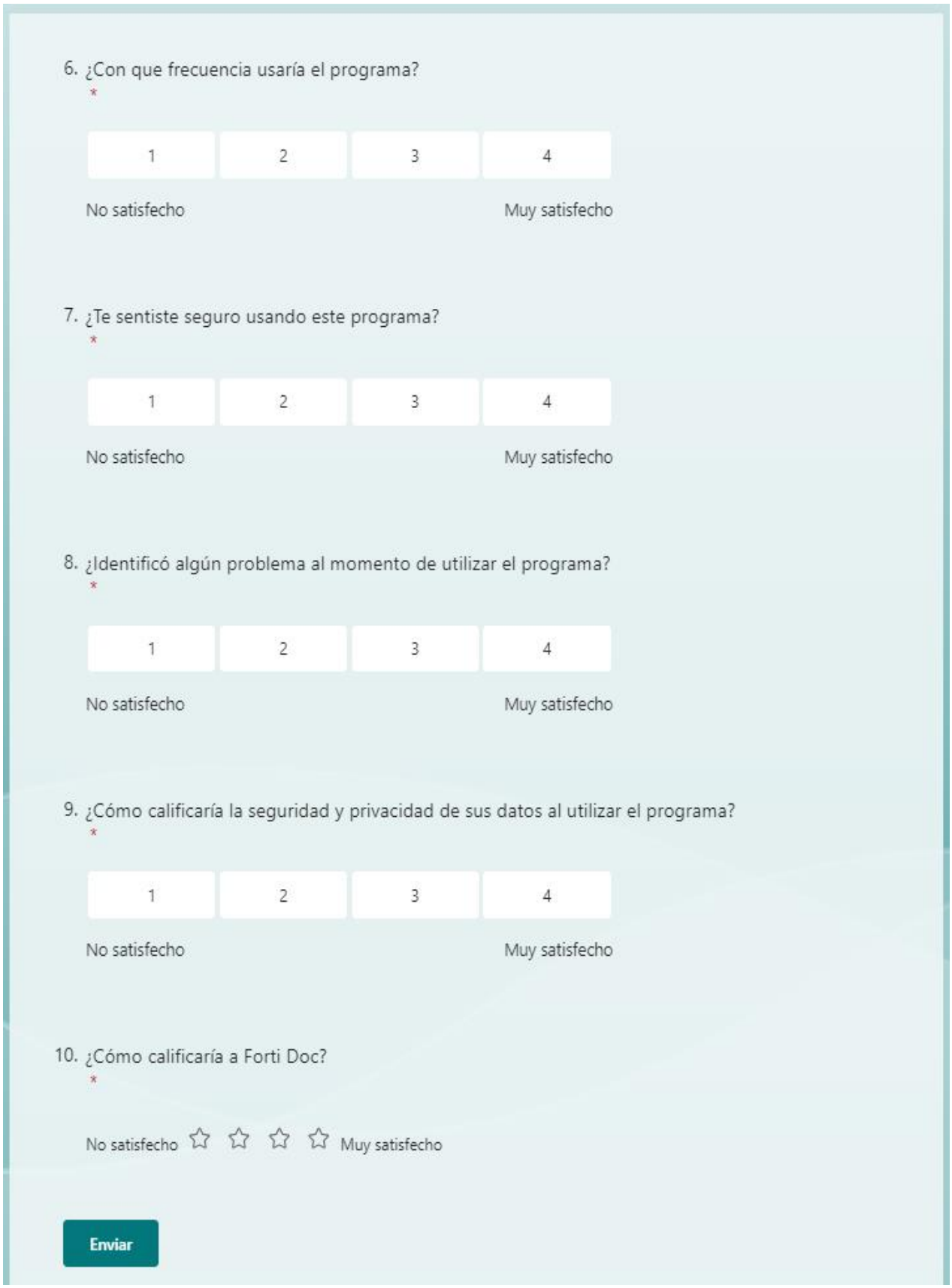


Fig. 40. Cuestionario de Satisfacción de Usuario del 6 al 10.

