UNIVERSIDAD SEÑOR DE SIPÁN

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

Evaluación de técnicas de cifrado para el intercambio de datos en internet de las cosas en el ámbito de la salud

PARA OPTAR TÍTULO PROFESIONAL DE INGENIERA DE SISTEMAS

Autor(a) (es):

Bach. Niño Moreno Najhely Yamilett

https://orcid.org/0000-0001-8560-867X

Bach. Rimarachin Escribano Neri Rut

https://orcid.org/0000-0002-5381-9189

Asesor(a):

Dr. Ramos Moscol, Mario Fernando

https://orcid.org/0000-0003-3812-7384

Línea de Investigación:

Infraestructura, Tecnología y Medio Ambiente

Pimentel – Perú 2023

EVALUACIÓN DE TÉCNICAS DE CIFRADO PARA EL INTERCAMBIO DE DATOS EN INTERNET DE LAS COSAS EN EL ÁMBITO DE LA SALUD

Aprobación del jurado

_____

Dr. Bravo Ruiz, Jaime Arturo
Presidente de Jurado de Tesis

_____

Dr. Vásquez Leyva Oliver
Secretario del Jurado de Tesis

_____

Dr. Tuesta Monteza, Victor Alexci
Vocal De Jurado de Tesis

DECLARACIÓN JURADA DE ORIGINALIDAD

Quien suscribe la **DECLARACIÓN JURADA**, soy Najhely Niño Moreno. del Programa de Estudios de la Universidad Señor de Sipán S.A.C, declaro bajo juramento que soy autor del trabajo titulado: **EVALUACIÓN DE TÉCNICAS DE CIFRADO PARA EL INTERCAMBIO DE DATOS EN INTERNET DE LAS COSAS EN EL ÁMBITO DE LA SALUD.**

El texto de mi trabajo de investigación responde y respeta lo indicado en el Código de Ética del Comité Institucional de Ética para la Investigación de la Universidad Señor de Sipán (CIEI USS) conforme a los principios y lineamientos detallados en dicho documento, en relación a las citas y referencias bibliográficas, respetando al 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:

Niño Moreno Najhely Yamilett	72212041	
Rimarachin Escribano Neri Rut	76002625	

Pimentel, 08 de 02 de 2023.

Dedicatorias

A mis padres Cecilia y Carlos, por ser el pilar fundamental, en mi formación personal y profesional, ya que han velado por mí durante este arduo camino estudiantil, asimismo, con su amor y apoyo incondicional, ha sido posible llegar a culminar con éxito mis estudios universitarios y metas trazadas.

A mi hermana Lissett, por compartir momentos significativos conmigo, por siempre escucharme y darme ánimo para continuar y no renunciar a mis aspiraciones.

Najhely

A mis padres Román y Ascención por ser mi sostén día a día y mi motivo constante para cumplir mis metas trazadas.

A mis hermanas Anggi y Kathy, que son parte importante en mi vida.

A mi abuela Clara, por siempre escucharme y darme ánimo para continuar y no renunciar a mis sueños.

Rut

Agradecimientos

A Dios, por protegerme durante todo mi camino, y haberme dado fuerza para superar los obstáculos a lo largo de mi vida universitaria, permitiéndome culminar con éxito mi carrera profesional.
A mis padres y hermana, quienes demostraron en todo momento su amor, corrigiendo mis faltas y celebrando mis triunfos.
A mi amiga y compañera de tesis, Rut, con quien no sólo aprendí, sino que también disfruté momentos inolvidables.
A mis profesores, por su labor tan valiosa, tiempo, paciencia, compromiso y enseñanzas, que me transmitieron en el desarrollo de mi formación profesional.

Najhely

A Dios y a la Virgen del Carmen, por protegerme en esta travesía.
A mi madre, que con su infinito amor ha sabido soportar mis momentos difíciles propios de la vida universitaria.
A mi abuela Clara por haberme incentivado a iniciar mi vida académica y estar presente a lo largo de ella.
A mi compañera de tesis, Najhely, en quien encontré mi mejor complemento académicamente.

Rut

Resumen

A nivel mundial, la cifra de cibernautas que al conectarse a Internet creció de 3.9 mil millones en el 2018 y crecerá en 2023 a 5.3 millones. Otro concepto valioso es que Internet de las cosas permitirá a las personas usar los servicios de conexión por medio de dispositivos IoT, en dónde la asistencia médica tendrá menor intensidad laboral y costo operacional logrando así la automatización de dispositivos médicos conectados y el manejo de teléfonos inteligentes, así como también el uso de software que aceptará realizar pruebas en pacientes de manera más rápida y precisa. La seguridad de estos datos es una tarea desafiante; sin embargo, se pueden mitigar con técnicas de cifrado de datos, siendo cruciales cuando se trata de la autenticación de usuarios y la privacidad de los datos. Quedando claro que, en el campo de ingeniería, se ha advertido que a lo largo de los años estás técnicas presentan problemas de deficiencia en el intercambio de datos, debido que con el auge de los dispositivos IoT y su gran interconexión genera un gran número de amenazas y ataques. Por lo que en esta investigación se realizó un proceso de verificación de artículos probados en el banco de datos IEE Xplore y ProQuest, para la elección de técnicas de cifrado, después se identificó los criterios de valoración de dichas técnicas de cifrado de datos en el cual se toman factores de evaluación que son de mucha importancia e influyentes en la ejecución de los algoritmos de cifrado, posteriormente se diseñó un escenario de pruebas para evaluar las técnicas de cifrado de datos seleccionado, diseñando un escenario de COVID-19 debido que con la pandemia se ha visto la dificultad de cómo proteger los datos de los trabajadores, después se aplicó las técnicas de cifrado simétrica (AES), asimétrica (SHA y RSA), por bloques(AES) e híbrida (SHA) y por último se estimó los resultados alcanzados. Obteniendo como resultado que, en el campo de la salud, la técnica más eficiente para el intercambio de datos es el cifrado Híbrido (algoritmo SHA 256), ya que, al intercambiar los datos, es el cifrado más eficiente en cuanto a tiempo de procesamiento total con un resultado de 0.00450016 segundos y rendimiento 0.276%, es decir que entre el tiempo de cifrado y descifrado es la que realizó la tarea más velozmente.

Palabras Claves: Internet de las Cosas, cifrado, intercambio de datos, privacidad

Abstract

Globally, the number of cybernauts connecting to the Internet grew from 3.9 billion in 2018 and will grow to 5.3 billion by 2023. Another valuable concept is that the Internet of Things will allow people to use connected services through IoT devices, where medical assistance will have less labor intensity and operational cost, thus achieving the automation of connected medical devices and the management of smartphones, as well as the use of software that will accept to perform tests on patients in a faster and more accurate way. The Internet of Things, on the other hand, is a domain in which data transfer takes place every second. The security of this data is a challenging task; however, it can be mitigated with data encryption techniques, being crucial when it comes to user authentication and data privacy. It is clear that, in the field of engineering, it has been noticed that over the years these techniques present problems of deficiency in the exchange of data, because with the rise of IoT devices and their great interconnection generates a large number of threats and attacks. Therefore, in this research a verification process of articles tested in the IEE Xplore and ProQuest databank was carried out, for the choice of encryption techniques, then the evaluation criteria of these data encryption techniques were identified in which evaluation factors that are of great importance and influential in the execution of encryption algorithms are taken, Afterwards, a test scenario was designed to evaluate the selected data encryption techniques, designing a COVID-19 scenario due to the fact that with the pandemic the difficulty of how to protect workers' data has been seen, then the symmetric (AES), asymmetric (SHA and RSA), block (AES) and hybrid (SHA) encryption techniques were applied and finally the results achieved were estimated. As a result, in the field of health, the most efficient technique for data exchange is the Hybrid encryption (SHA 256 algorithm), since, when exchanging data, it is the most efficient encryption in terms of total processing time with a result of 0.00450016 seconds and performance 0.276%, i.e. between encryption and decryption time it is the one that performed the task faster.

Keywords: Internet of Things, encryption, data interchange, privacy

Índice

Tabla de Figuras

Índice de tablas

I. INTRODUCCIÓN

1.1. Realidad Problemática.

En la actualidad, Internet de las cosas (IoT) se define como lo último en tendencia continua de conectar varios objetos físicos a una red, dominándolos "inteligentes" en especial objetos que anteriormente no se podría imaginar. (RedHat, 2019).

Según ISAM (Medicina Científica Industrial), actualmente hay cientos de millones de usuarios navegando mediante Internet en todo el mundo, y están aumentando cada día. El crecimiento de la concurrencia de navegabilidad se debe al gran avance tecnológico desarrollado a gran escala. Actualmente, los avances tecnológicos nos han permitido conectar 54 millones de dispositivos a Internet. (Susana, 2020). Se podría indicar que, la comunicación de dispositivos en IoT nos beneficia para que "la comunicación sea liviana, segura y confiable para poder transportar los datos de comunicación en diferentes capas sin comprometer limitaciones computacionales y de energía en los dispositivos de IoT" (Sharma, Shri, Vaishno, Shri, & Vaishno, 2018)

A nivel mundial, el número de usuarios que se conectan al Internet creció de 3.9 mil millones en el 2018 y crecerá en 2023 a 5.3 millones. En términos de población, esto representa el 51% de la población mundial en 2018 y el 66% de la población mundial del 2023. Sin embargo, "Internet de las cosas (IoT) ha crecido a un ritmo sin precedentes es decir en estos años pasados se han conectado miles de millones de dispositivos en diversas industrias crezca durante la próxima década" (Corak, Guzel, Murt, & Ozdemir, 2018)

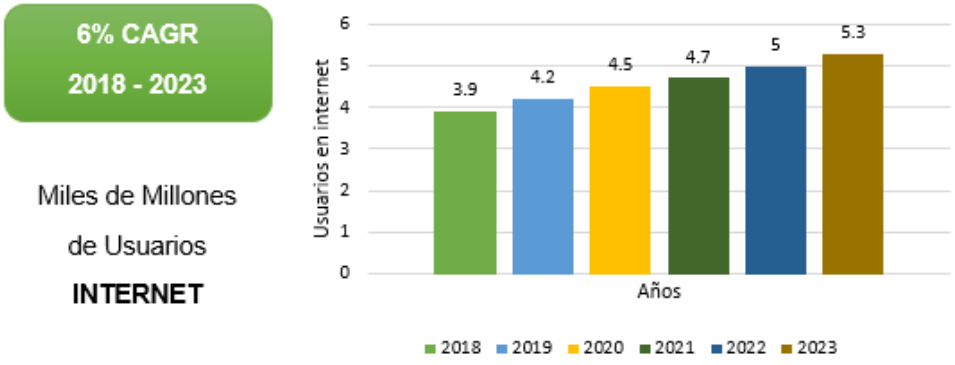


*Figura 1.*Usuarios Conectados a Internet. Fuente: *(Cisco, 2016)*

Otro concepto valioso que recalcan diferentes autores en sus investigaciones es que el Internet de las cosas permitirá a las personas usar los servicios de conexión por medio de dispositivos IoT, en dónde la asistencia médica tendrá menor intensidad laboral y costo operacional. (InternetSociety, 2015)

Para el desarrollo global con relación a la salud el Internet de las Cosas, obtendrá casi USD 409.9 mil millones para el año 2022, por la automatización de dispositivos médicos conectados y el manejo de teléfonos inteligentes, así como también el uso de software que aceptará realizar pruebas en pacientes de manera más rápida y precisa.