Matriz de Consistencia Lógica

MATRÍZ DE CONSISTENCIA LÓGICA DE PROYECTO DE INVESTIGACIÓN

Enfoque metodológico

Titulo	EVALUACIÓN DE ALGORITMOS CRIPTOGRÁFICOS PARA MEJORAR LA SEGURIDAD DE LA INFORMACIÓN EN EL ENVÍO DE TEXTO PLANO POR INTERNET

Tipo de investigación	Problema	Variables	Indicadores	Población	Muestra	Método de recolección de Datos	Técnicas de procesamiento de datos
Cuantitativa Tecnológica Aplicada	¿Cómo asegurar los datos de un texto plano transmitida por la red pública de internet?	**Variable Independiente**: Algoritmos criptográficos. **Variable Dependiente**: Seguridad de la información en el envío de texto plano por internet	Calidad del algoritmo -Rendimiento de cifrado -Rendimiento de descifrado -Fortaleza del algoritmo Seguridad de la información -Confidencialidad -Integridad	La población seleccionada para esta investigación fueron aquellos algoritmos criptográficos empleados en el envío en texto plano por internet, y que han sido identificados en la literatura científica, los cuales son quince (15) algoritmos criptográficos.	La muestra seleccionada para esta investigación fue seleccionada mediante muestreo no probabilístico, y fueron aquellos algoritmos criptográficos empleados en el envío en texto plano por internet que, según la literatura científica, han obtenido mejores resultados respecto a rendimiento de cifrado y rendimiento de descifrado, los cuales son tres (03) Blowfish, AES-256 y Chacha20	Registros electrónicos. Bibliografías.	Análisis electrónico Análisis estadísticos

Diseño de investigación	Hipótesis	Objetivo General	Objetivos específicos	Método propuesto y desarrollado	Resultados preliminares
Línea de investigación (Infraestructura Tecnológica y Medio Ambiente), en el área prioritaria de (Ingeniería del Software).	Mediante la implementación de la criptografía permitirá asegurar los datos de un texto plano transmitido por la red pública de internet	Evaluar algoritmos criptográficos para mejorar la seguridad de la información del envío en texto plano por internet	**OBJ_01**: Seleccionar los algoritmos criptográficos más relevantes que garanticen la seguridad de los datos. **OBJ_02**: Identificar los principales ataques conocidos contra los algoritmos criptográficos seleccionados previamente. **OBJ_03**: Desarrollar en lenguaje de programación los algoritmos criptográficos para cifrar los datos de un texto plano. **OBJ_04**: Proponer recomendaciones precisas que promuevan la ejecución segura y medir la satisfacción de usuario.	1. Seleccionar los algoritmos criptográficos más relevantes que garanticen la seguridad de los datos. 1.1. En la literatura científica se adoptaron las directrices para la revisión sistemática que propusieron Kitchenham & Charters con el propósito de seleccionar los algoritmos criptográficos. 2. Identificar los principales ataques conocidos contra los algoritmos criptográficos seleccionados previamente. 2.1. se ejecutó una revisión de artículos que pusieran de manifiesto dichos ataques para cada uno de los tres algoritmos seleccionados en el paso previo: Blowfish, AES-256 y Chacha20 . 3. Desarrollar en lenguaje de programación los algoritmos criptográficos para cifrar los datos de un texto plano. 3.1. Se llevaron a cabo las implementaciones de los algoritmos AES-256, ChaCha20 y Blowfish de acuerdo a sus respectivas especificaciones. Para ello, se utilizó el lenguaje de programación Python. 4. Proponer recomendaciones precisas que promuevan la ejecución segura y medir la satisfacción de usuario. 4.1. Se realiza recomendaciones qu esirva para el usuario registre de manera adecuada su información y tambien tiene la opcion de calificarnos.	Encriptación de datos y seguridad de la información

Evidencias de ejecución

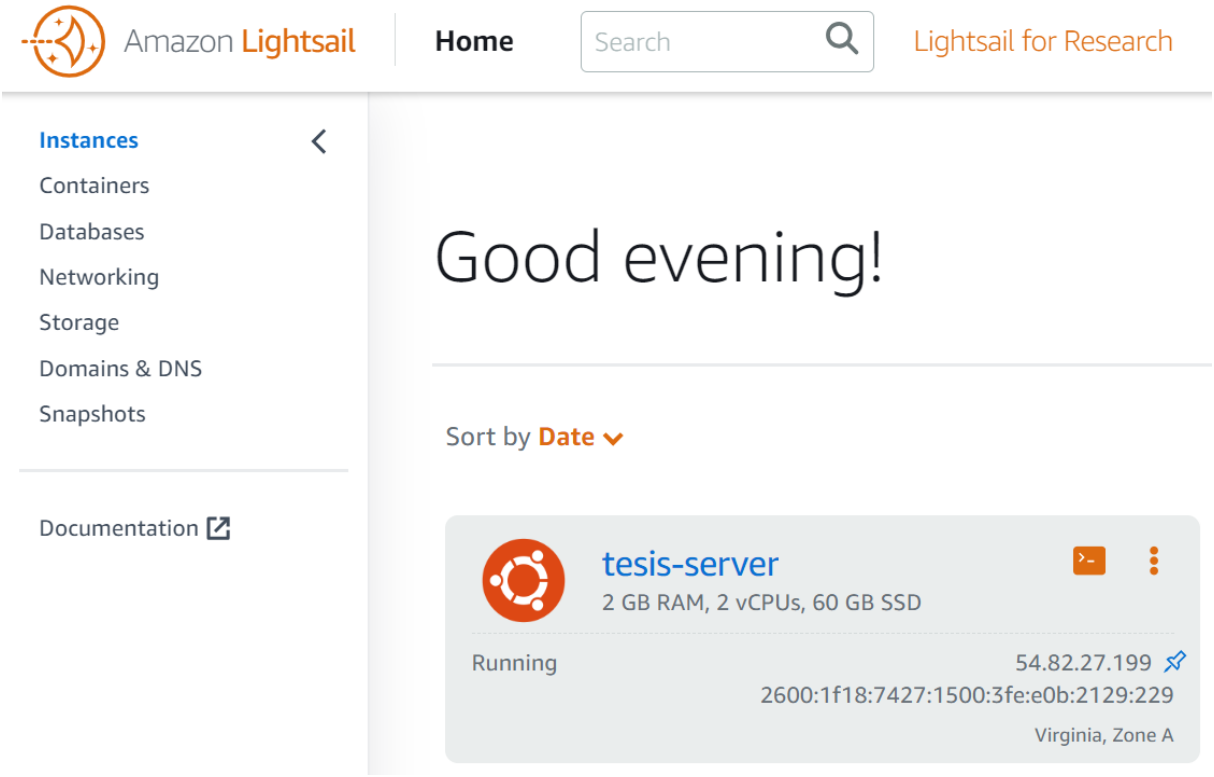


Fig. 41. Servidor tesis-server.

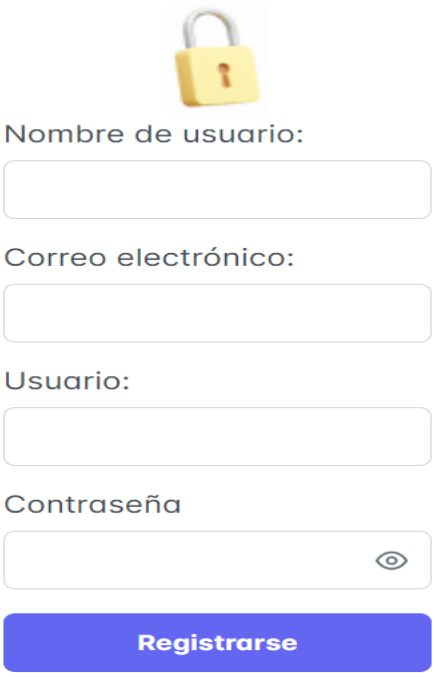


Fig. 42. Registro de usuario.