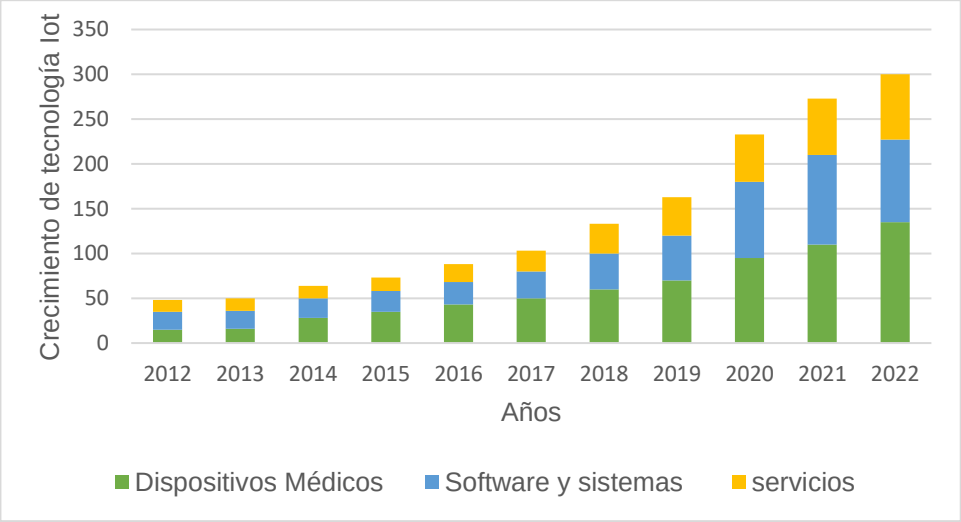


Figura 2. Dispositivos médicos en el mercado de la salud. Fuente: *(Quental, 2016)*

Se puede comentar que, otras investigaciones nos mencionan que hoy en día la atención médica está revolucionando con eficacia en la transferencia de datos con distintos proveedores de servicios relacionados, por lo que toda información que nos brindan los médicos se dará en forma automática, segura y remota, teniendo como resultado atender a muchos pacientes. (Quental, 2016)

Por otro lado, en el Internet de las cosas, es un dominio en el que se realiza la transferencia de datos cada segundo. La seguridad de estos datos es una tarea desafiante; sin embargo, los desafíos de seguridad se pueden mitigar con técnicas de cifrado de datos, estás técnicas son cruciales cuando se trata de la autenticación de usuarios y la privacidad de los datos. (Khari, 2019)

Al largo de los estudios se crearon varias técnicas de cifrado para el intercambio de datos en IoT en el ámbito de la salud, a continuación, se mencionará algunas investigaciones realizadas:

En el estudio realizado por Manju, Kumar, & Amir, (2019), realizó la investigación, Securing Data in Internet of Things (IoT) Using Cryptography and Steganography Techniques, en la Universiti de Balusamy. Los datos que se transfieren a través de la red de IoT son vulnerables a los ataques, por lo tanto, la información personal puede ser fácilmente pirateada por el sistema.

Por esta razón, se presenta y se discute el protocolo de criptografía elíptica de Galois, en este protocolo, se utiliza una técnica de criptografía para cifrar datos confidenciales que provienen de diferentes fuentes médicas, se utiliza una técnica de esteganografía de codificación Matrix XOR para incrustar los datos cifrados en una imagen de baja complejidad, también se utiliza un algoritmo de optimización llamado Adaptive Firefly para optimizar la selección de bloques de cobertura dentro de la imagen, se propone la criptografía elíptica de Galois (EGC) contra la filtración de datos durante la transmisión a través de la red IoT, el algoritmo cifrado dentro del controlador encripta los datos usando protocolo EGC y luego se oculta el mensaje encriptado y seguro en capas de la imagen con ayuda de la técnica estenográfica, la herramienta de simulación MATLAB se utiliza para la implementación del trabajo propuesto y utiliza H.264/AVC estándares de video, se evaluaron los parámetros para mostrar la eficiencia del sistema EGC propuesto, como la eficiencia de incrustación, capacidad de portadora, señal de pico de ruido, el error cuadrático medio (MSE) y la complejidad del tiempo.

Los resultados obtenidos señalan que el protocolo EGC dio un mejor rendimiento con respecto a incorporación de eficiencia, capacidad de portadora, PSNR, MSE y tiempo complejidad en comparación con otras técnicas, como LSB esteganografía, esteganografía FMO y esteganografía OMME, la complejidad del protocolo EGC propuesto es muy baja, en comparación con los

métodos existentes. El protocolo propuesto arrojó un mejor rendimiento de PSNR en comparación con LSB, FMO y OMME (32,42%, 45,62% y 52,24% de mejor rendimiento); el protocolo propuesto produjo un mejor rendimiento de la capacidad de portadora en comparación con LSB, FMO y OMME (0,33%, 16,35%, y 9.36% mejor desempeño, respectivamente); el protocolo propuesto arrojó un mejor rendimiento de eficiencia de integración en comparación con LSB, FMO y OMME (31%, 21% y 1,16% de mejor desempeño, respectivamente), por lo tanto, en general, el método propuesto está bien optimizado y arrojó mejores resultados en comparación con el existente protocolos. Concluyeron que el protocolo EGC generó altos niveles de seguridad de los datos para proteger los datos durante la transmisión en el IoT, con el nuevo ECC sobre el campo de Galois, el protocolo EGC propuesto proporcionó una mayor seguridad. Debido a la eficiencia de incrustación mejorada, se puede lograr una capacidad avanzada de ocultación de datos, con la ayuda del protocolo propuesto y la optimización Adaptive Firefly, cualquier cantidad de datos se puede transmitir fácilmente a través de la red de IoT oculta de forma segura dentro de las capas profundas de imágenes.

En otra investigación de (Bhandari & Kirubanand, 2019), propusieron una solución criptográfica mejorada que combina las características de los algoritmos de cifrado asimétrico y el servidor de clave pública, enfrentándose al problema de que mediante el uso de códigos para que aquellos a quienes se destina la información puede ser leída y que se genere una carga de datos, que carezcan de estándares adecuados de seguridad y privacidad. Como método de solución implementaron una técnica de cifrado mejorado, para que los datos que se transmiten entre diferentes nodos estén seguros y protegidos, obteniendo como resultado que en criptografía simétrica solo se tiene una clave común para el proceso de cifrado y descifrado Al tener una clave común para ambos procesos, el algoritmo simétrico tiende a ser más rápido en todo el proceso ya que no hay generación de pares de claves involucrada.

En otra investigación de (Er & attri, 2017), discuten las diversas técnicas de cifrado para eliminar las vulnerabilidades que determinan un conjunto de

características de seguridad como autenticación mutua, el anonimato y la confidencialidad, donde utilizaron la criptografía de curva elíptica y se puede usar para crear claves criptográficas que son muy eficientes y de tamaño pequeño. Donde sus resultados dicen que la mayor parte de la técnica existente tiene ciertas deficiencias porque ha descuidado algunas la velocidad de transmisión de datos sigue siendo un problema difícil y el uso de la técnica de encriptación basada en el ADN es ignorado por la mayoría de las técnicas existentes por los investigadores en el campo de IoT. (resultados puntuales)

Finalmente debe quedar claro que, en el campo de ingeniería en lo que respecta la Evaluación de técnicas de cifrado para el intercambio de datos en internet de las cosas en el ámbito de la salud, si bien es cierto, existen varias técnicas de cifrado, sin embargo, se ha podido advertir que a lo largo de los años estás técnicas presentan problemas de deficiencia en el intercambio de datos, debido que con el auge de los dispositivos IoT y su gran interconexión genera un gran número de amenazas y ataques, dando lugar a que existan ciberdelincuentes; por lo tanto, de todas las técnicas se debe evaluar cual es la más eficiente, adecuada y asertiva a fin de proteger a los usuarios de un incidente de ciberseguridad y de esta manera reducir los riesgos y amenazas, es por ello que se seguirá buscando información para poder seguir investigando y llegar a resultados favorables que hoy en día se tiene con diferentes técnicas de cifrados en IoT.

1.2. Trabajos previos.

Chaudhary & Chatterjee, (2020), realizó la investigación, An Efficient Lightweight Cryptography Technique For IoT based E-healthcare System, en el National Institute of Technology Patna de la India. La transferencia segura de datos de salud es esencial para el tratamiento, además, al almacenar el registro del paciente, debe mantenerse a salvo del uso indebido y la modificación de datos, ya que puede ser rastreado fácilmente por otros dispositivos, debido a los dispositivos de IoT restringidos, es muy difícil cifrar los datos con técnicas criptográficas de alto nivel.

Por esta razón, se propuso una nueva técnica de cifrado de bloques para la transmisión segura de datos desde estos dispositivos de IoT, en donde se analizan tres cifrados de bloques ligeros basados en Addition-Rotaion-XOR (ARX), donde se basan en una función de rotación de matriz, XoR y expansión, el proceso depende principalmente de la función de expansión, función de generación de claves redondas, luego las técnicas de cifrado se implementan en el microcontrolador AVR(Arduino) de 8 bits, se procesaron datos de tamaño de bloque de 2kb para cifrado y se muestran los resultados experimentales. Los resultados obtenidos mostraron que para analizar el estudio verificaron el uso y la velocidad de la memoria, y aplicaron las otras técnicas de cifrado de bloques como AES, DES, SIMON, muestran los resultados de comparación en donde se encuentra que el uso de memoria de AES, DES y SIMON es superior al propuesto, además el tiempo de cifrado y descifrado es menor que con el otro cifrado ligero, se obtiene que en el algoritmo AES la velocidad (ciclo de reloj/ bytes) es 24695 y en el algoritmo DES la velocidad es 28401 y por último en el algoritmo SIMON es 39313 y la Memoria uso (Bytes) da como resultado en AES 1709, en DES 1608 y en SIMON 755 en donde se muestra los diferentes tiempos de cifrado de los mismos datos de entrada utilizando diferentes algoritmos, la técnica de cifrado lleva un tiempo equivalente al de otros cifrados de bloques estándar. Este trabajo propuesto concluye que la implementación muestra que la ocupación de memoria es pequeña mientras que la velocidad es considerable para generar un cifrado eficiente y que esta técnica propuesta se puede implementar y probar en varios dispositivos restringidos.