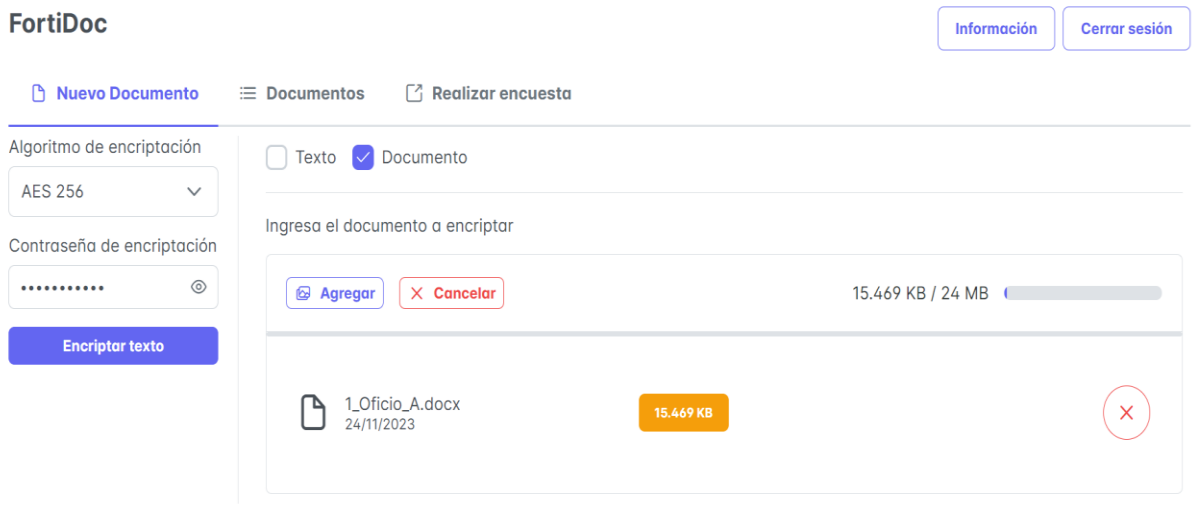


Fig. 43. Seleccionar tipo de algoritmo y cargar documento.

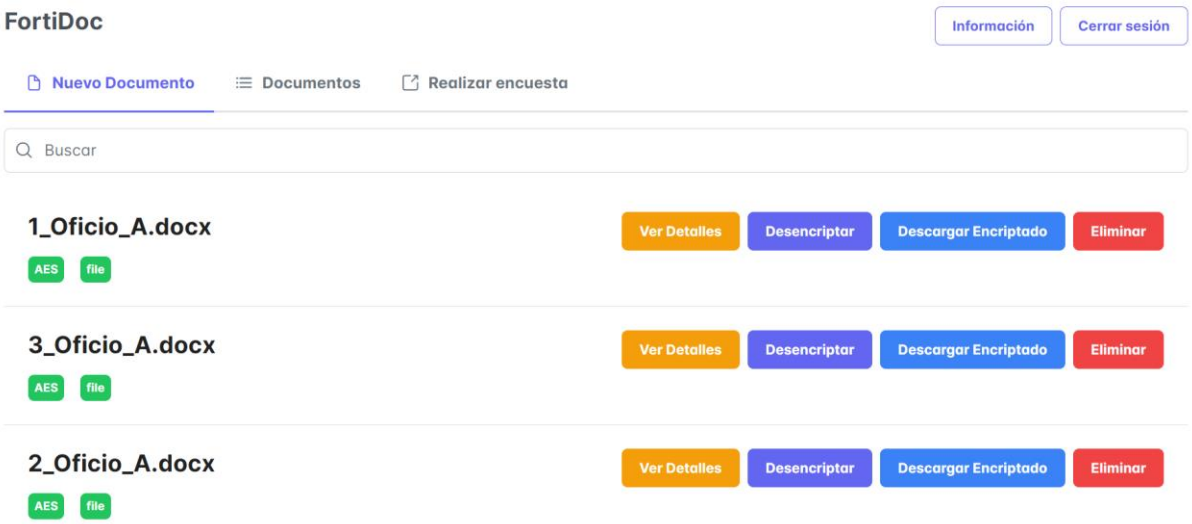


Fig. 44. Lista de documentos encriptados.

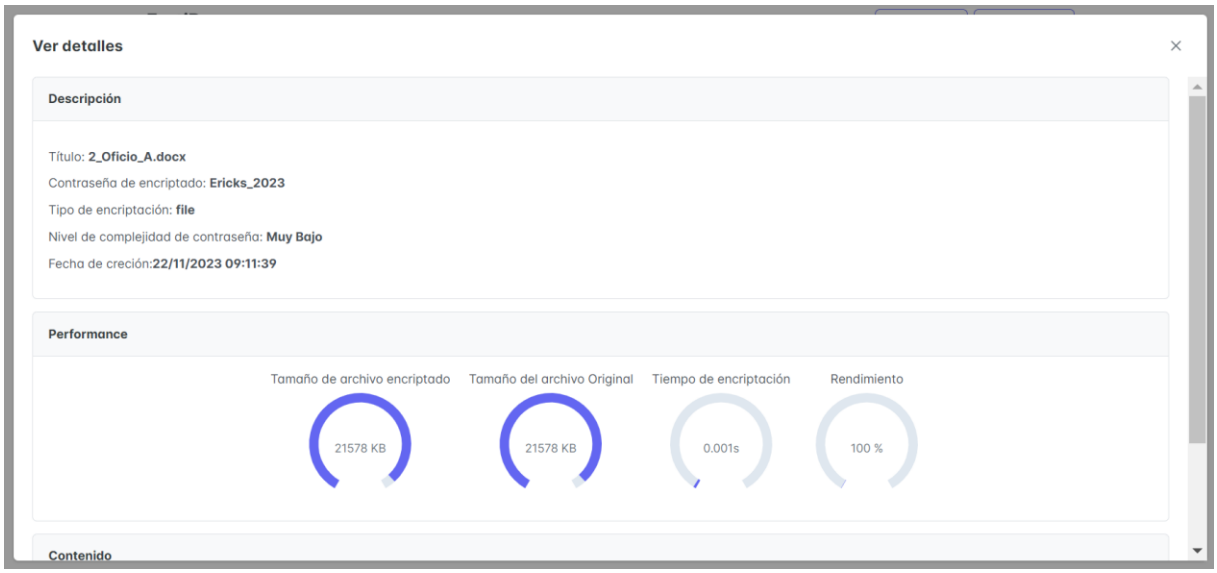


Fig. 45. Detalles del documento encriptado.