El-Latif, Abd-El-Atty, Mazurczyk, & Member (2020), realizó la investigación, Secure Data Encryption Based on Quantum Walks for 5G Internet of Things Scenario, en lMenoufia University de Egipto. Las redes de quinta generación (5G) son la tecnología de comunicación básica para conectar objetos en el entorno a Internet de las cosas (IoT), pero conducirá a nuevos y enormes desafíos de seguridad y privacidad, ya que, el contexto de 5G-IoT impulsa a las personas, las organizaciones y los agentes de software a interactuar con

entidades que no son de confianza e intercambiar información con diversas fuentes desconocidas que, como resultado, pueden conducir a muchos problemas de seguridad potenciales.

Por esta razón, se propone la utilización de tecnologías cuánticas en el campo de las tecnologías 5G-IoT para mejorar el rendimiento y resistir algunos posibles ataques de escenarios clásicos y cuánticos**,** para construir un nuevo método de caja S que juega un papel importante en las técnicas de cifrado de bloques para las tecnologías 5G y también se propone un nuevo mecanismo de cifrado de video robusto, utilizando las características de Quantum Walk para proponer una nueva estrategia de cifrado para la transmisión segura de archivos confidenciales en el paradigma 5G-IoT y para evaluar los mecanismos de cifrado de datos propuestos, iniciaron con los parámetros de CAQW y los valores iniciales utilizados durante la construcción de la caja S, después analizaron el espacio clave, por el número total de claves disponibles, luego analizaron el rendimiento del cifrado del video propuesto y seleccionaron seis videos cada uno con 150 cuadros y diferentes tamaños como videos de prueba. Los resultados obtenidos para el tiempo de cifrado del mecanismo presentado y el mecanismo de cifrado de archivos propuestos para la computación en la nube y para varios tamaños de archivo resultan que el tamaño de archivo de 1KB propuesto da como resultado 0,0069 para 10KB el propuesto es 0,0247, para 100 KB es 0,1940, para 1MB es 2,0845, 10MB 21,9606, y finalmente para 100MB el propuesto es 250,3024; para el mecanismo es el proceso inverso de método de cifrado, por lo tanto, basta con calcular el tiempo de cifrado del mecanismo propuesto para la computación en la nube para varios tamaños de archivo que evidencia de la buena ejecución de la propuesta. Este trabajo propuesto concluye que los criptosistemas propuestos demostraron que tiene propiedades de seguridad robustas y es eficiente en términos de rendimiento criptográfico gracias al nuevo mecanismo de encriptación de video robusto basado en CAQWs y un mecanismo S-Box.

Lin, Cheng, & Chuan, (2017), realizó la investigación, Lower Power Data Transport Protection for Internet of Things (IoT), en el Institute Information Industry de Taiwan. Las preocupaciones sobre la seguridad del transporte de datos de internet de las cosas obstaculizan el progreso de servicios inteligentes, los datos actuales se transportan principalmente en texto claro sin protección de la privacidad y los recursos del equipo de IoT son escasos, por lo que no se pueden aplicar cifrados y descifrados complejos.

Por consiguiente, propuso el método de respuesta dinámica centrado en el tiempo (TBDR) para asegurar la exactitud de la restauración de datos mediante la negociación de los parámetros aleatorios y las ID dinámicas de ambas partes a lo largo del intercambio de datos, para ello se estimó previamente una tabla de lista de candidatos C para un análisis rápido de restauración, para cada una de las cadenas de datos candidatas, se utilizó el filtro Bloom y los parámetros k*, h*, y semilla* para establecer el conjunto de datos B_all, para aquellos bits en B_all que no han sido procesados por la respuesta aleatoria permanente, se construye una table de lista de candidatos C de la cual bit=verdadero, donde S5 representa la quinta cadena candidata, lo que indica que el primer bit es verdadero después que la cadena se somete al filtro Bloom y también se recopila estadísticas para cada id de cadena de lista de candidatos para obtener el número total de (bit=Verdaderos) y la cadena con nmáx más alto es la cadena de datos originales, en este experimento el 98,81% de todas las cadenas probadas tienen nmax = 4, que se pueden identificar inmediatamente como cadenas originales, mientras que en el 1,19% restante, el número máximo de cadena original candidata es 10, es decir, solo necesita comparar más 10 veces para encontrar la cadena original correcta. Los resultados del experimento que eligió semilla= [2,3,7,13], h = 4, k = 256 para el filtro Bloom, f1 = 0.02, f2 = 0.01 para la respuesta aleatoria permanente y fD = 0.3, m = 100 para la respuesta aleatoria dinámica, los cuales muestran que la tasa de error en el transporte de la cadena de datos se puede reducir al 0% en un rango de entrada de datos limitados, además el cálculo puede ahorrar más del 90% del tiempo en comparación con las técnicas comunes de cifrado y descifrado (como AES), que

son adecuadas para la protección del transporte de datos entre dispositivos ligeros de IoT, también la comparación del costo de tiempo entre el método propuesto en este estudio y los métodos tradicionales de encriptación y desencriptación muestran que el tiempo y el costo para 100,000 veces de operaciones bajo condición de protección de 256 bits, el método propuesto ofrece un costo de tiempo significativamente menor que los algoritmos tradicionales AES, DES, logrando así eficazmente el rendimiento competitivo de baja potencia en la tabla se muestra los resultados de comparación de los costos de tiempo para 100K operaciones de diferentes algoritmos que evalúan la longitud de clave , bits obteniendo como resultado para AES = 766, para DES =954, y para TBDR en 256 = 56. Concluyeron que las cadenas de datos candidatas de “0” a “9999” se comprobaron mediante el transcurso de codificación y decodificación y que las técnicas de cifrado y descifrado como AES es adecuado para aplicaciones en la protección del transporte de datos entre dispositivos IoT ligeros de menor potencia.

Sarmilla & Manisekaran, (2019), realizó la investigación, A Study on Security Considerations in IoT Environment and Data Protection Methodologies for Communication in Cloud Computing, en la University Regional Campus de India. La gran demanda de dispositivos conectados a través de un marco de IoT incrementa el riesgo de ataques, como defectos en la administración de aplicaciones, denegación de servicio, compromiso de privacidad del usuario, con consecuencias sociales.

Por ello, se concentran en la protección de datos entre todos los problemas de seguridad en el entorno de IoT, se discuten las técnicas de varios métodos criptográficos, se ilustran algunos métodos utilizados y se realiza el debate sobre la criptografía ligera en el cifrado extremo a extremo (E2EE) para proteger el secreto y privacidad del usuario, el método que utilizaron es que en la sesión dos hablar sobre el marco general de Internet de las cosas y cuestiones relacionadas en la IoT, luego se enfocan en los problemas de seguridad de la computación en la nube y la integración, seguido de varias técnicas

criptográficas ligeras utilizadas para resolver varios problemas de seguridad en el entorno de IoT, después hacen una discusión sobre el cifrado de extremo a extremo para obtener una propiedad de seguridad poco común , y finalmente realizan una encuesta con un resumen de manera prometedora, que brindan varias direcciones para futuras investigaciones sobre la protección de datos en el entorno de IoT para la comunicación de datos en la nube. Los resultados obtenidos muestran que, la protección de datos necesita una técnica criptográfica para prevenir amenazas a los datos, como la criptografía ligera estándar desarrollado de forma similar a AES, considerando la seguridad y la eficiencia del hardware como resultado en la cifras de bloques ligeras con equivalencia de puerta para el algoritmo PRESENT-80, el tamaño de bits es de 64, en el algoritmo AES-128 es de 16, en el GIFT es de 128, SIMON es de 64, SPECK es de 64, TWINE es de 64 junto al de KATAN y en los tamaños de clave en el algoritmo PRESENT-80 es de 80, AES es de 128 igual que el de GIFT, SIMON que es de 64, SPECK de 96, TWINE de 80 junto al de KATAN, también indican que ANU-II es una técnica de cifrado ultraligera tiene un buen rendimiento de software y hardware, tiene una versión modificada y eficiente de ANU con menor consumo de energía de 24 Mw y tiempo de ejecución de 437,31 kbps, con baja equivalencia de puerta(1010) y alto rendimiento(437.31Kbps), también muestra seguridad contra ataques lineales y un número mínimo de rondas, el resultado muestra una buena propiedad de avalancha de esta técnica. Concluyeron que se revisaron algunas técnicas como ANU-II y que la mayoría de las técnicas utilizadas en un entorno restringido tuvieron éxito en la resolución de diferentes vulnerabilidades, satisfaciendo algunas propiedades de seguridad, también señalaron que el análisis de cifrado de extremo a extremo se realiza para perfeccionar la seguridad y la privacidad de los usuarios de confianza.

Shafiq, Tian, Bashir, & Du, (2020) realizó la investigación, CorrAUC: A Malicious Bot-IoT Traffic Detection Method in IoT Network Using Machine-Learning Techniques en la University Regional de China. Identificación de anomalías y tráfico malicioso en la red de Internet de las cosas (IoT) es esencial para la

seguridad para vigilar y bloquear los flujos de tráfico no deseados en la red de IoT.