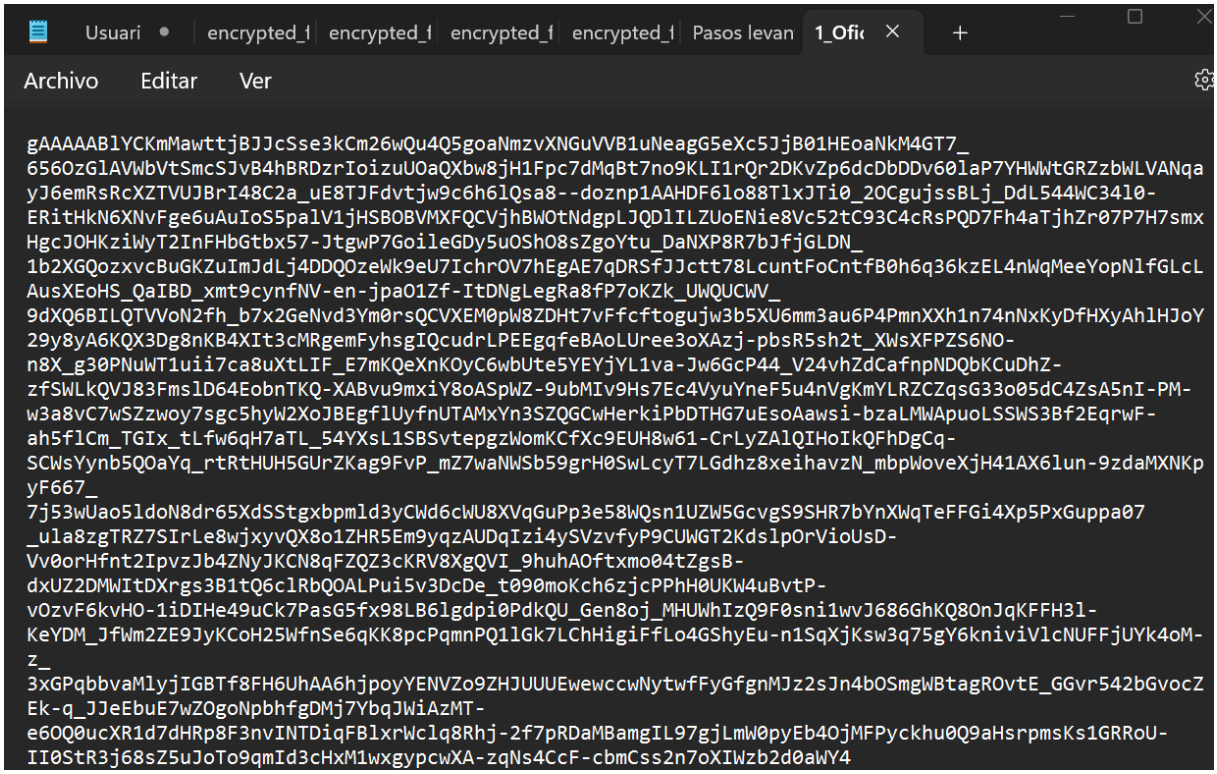


Fig. 46. Documento encriptado con el algoritmo AES-256.

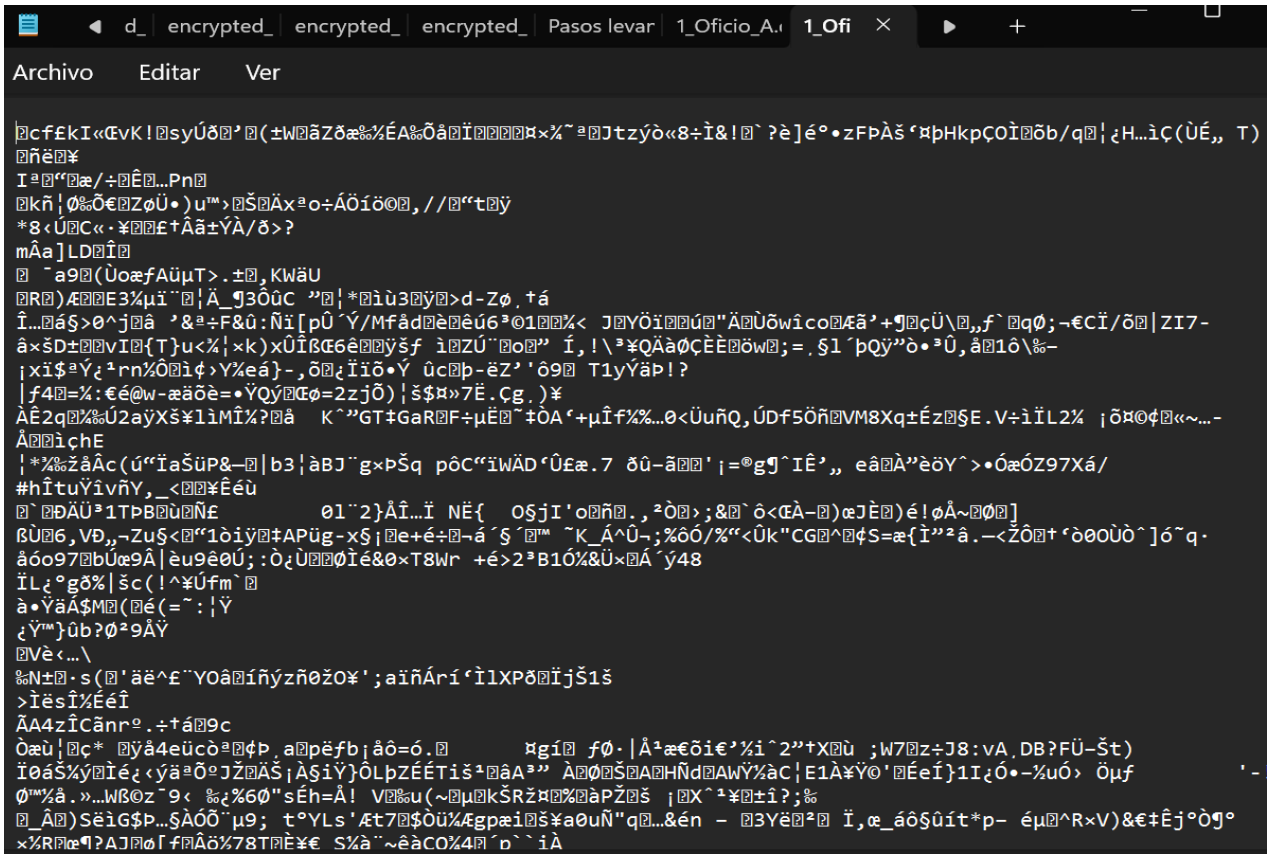


Fig. 47. Documento encriptado con el algoritmo Blowfish.

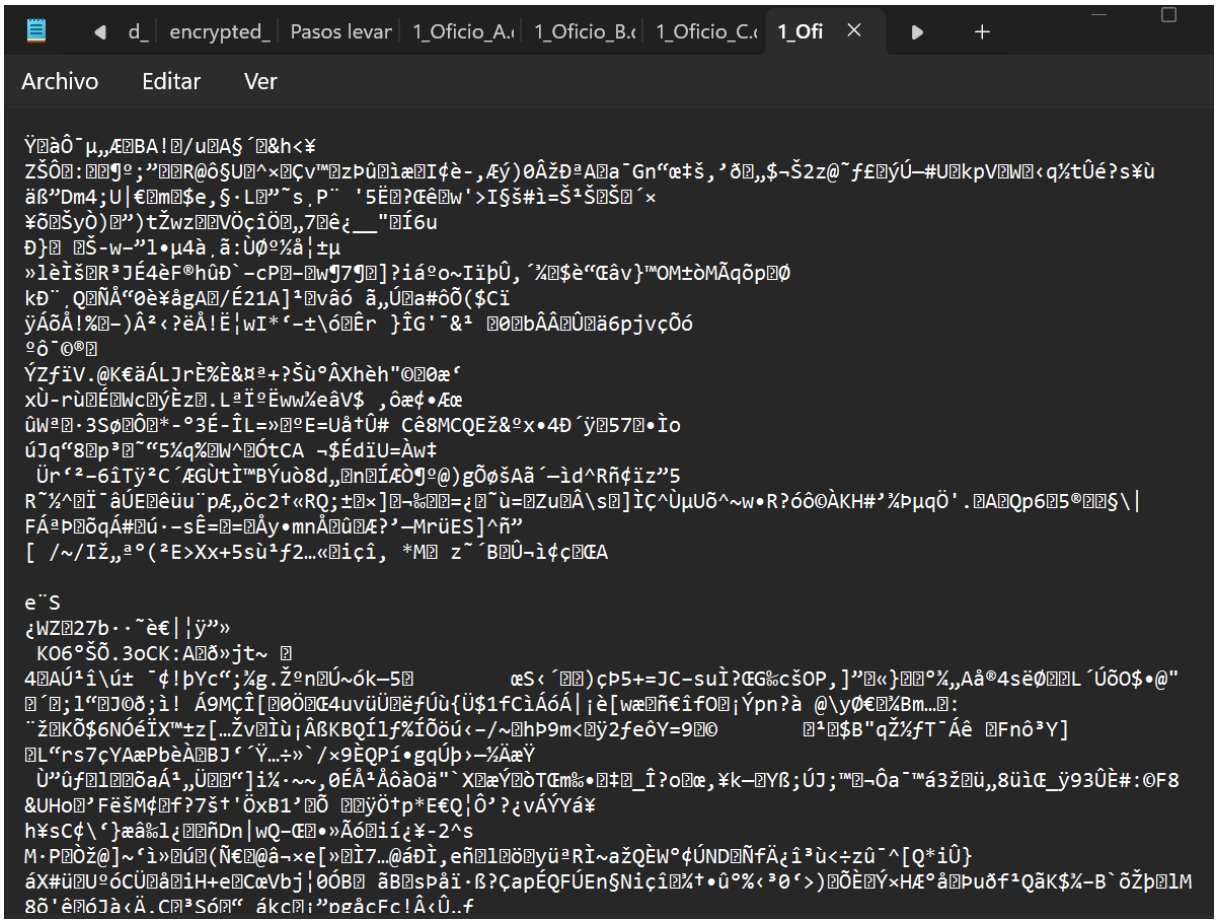


Fig. 48. Documento encriptado con el algoritmo ChaCha20.