Por ello, proponen una nueva técnica para la detección de ataques Bot-IoT en el entorno de red de IoT, para la selección efectiva de características, el método propuesto seleccionó solo cinco características efectivas, que contienen suficiente información para la detección de ataques de Bot-IoT en el entorno de red de IoT, para este objetivo de seleccionar características efectivas, se aplicaron cuatro algoritmos de aprendizaje automático diferentes para la evaluación del rendimiento de la técnica propuesta, como el árbol de decisión (C.4.5), la máquina de vectores de soporte(SVM), Naive Bayes y los algoritmos de aprendizaje automático de bosque aleatorio. Los resultados obtenidos muestran que los cuatro algoritmos de aprendizajes automáticos aplicados son efectivos para la detección de ataques de Bot-IoT en el entorno de la red de IoT mediante el uso del conjunto de características seleccionado por la propuesta con una técnica con exactitud, precisión, sensibilidad y especificidad respectiva, ataques como 99,9%, que es un resultado de rendimiento muy eficaz, sin embargo el resultado de rendimiento de Naive Baye es bajo en comparación con otros clasificadores de aprendizaje automático mediante el uso del conjunto de características seleccionadas con respecto a la métrica de precisión para la detección de ataques de Bot-IoT del mismo modo, el resultado de rendimiento de SVM es ligeramente superior con la precisión respectiva en comparación con los clasificadores Naive Bayes ML, también, los resultados generales de rendimiento del clasificador ML aplicado del árbol de decisiones C4.5 dan resultados efectivos en comparación con otros clasificadores ML aplicados. Por lo tanto, el algoritmo C4.5 ML funciona mejor al usar las características seleccionadas configuradas para la detección de ataques de Bot-IoT como 99.9%, que es un resultado de rendimiento muy efectivo. Concluyeron que la detección de ataques de red de IoT es esencial para que la seguridad de IoT vigile y bloquee los flujos de tráfico no deseados, sin embargo, el problema notable aún debe estudiarse más a fondo, es decir, cómo seleccionar funciones efectivas para la detección precisa de tráfico malicioso en las redes de IoT,

también proponen un nuevo modelo marco, que se propuso un enfoque de métrica de selección de características llamado CorrAUC, se desarrolla y diseña un nuevo algoritmo ML seleccionado utilizando la métrica AUC, también aplican TOPSIS integrado y Shannon Entropy basados en un conjunto de software y cuatro diferentes algoritmos; El análisis de resultados experimentales mostró que nuestro método propuesto es eficiente y puede lograr> 96% de resultados en promedio.

Roy, Rawat, & Karjee (2020), realizó la investigación, A Lightweight Cellular Automata Based Encryption Technique for IoT Applications en la Manipal University Jaipur de India. Varios servicios útiles proporcionados por diferentes aplicaciones de IoT son de gran beneficio para la vida humana, sin embargo, tienen un costo muy grande al considerar la seguridad y la protección de la privacidad de las personas, dado que los fabricantes de IoT no lograron implementar un sistema robusto y seguro a nivel de dispositivo, los expertos en seguridad ya advirtieron sobre el riesgo potencial de utilizar una gran cantidad de dispositivos vulnerables conectados a Internet.

Por esta razón, se propone un cifrado ligero basado en autómatas celulares (CA), denominado Lightweight CA Cipher (LCC) para aplicaciones de IoT, en el método propuesto, el cifrado se realiza en la capa de percepción, donde se despliegan los nodos sensores y el descifrado se realiza en la capa de red donde se instalan los dispositivos de puerta de enlace, para ello se discuten las funcionalidades detalladas del trabajo propuesto, se discute la importancia y las características clave de CA en las técnicas de cifrado ligeras y cómo el enfoque propuesto las aprovecha para desarrollar LCC, luego se propone un algoritmo basado en Group Cellular Automata(GCA) llamado Lightweight CA Cipher (LCC), que se implementa en dispositivos IoT para el cifrado de datos, finalmente el proceso de descifrado se realiza en los nodos de niebla en la capa de red.

Los resultados obtenidos muestran que, para el procesamiento de datos, cada dispositivo IoT genera 64 observaciones en cada marca de tiempo, para capturar el procesamiento de datos, la entrada al algoritmo de cifrado es una matriz, D teniendo dimensión, M × N = 25 × 64, donde 25 dispositivos IoT son utilizado y cada dispositivo genera 64 observaciones en cada momento intervalo, por lo tanto, todos los datos detectados se dividen como un bloque de matriz de 25x64, D, transmitido desde el nodo de niebla a la nube; el algoritmo propuesto es fácil de implementar en comparación con DES Y 3-DES, además, las pruebas NIST y DIEHARD se realizan en cifrado LCC, donde genera 512 bits de salida para 512 bits de entrada y se han considerado 25 nodos de este tipo con fines experimentales, este número se puede escalar hacia arriba o hacia abajo a cualquier número. Sin embargo, para las pruebas NIST se consideraron 6,40,000 bits de salida a la vez, el método propuesto es capaz de generar verdaderas secuencias aleatorias, también se observa que el tiempo de ejecución también es menor en el caso de LCC, lo que significa que esto se puede implementar para el sensor con recursos limitados. Concluyeron que este método es una técnica de cifrado de clave simétrica y se ha manejado de manera eficiente las limitaciones de los nodos de IoT implementados en la capa de percepción, debido a su simplicidad inherente, la capacidad de crear consecuencias caóticas ha cumplido su propósito. LCC se puede utilizar en cualquier tipo de nodos sensores, sin embargo, la técnica de cifrado de claves simétricas adolece de complejidades en la gestión de claves.

Rahman, Ahmed, & Hussain, (2016) realizó la investigación, Comparison of Data Aggregation Techniques In Internet of Things (IoT) en la Hill University Shillon de India. Los datos generados por los nodos de sensores vecinos suelen estar correlacionados y son muy redundantes, además, la cantidad de datos generados en enormes redes de sensores suele ser excesiva para que los procese el nodo receptor; por tanto, se requiere un método para combinar estos datos redundantes y correlacionados en una información válida de alta calidad en los nodos intermedios.

Por esta razón se propone, un análisis comparativo de algunas técnicas de agregación de datos en Internet de las cosas con referencia a parámetros como disipación de energía, vida útil de la red, rendimiento, latencia y número de nodos activos, el cual implica calcular datos de múltiples nodos sensores en nodos intermedios y enviar datos agregados al nodo receptor, se utiliza la herramienta de simulación NS-2 para varios parámetros selectivos importantes y analizaron los resultados de la simulación con las métricas de rendimiento elegidas, se presenta un análisis del desempeño de algunos esquemas de agregación existentes, la simulación de los protocolos LEACH y LEACH-C se logrará utilizando NS-2, se usó también una red de 100 nodos que son aleatorios distribuidos entre los puntos (X = O, Y = O) y (X = 100, Y = 100) (distancia en metros) con el BS en la ubicación (50, 175). Los resultados obtenidos muestran que, la disipación total de energía aumenta junto con el tiempo en los protocolos LEACH y LEACH-C, pero LEACH disipa más energía que LEACH-C, en LEACH-C, la estación base tiene pleno conocimiento de posición y energía; los parámetros de simulación muestran que en el número de nodos se aprecia 100, en el área de simulación 100*100 metros, en el tiempo de simulación es de 600s, ancho de banda del canal 1Mbps, retraso de procesamiento 25J1.s, tamaño de paquete de 500 bytes, duración de cada ronda 20 segundos, muestran que todos los nodos dentro de la red hace que sea eficaz para decidir la selección de CH y la construcción del clúster, por tanto, esto resulta en una menor disipación de energía en comparación con la de LECH y como LEACH utiliza el método probabilístico de selección de CH y números de conglomerados, a veces la distribución desigual de CHs es posible que conduce un aumento repentino en la disipación de energía. Concluyeron que la agregación de datos reduce la cantidad de transmisiones necesarias para los datos recopilados, lo que a su vez reduce el consumo de energía, la latencia y el tráfico de la red, analizando el rendimiento de los protocolos de agregación LEACH Y LEACH-C en IoT.

Siddhart & Devgon, (2019), realizó la investigación, Analysis of Encryption and Lossless Compression Techniques for Secure Data Transmission en la

University Capgemini de India. La seguridad de la información es una preocupación importante en la actualidad, el problema de compartir información está relacionado con las limitaciones de ancho de banda y alto riesgo para la seguridad de datos.

Por esta razón se propone comparar técnicas de cifrado y comprensión que se prueban para encontrar el método más eficaz que puede ayudar a generar datos con cifrado seguro y compresión sin pérdidas con sus respectivas ventajas y desventajas, también clasifican y describen las obras relacionadas con algoritmos de cifrado.

Los resultados obtenidos muestran la comparación de todos los algoritmos considerados, en el trabajo de investigación de comprensión de imagen compuesta asegurada utilizando técnicas de encriptación con el algoritmo DCT y operación de mezcla de imagen codificada con el tamaño de imagen de 512x512 y resulta con un índice de comprensión de 44,16% y ventaja que utiliza la técnica de comprensión es preferible para las imágenes, el siguiente paper es un esquema de cifrado escalable para el estándar de comprensión de datos de imágenes CCSDS, resulta con un algoritmo de CCSDS y esquema de cifrado escalable, también con un 1024x1024 y con un índice de compresión de 12,19%, en el paper Mecanismo de comprensión para sistema multimedia teniendo en cuenta la seguridad de la información, muestra un resultado de 256x256 con un índice de comprensión de 56,8%, en el paper Fusión, comprensión y cifrado simultáneos de varias imágenes, con el tamaño de imagen de 256x256 con un índice de comprensión de 56,8%, en el paper Cifrado parcial de comprimidos con el tamaño de imagen de 512x512 con un índice de comprensión de 26,8%.Concluyeron que se consideró la necesidad de técnicas de cifrado y comprensión para la transmisión segura de datos a través de una red, y que analizaron y aprendieron muchos algoritmos diferentes para cifrar y comprimir, a su vez describen las relaciones de compresión para diferentes tamaños de imagen y algoritmos, la investigación que realizaron se clasificó en

tres categorías, es decir, comprensión luego de la encriptación, encriptación luego comprensión, finalmente comprensión y encriptación conjunta.

Kumar, Khan, & Kumar, (2019), realizó la investigación, A Cellular Automata-based healthcare data encryption technique for IoT Networks en Institute of Technology Patna en la India. Proporcionar una comunicación segura es un desafío clave para varias aplicaciones sensibles como en el hogar inteligente, la red inteligente, los algoritmos de cifrado y descifrado convencionales no son adecuados para aplicaciones de IoT debido a sus complejidades y requisitos de energía.

Por esta razón en el método propuesto proponen una técnica de cifrado basada en autómatas celulares (CA) para datos de salud de ECG, donde señalan que la CA ligera es adecuada para dispositivos IoT porque no utiliza datos redundantes para el cifrado y utilizan tres algoritmos, el primer algoritmo convierte los datos sanitarios sin procesar a un formato específico para que se pueda aplicar las reglas de CA, luego se aplica un CA uniforme a los datos formateados y se transmite por el canal de comunicación, usando claves simétricas para el cifrado y descifrado, consideran la propiedad de la regla 60 que se puede utilizar con fines de cifrado, se analiza si las secuencias de bits generados son aleatorios o no, se realiza pruebas NIST (Instituto Nacional de Estándares y Tecnología) sobre datos cifrados. Los resultados obtenidos muestran los datos ECG en tiempo real se adquieren utilizando el kit de atención médica MySignals con los pocos valores de ECG normalizados 940, 920, 910, 915, 845, 935, 905, 955, 800, 175 dando como resultado de la matriz de datos de ECG encriptados 1268, 1192,1170, 1205, 1494. 1257, 1179, 1229, 1376 y finalmente 497, también muestran resultados de la representación binaria de datos de ECG normalizados y su versión encriptada en dónde se muestra los resultados de la prueba de frecuencia con un valor de 1252, en la prueba universal con 3848, en la prueba de suma acumulada de 2699 y en la prueba de variante de excursiones aleatorias de 2882 dada la representación binaria de la misma parte después de aplicar la regla 60 de CA. Concluyeron que se

presenta una técnica de cifrado eficiente y ligera que utiliza CA que se utiliza para la seguridad de los datos sanitarios, y se están realizando más trabajos considerando el ruido en el canal de comunicación, en dónde se implementa reglas de CA híbridas en diferentes instantes de tiempo para el cifrado y descifrado.

Afsoon & Seyed, (2017), realizó la investigación, Improving the Security of Internet of Things using Encryption Algorithms, en la Universiti Azad de Iran. La seguridad de la transmisión de datos es ineficiente muchas veces para la seguridad del sistema.

Por esta razón, propuso un algoritmo cifrado híbrido con el nombre de HAN, mediante la combinación de dos algoritmos de AES y NTRU, algoritmo sugerido tiene características especiales en cifrado y descifrado en términos de velocidad incluso en la construcción de claves, se realizó para disminuir riesgos de seguridad, restablecer la velocidad del cifrado y una menor complejidad computacional, también el software MATLAB simuló el algoritmo de cifrado AES y RSA, se evaluó su velocidad, eficiencia y seguridad en comparación con el algoritmo de cifrado convencional, se calcula el parámetro b multiplicar la clave privada f en el mensaje inicial que se ha enviado por el remitente, la firma digital en HAN se utiliza para la validez del mensaje , prueba de identidad y seguridad.

Los resultados obtenidos señalaron que el algoritmo HAN se considera un método sugerido, que es una combinación del algoritmo de cifrado simétrico AES y el algoritmo de cifrado asimétrico NTRU para la mejora de IoT, este algoritmo tiene alta velocidad para crear una clave, los siguientes resultados del tiempo de velocidad total del algoritmo de HAN en comparación con otros dos algoritmos de encriptación es de HAN = 0.5 , AES = 2.6, RSA=2.3, al comparar la firma con otros dos algoritmos que no habían aplicado firma digital los resultados se muestran que algoritmo HAN por señal digital 0.58, AES sin señal digital 2.718182, RSA sin señal digital 2.350752; HAN consume menos energía en lugar de los otros dos algoritmos HAN = 11/81%, Y RSA=13/65%.

Concluyeron que se proporcionó un método sugerido que puede mejorar IoT mediante un algoritmo de cifrado híbrido, el algoritmo HAN se considera un método sugerido que es una combinación del algoritmo de cifrado asimétrico AES y el algoritmo de cifrado asimétrico NTRU para la mejora de IoT, este algoritmo tiene alta velocidad para crear una clave, cifrado y descifrado y seguridad aceptable en IoT, la seguridad de este algoritmo se debe al uso multinomial en cifrado, descifrado y digital.

Oualha (2018), realizó la investigación, Reinforcing IoT-Enforced Security Policies, en la Universiti CEA de Francia. La falta de protección de la confidencialidad y el control de acceso a una gran cantidad de datos generados por Internet de las Cosas.

Por esta razón en el método propuesto proponen, una nueva técnica que radica en el uso de cifrado basado en atributos (ABE), que se basa en entidades intermedias de confianza parcial para reforzar las políticas de gran acceso derivadas de las políticas pequeñas impuestas por los dispositivos de la IoT sin relevar los datos cifrados; distinguieron entre dos tipos de atacantes, externo y un interno, en el modelo de escenarios agregaron un nuevo actor, el PRP, que refuerza el texto cifrado con una nueva política, también describieron su implementación de software del esquema Waters en una plataforma de sensores del mundo real, que plantea el número de atributos, implementaron la red de sensores, usando la plataforma Zolertia Remote Revisión, que cuenta con un microcontrolador ARM Cortex M3 que corre hasta 32 MHz e incluye 32 Kbytes de RAM y 512 Kbytes de Flash, todos los mensajes intercambiados entre los sensores y la PC se transmitieron utilizando el protocolo de aplicación restringida (CoAP), que es un protocolo de aplicación de Internet para dispositivos.

Los resultados obtenidos señalaron que se obtiene un número máximo de 12 atributos ejecutando una pila de redes centradas en el contenido directamente sobre la radio 802.15.4 sin ninguna capa UDP o IP subyacente, hace que la pila

de comunicación sea liviana, dejando más RAM y procesador para los cifrados ABE, también se muestra el consumo total de energía del algoritmo de encriptación del esquema Waters, variando el número de atributos en la política de acceso, demuestra que el consumo de energía aumenta, aproximadamente de forma lineal, con el número de atributos, de 2 a 9 atributos, la energía consumida aumenta de alrededor de 9mJ a alrededor de 25mJ, en el siguiente resultado se consolida el resultado del aumento lineal de los recursos utilizados con el número de atributos en la política, el tiempo de ejecución aumenta alrededor de 5 segundos a alrededor de 14 segundos al aumentar el número de atributos de 2 a 9, para 9 atributos en la política de acceso, el tiempo de ejecución de algoritmo de cifrado en el dispositivo sensor llega a 13 segundos, también se obtiene que al agregar 100 atributos a la política de acceso a través de la encriptación del proxy, requiere alrededor de 2,4 segundos. Concluyeron que cuantos más dispositivos estén conectados al IoT, más datos se generarán, así mismo, , CP-ABE ofrece un enfoque elegante para proporcionar protección de la confidencialidad de los datos y un control de acceso detallado, se describe una técnica propuesta que permite construir a partir de textos cifrados bajo políticas de acceso pequeñas aplicadas, por ejemplo, por dispositivos IoT, textos cifrados bajo políticas de acceso más complejas y grandes, esto se logra sin descifrar el texto cifrado, pero mediante el refuerzo de políticas.

Sarin, Thanawala, Verma, & Prakash, (2020) realizó la investigación Implementation of New Approach to Secure IoT Networks with Encryption and Decryption Techniques, en la Universiti Mukesh Technology Management and Enginerring de Mumbai India. La tecnología de conectar todo a Internet puede resultar desalentador, los datos podrían ser atacados fácilmente por usuarios malintencionados y, por lo tanto, el uso de dichas tecnologías queda en una situación precaria. Los riesgos de seguridad en IoT aumentan debido a la gran cantidad de puntos finales en la red que son más susceptibles a ataques externos.

Por esta razón en el método propuesto proponen, utilizar el algoritmo AES para proteger los bits de datos y también diseñaron un algoritmo de cifrado y descifrado de dos etapas, este algoritmo se probó con éxito transmitiendo bits de datos a través de un canal entre dos nodos de sensores inalámbricos para cinco casos de prueba, el modelo de prueba de tal manera que puede ser universalmente aplicable a todas las redes de IoT, en el modelo propuesto los datos se mantienen protegidos con una función de seguridad de dos niveles , el nivel 1 es la codificación de los datos en el nodo 1, y el nivel 2 está cifrando los datos codificados mediante el algoritmo AES, utilizaron un Arduino Uno para el nodo del sensor en el extremo del usuario y también en el otro extremo.

Los resultados obtenidos señalaron que se registraron 100 entradas de datos a los 5 sensores, en donde se muestra la salida ultrasónica como 177cm y el número aleatorio que se generó, en ese caso es 69 que cumple la primera condición de los datos y, por lo tanto se convertirá un número que es la mitad de esto, al número decimal más cercano, también se muestra que los datos que se capturaron después de filtrar las 2000 entradas el tamaño de entrada de datos se varía para cumplir con la irregularidad que podría enfrentar el modelo en datos genéricos. Concluyeron que el avance en internet y las tecnologías inalámbricas brindan innumerables comodidades, sin embargo, surge que nuestra información importante es accesible para usuarios fraudulentos, por lo tanto, desarrollaron un modelo de codificación dos veces en el extremo de transmisión antes de enviarlos al canal y después de ser recibidos en el receptor se cifran utilizando el algoritmo de cifrado AES, luego los datos se almacenan y el usuario en el receptor puede descifrar los datos seguido de una decodificación de dos etapas para recuperar los datos originales.

Singh, Johari, Singh, & Tyagi, (2019) realizó la investigación Mercurial Cipher: A new Cipher technique and comparative Analysis with classical Cipher Techniques, en la Universiti USICT, GGSIP de Dwarka, India. Las técnicas de cifrado tradicionales se han quedado obsoletas porque adolecen de muchas vulnerabilidades, tienen problemas de la gestión de claves, ya que la mayoría

de las técnicas de cifrado se basan en la clave y es la parte más vulnerable de cualquier mensaje cifrado.

Por esta razón en el método propuesto proponen una nueva técnica de cifrado mercurio modificada después de analizar las técnicas de cifrado clásicas, que puede manejar la clave fácilmente entre el remitente y el receptor, brinda suficiente seguridad al mensaje para que no se pueda descifrar fácilmente, la primera clave toma la ayuda de la función aleatoria para generar las claves a partir del conjunto de 36 caracteres (az y 0-9) para obtener una longitud de 36 caracteres, lo que hace que el conjunto total de claves posibles sea 3636 (es decir, 256 aproximadamente) de claves en las que cada carácter correspondiente de texto plano se multiplica por el carácter clave y aumenta su valor del carácter clave en cada paso subsiguiente y se forma el carácter resultante, luego se realiza la operación XOR en la segunda clave, en la segunda fase del proceso de cifrado se involucra la implementación de la segunda clave y se logra con cálculos.

Los resultados obtenidos señalaron que, al analizar un análisis comparativo detallado sobre la base de parámetros como el tiempo de ejecución, la memoria requerida, el tiempo de CPU, la línea de códigos entre las técnicas de cifrado, se muestra que, en el tiempo de ejecución el cifrado Cesar obtuvo 300, el cifrado de ferrocarril 200, el cifrado Vigenere 355, el cifrado Mercurial 200, también se muestra la memoria requerida (en Kilobytes) en el cifrado Cesar se muestra 23, en el cifrado de Ferrocarril 5, el cifrado Vigenere 23 y finalmente en el cifrado Mercurial 24, en el tiempo de CPU(en milisegundos) se muestra que el cifrado Cesar obtiene un 40, cifrado de Ferrocarril 10, cifrado Vigenere 40, y por último el cifrado Mercurial 10, finalmente se obtiene que en términos de tiempo de ejecución del cifrado Mercurial, calcularse en nanosegundos es el menos entre las otras técnicas de cifrado, el tiempo de CPU del cifrado Mercurial, en milisegundos, es significativamente menor que el cifrado Cesar y las técnicas de cifrado Vigenere. Concluyeron que Mercurial se ha generado con éxito resolviendo problemas de clave de cifrado de las técnicas de cifrado, capaces

de funcionar dentro de los diferentes lenguajes debido al valor Unicode que se forma en los cálculos de la fase de cifrado, y que, en un futuro, el cifrado mercurial se haría más seguro aumentando el número de claves de dos tres al incluir la dirección IP del sistema también como clave potencial.