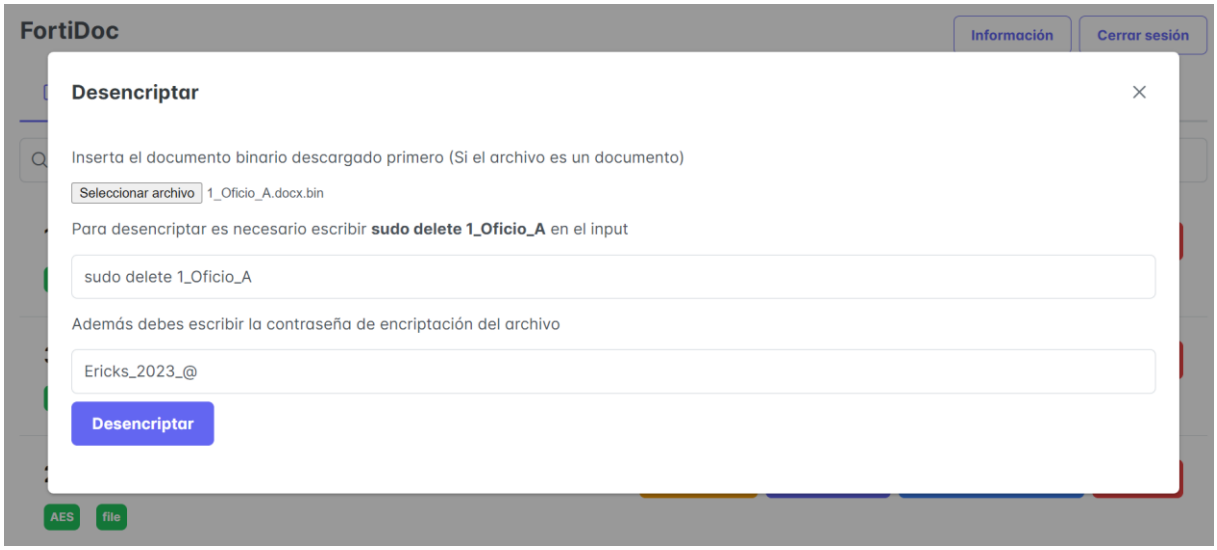


Fig. 49. Proceso de desencriptado.

Cutervo, 28 de enero del 2022

OFICIO N.° 001 - 2022-GR.CAJ-GSR.C/SGO

Señor:
Raul Pinedo Vasquez
Alcalde provincial

Presente. -

ASUNTO : PROPUESTA ECONOMICA ALQUILER DE TIENDAS DEL CENTRO COMERCIAL

Es grato dirigirnos a Usted todos los **Comerciantes Ambulantes** en forma agrupada del jirón progreso cuadra 1 y el jirón la Merced cuadra 9 de rubros confecciones y zapatería le expresamos nuestro cordial y afectuoso saludo y por su intermedio a su equipo de regidores seguidamente para informar que somos conocedores de la realidad socioeconómica de nuestra provincia, preocupados por el funcionamiento del **CENTRO COMERCIAL** nos dirigimos a su digna persona con el único fin de pedirle que tome en cuenta nuestra propuesta realista y que todos estamos de acuerdo que tengamos la oportunidad de adquirir un lugar donde se pueda realizar nuestras actividades comerciales, en forma de alquiler conocedores de su alto espíritu de progreso y comprensión como comerciantes no dudamos que tomara en cuenta nuestra propuesta.

Agradeciendo su atención al presente, hacemos propicia la oportunidad para expresarle las muestras de nuestra especial consideración y estima.

Atentamente,
los comerciantes

Fig. 50. Documento desencriptado.

Reporte de similitud

NOMBRE DEL TRABAJO
LOZANO-DELGADO--GUERRERO-VEGA_TURNITING.docx

AUTOR
Lozano Guerrero

RECUENTO DE PALABRAS
20997 Words

RECUENTO DE CARACTERES
114686 Characters

RECUENTO DE PÁGINAS
94 Pages

TAMAÑO DEL ARCHIVO
1.4MB

FECHA DE ENTREGA
Jan 12, 2024 9:16 AM GMT-5

FECHA DEL INFORME
Jan 12, 2024 9:17 AM GMT-5

11% de similitud general
El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base c

- 9% Base de datos de Internet
- 2% Base de datos de publicaciones
- Base de datos de Crossref
- Base de datos de contenido publicado de Crossr
- 7% Base de datos de trabajos entregados

Excluir del Reporte de Similitud

- Material bibliográfico
- Material citado
- Coincidencia baja (menos de 8 palabras)