1.3. Teorías relacionadas al tema.

1.3.1. Internet de las cosas

En la actualidad, la tecnología se ha integrado en diferentes puntos y ámbitos de nuestra vida cotidiana desde tecnologías en información como en los dispositivos e infraestructuras. (Vermesan, 2011) explica que IoT permite la conexión a internet entre las personas y los objetos inteligentes, desde cualquier ambiente o lugar donde este se encuentre, accediendo a múltiple servicios o actividades desde un dispositivo conectado a internet. Está disposición de tecnología facilita el proceso de muchas actividades a que el usuario obtenga lo que desea con solo un clic, generando una orden inmediata al objeto.

IEEE (2015) define a Internet de las cosas (IoT) como la proporción hacia una red conectada a internet, manifestándose como dificultoso, pero que a su vez es ajustable y autoconfigurable que conecta "objetos o cosas" a Internet por medio de protocolos de comunicación estándar. Los dispositivos u objetos conectados entre sí tienen representaciones virtuales, funciones de detección y actuación, funciones programables y pueden identificarse de forma única en el mundo digital. La información contenida en esta representación incluye identidad, estado, ubicación o a su vez también obtiene información social y comercial del contexto en el que se desarrollen. El Internet de las cosas proporciona servicios mediante el reconocimiento único, captura de datos y capacidades de desempeño manteniendo una comunicación sin intervención humana. El servicio se desarrolla mediante el uso de una interfaz inteligente y se puede utilizar en cualquier momento y en cualquier lugar, teniendo en cuenta la seguridad.

1.3.2. Arquitectura de internet de las cosas

La arquitectura funcional de Internet de las cosas es un modelo que puede ayudar a mapear las funciones de los subsistemas, las partes interesadas, sus interacciones y los requisitos técnicos correspondientes. Esta arquitectura funcional puede ser utilizada por diferentes expertos, como los expertos empresariales en el campo (PYME). (Geng, 2017)

Tabla 1

Arquitectura IoT de cuatro capas.

Arquitectura IoT de cuatro capas	
Capa de detección	La capa de detección permite detectar los incidentes de datos por medio de sensores, objetos físicos.
Capa de Intercambio de Datos	Transferencia de información viable por medio de redes de comunicación para el intercambio de datos.
Capa de integración de la información	Proceso de investigación obtenida del circuito de red de comunicación, integrando los datos para los usuarios finales.
Capa de servicio de aplicación	Brinda al usuario la posibilidad de acceder al contenido de la aplicación.

Fuente: Salazar & Silvestre (2018)

1.3.3. Ecosistema de tecnologías de IoT

Se conforma mediante el de nivel de dispositivo, datos, negocio y usuarios. Micrsoft Azure (2020) plan los ecosistemas de tecnologías de IoT por niveles:

Nivel de dispositivos

Acoplamiento de recursos de hardware y software para la interconexión de las puertas de enlace que conforman los dispositivos que mantienen una conexión e interacción en la red de comunicación.

Nivel de datos

Transferencia de información obtenida, siendo procesada y analizada por medio de un circuito de red.

Nivel de negocio

Procesos empresariales en IoT, que administra los diferentes procesos del negocio mediante los datos obtenido. Un ejemplo de ello es Marketplace.

Nivel de usuario

Interacción de los usuarios con los diversos dispositivos conectados a IoT.

1.3.4. Protocolos en internet de las cosas

Los dispositivos de IoT mantienen una comunicación mediante protocolos. El Protocolo de Internet (IP) es conformado por reglas que determinan el procese de envió de información a Internet. Los protocolos de IoT generan garantía que los dispositivos, puertas de enlace o servicios lean y comprendan la información enviada por otros dispositivos o sensores. Se diseñan y optimizan distintos protocolos de IoT en diferentes contextos. Teniendo en cuenta la gran variedad de dispositivos disponibles en IoT, se indica que el protocolo a implementar debe ser adecuado según al contexto en el que se encuentra.

1.3.4.1. Nivel de aplicación

La capa de aplicación proporciona comunicación entre usuarios y dispositivos tecnológicos con protocolos de IoT específicos.

Protocolo de cola de mensajes avanzado (AMQP)

Crea un nivel de software de interoperabilidad. Ayudando a que varias aplicaciones del sistema se comuniquen para crear mensajes en la industria. (Micrsoft Azure, 2020)

Servicio de distribución de datos (DDS)

Es un protocolo de comunicación multifuncional que realizar todas las operaciones, desde la compilación con pequeños dispositivos hasta la conexión con redes de alto rendimiento tecnológico. DDS simplifica la ejecución, mejorando la fiabilidad y reduciendo la dificultad. (Micrsoft Azure, 2020)

Clase de transporte

La capa de transporte protege y habilita la red de comunicación para la transmisión de datos entre las capas.

Protocolo de control de transmisión (TCP)

Es el protocolo en las diversas conexiones de internet. Dominando frecuentemente en la mayoría de las conexiones y que a su vez proporciona comunicación entre hosts, este protocolo también diversos y grandes conjuntos de datos en paquetes para reensamblarlos según sea necesario y reenviar a su destino. (Micrsoft Azure, 2020)

1.3.4.2. Nivel de red

La capa de red permite la comunicación con un solo dispositivo y el enrutador.

Internet Protocol (**IP)**

Es conjunto de normas de la información mediante internet o una red local. Los diversos protocolos de IoT IPv4, pero existen nuevos protocolos de implementación basado en IPv6. Estas nuevas actualizaciones de IP redirigen el tráfico de internet ubicando y dirigiendo los dispositivos en la red. (Micrsoft Azure, 2020)

6LoWPAN

Es un protocolo de IoT siendo más eficiente para dispositivos con limitaciones como bajo consumo de capacidades de procesamiento.

1.3.4.3. Nivel de enlace de datos

Es la capa encargada de trasmitir los datos de información fiable, por el circuito de red. La capa de datos es parte del protocolo IoT, tiene poder de detectar y corregir errores hallados en el nivel de la capa física. (Micrsoft Azure, 2020)

IEEE 802.15.4

Es un estándar para conexiones inalámbricas con una potencia baja. Por ello es recomendable utilizarlo con Zigbee, 6LoWPAN y otros estándares para así crear redes integradas.

LPWAN

La red de área amplia de bajo consumo (LPWAN), son diversas tecnologías que permite la comunicación desde 500 metros hasta un rango mayor a 10 kilómetros en determinadas coordenadas. LOng RAnge Wide Area Network (LoRaWAN) es un ejemplo de red LPWAN ya optimizada para un bajo consumo de energía. (Micrsoft Azure, 2020)

1.3.4.4. Nivel físico

Es el medio de comunicación de los dispositivos transformando a la secuencia de bits para dirigirlo a su lugar de destino.

Ethernet

Es una tecnología tradicional de conexión por cable, se considera que es la elección de menor costo que favorece la conexión más veloz de data con un bajo lapso.

Evolución a largo plazo (LTE)

Es un método de comunicarse de manera inalámbrica que brinda mayor velocidad en la transferencia de data por las telecomunicaciones. LTE permite secuencia de difusión y multidifusión debido al incremento de su velocidad en las redes inalámbricas.

Power Line Communication (PLC)

Tecnología de comunicación que consiente enviar y recibir información de datos por medio de cables eléctricos. Controla un dispositivo tecnológico de IoT por la red cableada.

Identificación por radiofrecuencia (RFID)

Es un protocolo que emplea diversos ambientes electromagnéticos realizando seguimiento exhaustivo de la alimentación de otro nodo por las etiquetas electrónicas. La compatibilidad con el hardware genera energía para la comunicación con las etiquetas con la finalidad de leer la información de autenticidad y de identificación. (Micrsoft Azure, 2020)

Wi-Fi/802.11

Wi-Fi/802.11 es una propuesta de conectividad usual en lugares como las oficinas y los hogares. Se presenta como la elección más económica, pero regularmente no es ajustables a todos los escenarios debido al consumo interrumpido de energía eléctrica y su trascendencia restringida. (Micrsoft Azure, 2020)

Z-Wave

Red en malla que, debido a su baja energía para la comunicación de un dispositivo con otro, es que implementa ondas de radio. Realizando así una comunicación fluida y viable. (Micrsoft Azure, 2020)

1.3.5. Dispositivos de internet de las cosas

El artefacto de Internet de las cosas está compuesto por un objeto que proporciona conectividad a Internet mediante un software inteligente. Puede

medir parámetros físicos u operar de forma remota, generando así un ambiente de servicios en el contorno implementado. (TEC, 2020)
tipos de dispositivos que son utilizados.

Sensores y actuadores

Sensores

Se define como un instrumento químico o físico. Que muchas veces pueden dar sensación a las máquinas y los objetos midiendo parámetros de energía eléctrica, temperatura, entre otros. (TEC, 2020)

Actuador

Es un dispositivo experto en la transformación de energía de una acción para así automatizar el proceso.

Etiquetas inteligentes (smart tags)

En la actualidad las empresas, para conocer el detalle de los costos de producción están obligadas a documentar toda la información, por ello se requiere un seguimiento y control del proceso de producción.

Etiquetas impresas

Son etiquetas pegada o impresa en el mismo producto que lo identifica únicamente de la categoría y tipo de producción. Las etiquetas más implementadas son las de código QR o de barras.

Etiquetas RFID

Es la etiqueta más sencilla de como identificar un objeto o producto ya que tienen instalado un chip pequeño y también una antena que permite redactar alguna información por medio de un lector, sin mantener algún tipo de contacto visual directo. es muchas ocasiones es posible que almacena información de su producción en este tipo de etiquetas.

1.3.6. Seguridad en internet de las cosas

Shea & Wigmore (2020) define a la seguridad en IoT como la parte de la tecnología que se centra en proteger las redes y los dispositivos conectados en el Internet de las cosas (IoT). Internet de las cosas implica agregar conectividad a Internet a sistemas de dispositivos informáticos interconectados, máquinas mecánicas y digitales, objetos, animales y personas. Cada “cosa” tiene un identificador único y tiene la capacidad de transmitir datos automáticamente a través de la red. Si el dispositivo no está protegido adecuadamente, permitir que el dispositivo se conecte a Internet lo expone a muchas vulnerabilidades graves. En términos generales se indica que la seguridad en IoT hace referencia con el procedimiento en la protección de la información con los dispositivos conectados en la red.

Cifrado de Datos

Es la transformación de la información en base código, al texto cifrado o en código se le conoce como encriptamiento o cifrado de datos. se utiliza para que la información sea indescifrable para usuarios o personal no autorizado. (Access Quality, 2021)

Cifrado asimétrico

El cifrado asimétrico mediante algoritmos utiliza dos claves relacionadas entre sí, una es pública la cual se emplea para encriptar y otra que es privada para el usuario de destino pueda descifrar los datos. Estas claves se utilizan para encriptar o desencriptar la información, esto posibilita que el proceso de transmisión de información sea veloz y segura. (Access Quality, 2021)

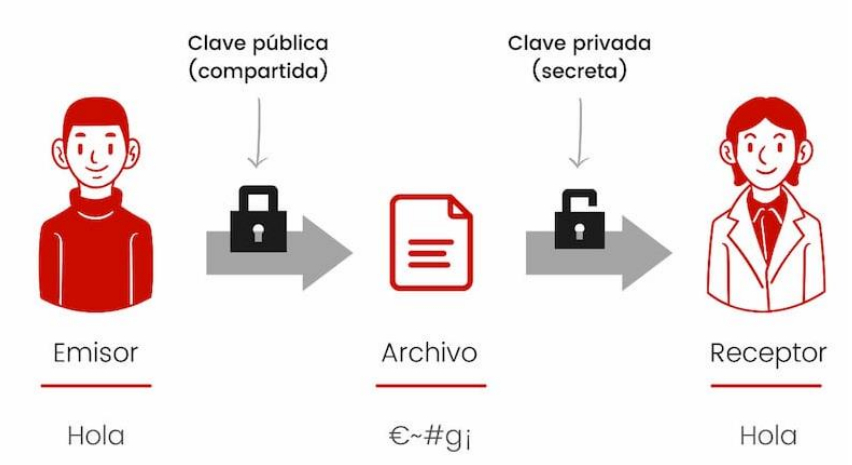


Figura 3. ¿Cómo funciona la criptografía asimétrica? Fuente: *(Ramírez, 2020)*

Algoritmos Asimétrico

RSA

Excelente algoritmo asimétrico basado en el par de claves públicas y privadas. La seguridad en el cual se enfoca este algoritmo consiste en descomponer enteros muy grandes porque, aunque es posible el descifrado parcial, hoy en día RSA se manifiesta como un algoritmo imposible que se logre descifrar los datos en su totalidad. Algunas de sus principales características de RSA son la longitud de la clave. Considerando que en la actualidad se recomienda utilizar como mínimo 2048 bits, aunque lo considerable es 4096 bits o más garantizando así una mayor seguridad. (López, 2021)

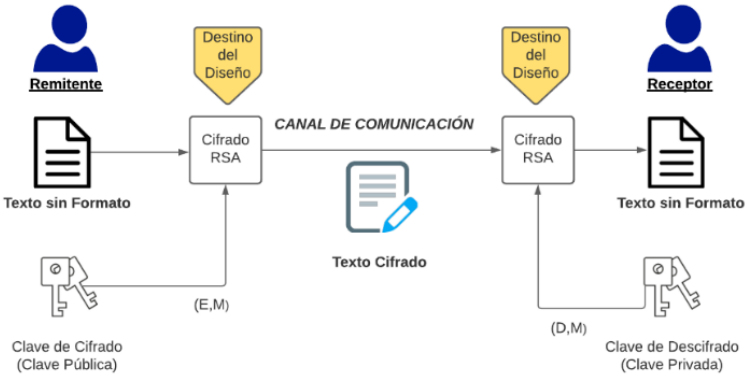


Figura 4 Estructura de algoritmo RSA. Fuente: (Bodur & Resul, 2015)

El Gamal

El cifrado El Gamal es un criptosistema de clave pública. Utiliza cifrado de clave asimétrica para comunicarse entre dos partes y cifrar mensajes. El criptosistema se basa en la dificultad de encontrar logaritmos discretos en el grupo cíclico Incluso si conocemos g a y g k, es difícil calcular g ak. (GeeksforGeeks, 2021)

ECC

La criptografía de curva elíptica es una familia de criptosistemas de claves públicas moderna que se enfoca en la estructura algebraica de curvas elípticas en campos finitos. ECC se considera el sucesor moderno natural del criptosistema RSA, porque ECC utiliza claves y firmas más pequeñas que RSA para lograr el mismo nivel de seguridad y proporciona una generación de claves muy rápida, un acuerdo de claves y firmas rápidos. (Cryptobook, 2021)

Cifrado simétrico

El cifrado simétrico por medio de algoritmo utiliza solo una clave para la encriptación de la transformación de la información legiblemente una vez sido recibido por el destinatario. Este procedimiento permite que el cifrado de los datos se realice de manera rápida y eficiente, pero es de importancia que la clave sea única y solo esté disponible para usuarios autorizados. (Access Quality, 2021)

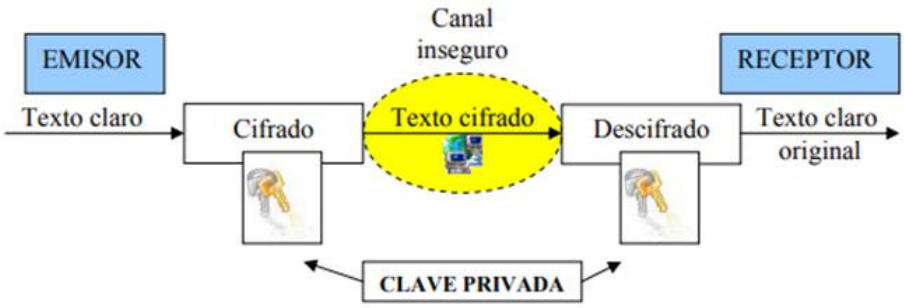


Figura 5. Esquema Simétrico. Fuente: *(OSI, 2019)*

Algoritmos Simétricos

AES (Advanced Encryption Standard)

Es un algoritmo de encriptamiento en bloques por el cual el tamaño de bloque fijo es de 128 bits. En el cual la longitud de la clave es

opcional, ya que brinda opciones de 128 bits, 192 bits y 256 bits. Siendo 128 bits es la longitud estándar, pero también se usa ampliamente 256 bits. AES es responsable de generar una matriz 4 × 4 y luego aplicarle una colección de rondas para encriptar los datos. Se específica que, las claves de 128 bits se aplican 10 turnos de encriptación, las claves de 192 bits se aplican 12 turnos y para las claves de 256 bits los turnos recomendados que se deben aplicar 14. (López, 2021)

MODOS DE OPERACIÓN

AES 256 GCM

AES-GCM o AES en Galois / Counter Mode es un algoritmo de cifrado autenticado, o más específicamente un algoritmo AEAD (authenticated encryption with related data). Estos tipos de algoritmos son convenientes porque brindan protección de confidencialidad e integridad en un paquete ordenado. El texto sin formato pasa por el algoritmo una vez, y la salida es el texto cifrado y la etiqueta de autenticación. Viceversa, el texto cifrado y la etiqueta de autenticación esperada pasan el algoritmo y la salida es texto sin formato, o si la etiqueta calculada no coincide con la etiqueta esperada, se producirá un error de descifrado. (Ponkanen, 2020)

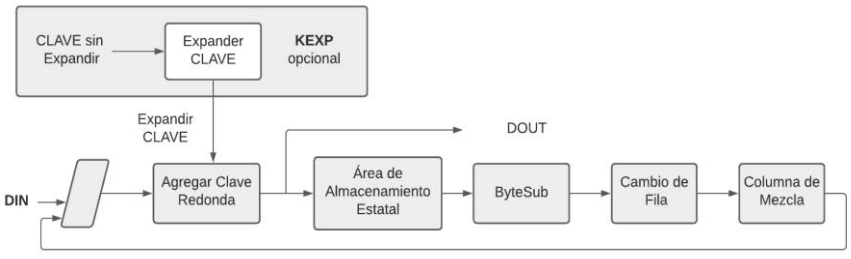


Figura 6 Esquema de operación de AES-GCM. Fuente: (Rosales, 2020)

AES 256 CBC

El modo Cipher Block Chaining (CBC) es un método de cifrado en bloques, que utiliza un algoritmo de cifrado de bloques. En

esta versión, proporcionamos capacidades de procesamiento de estándares de DES y AES. La longitud de la clave de cifrado de DES debe ser de 64 bits y AES de 128/192/256 bits. Otra limitación es que nuestro modo de trabajo funciona en unidades de tamaño fijo (1 bloque es de 64 o 128 bits), pero el texto en el mundo real tiene múltiples longitudes. Por lo tanto, antes del cifrado o descifrado, el último bloque del texto proporcionado a la primitiva debe rellenarse a 128 bits. (Xilinx, 2020)

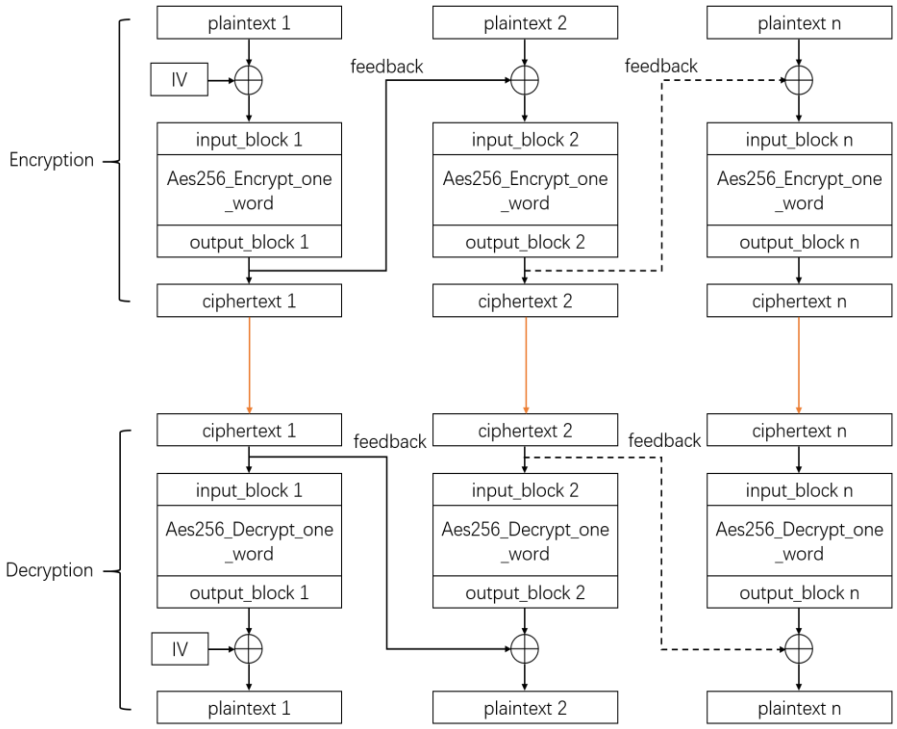


Figura 7

Figura 7 Diagrama de flujo del algoritmo AES-CBC. Fuente: (Xilinx, 2020)

CHACHA20

Es un algoritmo de cifrado de flujo el cual admite claves de alta velocidad de 128 y 256 bits. Su código ha sido publicado y estandarizado por el IETF en RFC 7539 e implementaciones de software, el cual se identifica por su eficiencia y rapidez y por ellos es considerado mejor que AES, ganando así un buen lugar por su trayectoria hoy en día. ChaCha20 se usa ampliamente para conexiones HTTPS, conexiones SSH para administrar servidores

logrando a su vez también administras al protocolo WireGuard VPN el cual lo utiliza solamente para el cifrado de datos simétrico. (López, 2021)

Cifrado de bloques

El cifrado de bloque utiliza un algoritmo determinista y una clave simétrica para cifrar los datos en el bloque.

El cifrado en bloque de 128 bits trae texto sin formato de 128 bits y lo cifra en texto cifrado de 128 bits. Cuando la cantidad de texto sin formato es inferior a 128 bits. El cifrado de bloques tiene las ventajas de una alta proliferación y una fuerte anti-manipulación sin ser detectado. Su desventaja es que la velocidad de cifrado es lenta, porque se debe capturar todo el bloque para cifrar / descifrar. Los cifrados de bloque también pueden generar errores, porque los errores en un solo símbolo pueden cambiar todo el bloque. (Essex, 2020)

1.4. Formulación del Problema.

¿Cuál es la técnica de cifrado más eficiente para el intercambio de datos en IoT en el ámbito de la Salud?

1.5. Justificación e importancia del estudio.

Con esta investigación se pretende evaluar y determinar cuál de las diferentes técnicas de cifrado es la más eficiente para el intercambio de datos en IoT en el ámbito de la Salud, este tema está embarcado en la línea de investigación de Infraestructura, Tecnología y Medio Ambiente; en su prioridad temática Internet de las Cosas. Con este trabajo se pretende demostrar que técnica de cifrado es la más eficiente para el cifrado de datos en el Internet de las Cosas.

Hoy en día hay una gran cantidad de libros, tesis e investigaciones, como la de (Perera Bartual, 2018) que utiliza técnicas de criptografía que permiten la interpretación de los datos sólo tras la correspondiente autenticación.

Esta investigación tiene como objetivo comprender la efectividad de las tecnologías que se utilizan siempre para la seguridad de IoT. Para ello, se escogieron dos tecnologías conocidas a nivel mundial para efectuar pruebas, análisis y ataques con el fin de poder medir la seguridad que brindan en función de sus mecanismos de seguridad. Al comprender su “fuerza” de esta manera, puede decidir si la seguridad que ofrecen a la indagación es incondicional o en qué condiciones se violan.

Asimismo, concede conocimiento para que los profesionales de la seguridad puedan elegir entre tecnologías de seguridad conocidas a nivel mundial para IoT u otras tecnologías de seguridad sin afectar la posibilidad de diseñar e implementar su propio Internet de las cosas. Esto impone a los ingenieros de sistemas no solo a diseñar el sistema en detalle, sino también a especificar sus especificaciones y la potencia de argumentación en el uso de instrumentos lógicos, de cálculo y matemáticas, y el embelesamiento de sucesiones difíciles para estimular inéditos algoritmos, Sistemas, y la facultad de englobar nuevas estructuras de procesar información y / o conocimiento.

1.6. Alcance

En el extenso mundo del internet de las cosas donde personas, sensores, micro controladores están conectados, este trabajo de investigación desarrollará un escenario de pruebas que se enfocará en evaluar la temperatura de un paciente debido a la llegada del Covid -19, con el fin de:

Evaluar cuál es la técnica de cifrado más eficiente en el intercambio de datos en Internet de las cosas en el ámbito de la Salud.

1.7. Hipótesis.

La técnica de cifrada híbrida es la más eficiente para el intercambio de datos en ambientes IoT en el ámbito de la Salud.

1.8. Objetivos.

1.8.1. Objetivo general.

Evaluar las técnicas de cifrado para el intercambio de datos en IoT en el ámbito de la Salud.

1.8.2. Objetivos específicos.

a) Seleccionar las técnicas de cifrado de datos.
b) Identificar los criterios de evaluación de las técnicas de cifrados de datos.
c) Diseñar un escenario de pruebas para evaluar la técnica de cifrado de datos seleccionada.
d) Aplicar las técnicas seleccionadas de cifrado de datos en el escenario de pruebas.
e) Evaluar los resultados obtenidos.

II. MATERIAL Y MÉTODO

2.1. Tipo y Diseño de Investigación.

Tecnológica-Aplicada (Vargas Cordero 2009) después de organizar y ejecutar el desarrollo en la investigación realizada, se busca adquirir o emplear otros conocimientos mientras se solicita adquirir o aplicar la información obtenida. La implementación de la información, conocimiento y resultados obtenidos en la investigación causó una manera drástica, pero a su vez organizada y minuciosa de entender el contexto de la realidad.

2.1.1. Diseño de la investigación

Es aquella que dirige la variable independiente deliberadamente para obtener su efecto o relación con otras variables. En el diseño cuasi experimental los grupos de investigación están formados antes del experimento se realizan comparaciones de respuestas. (Sampiere, 2010).

2.2. Población y muestra.

2.2.1. Población

La población está compuesta por cinco técnicas de cifrado para el intercambio de datos en IoT en salud. Para obtener la población se realizó la

revisión de artículos científicos de las bases de datos científicas IEEE Xplore y ProQuest, en el cual se obtuvo los resultados de la aplicación que realizaron. En sucesiva la tabla 2 muestra el listado de las técnicas de cifrado.

(Abdullah, Hamad, Abdulrahman, Moala, & Elkhediri, 2019) señala 5 técnicas de cifrado en IoT.

Tabla 2.

Técnicas de cifrados de datos

N.º	Población
1)	Técnica de Cifrado de Bloques
2)	Técnica de Cifrado por Flujos
3)	Técnica de Cifrado Simétrico
4)	Técnica de Cifrado Asimétrico
5)	Técnica de Cifrado Híbrido

Fuente*:* Elaborado propia.

2.2.2. Muestra

Con referencia a la Muestra, el tipo de muestreo es por conveniencia debido al interés de estudio se eligió cuatro técnicas de seguridad para el cifrado de datos en IoT.

1) Técnica de Cifrado Asimétrico
2) Técnica de Cifrado Simétrico
3) Técnica de Cifrado por bloques
4) Técnica de Cifrado híbrido

2.3. Operacionalización.

Tabla 3. Operacionalización por tipo de variable

.

Variables	Dimensión	Indicador	Ítem	Técnica e instrumentos de recolección de datos
Variable Independiente Técnicas de cifrado para el intercambio de datos	Eficiencia	Tiempo de Procesamiento	$x = \frac{X_1 + X_2 + X_3 + \cdots + x_n}{N}$	(Registro electrónico) observación, ficha técnica
		Margen Error – En el intercambio	$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - x)^2}{N}}$	
		Longitud de Clave	N°Bits	
		Rendimiento	$R = \frac{D}{E}$	
Variable Dependiente Intercambio de datos en Internet de las Cosas en el ámbito de la Salud		Cantidad datos encriptados	$\Sigma^n = (n_1 + n_2 + n_3 + \cdots + n)$	
		Cantidad desencriptados	$\Sigma^n = (n_1 + n_2 + n_3 + \cdots + n)$	

Fuente*:* Elaboración propia

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

En el trabajo de investigación realizado, emplea la técnica de observación por el cual se determinó que el instrumento a emplear es la ficha técnica. Por consiguiente, se expone en qué consiste dicha técnica y el instrumento.

Observación

La técnica se enfoca en observar eventos para registrar información y analizarla; conviene recolectar más datos sobre las variables de investigación. Como en esta investigación, el propósito es observar la eficiencia de la tecnología de encriptación en el intercambio de datos del Internet de las Cosas en el campo de la salud, y evaluarla en el área de diseño mediante simulación.

Instrumento de recolección de datos

Registro Electrónico

Se realizará la toma de datos percibidos en el sistema para el análisis de los resultados obteniendo la verdad o falsedad de la hipótesis planteada del tema de investigación indicado.

Ficha de Registro

La herramienta utilizada en este estudio es un registro de información porque es una herramienta prediseñada que se enfoca en los aspectos a observar. Su propósito es registrar datos de manera ordenada. En este caso, los resultados obtenidos de la tecnología de encriptación utilizada para intercambiar datos en Internet de las Cosas en el ámbito de la salud se registran en base a los indicadores establecidos para evaluar su eficiencia.

2.5. Procedimiento de análisis de datos.

La investigación se enfoca en la aplicación de técnicas o instrumentos de recolección de datos:

a) Tabulación de datos

Manejo de los datos en tablas estadísticas.

Manejar los gráficos estadísticos que se procesan como producto del procesamiento de los datos obtenidos de la ejecución y procesamiento de la prueba.

b) Análisis de datos

Interpretar los indicadores según la prueba a realizar.

Sumatorias: También conocida como notación Sigma, es una operación matemática que puede representar la suma de múltiples sumandos, o incluso un número infinito de sumandos. Su fórmula es la siguiente: $\sum n$ = (n1 + n2 + n3+. +n)

Media Aritmética: el resultado obtenido es debido a la sumatoria de todos los datos y dividiendo por el número total de datos.

Fórmula:

$$x = \frac{X_1 + X_2 + X_3 + \cdots + x_n}{N}$$

Donde: X1= T1; T1=TF-TI

X1: Prueba n°1;

T1: Tiempo 1 obtenido por la prueba 1.

TF: Tiempo final.

TI: Tiempo inicial.

Desviación Estándar: la fórmula permitirá establecer el margen de error para la seguridad del intercambio de datos.

$$\sigma = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \cdots + (x_n - \bar{x})^2}{N}}$$

Ejemplo: Se indica que tiene el tiempo de procesamiento para cada prueba realizada en diferentes pacientes, para lo cual tiene: 6 pruebas y su tiempo de procesamiento de referencia.

0.5; 0.3; 0.5; 0.4;0.2;0.9

$x = \frac{0.5+0.3+0.5+0.4+0.2+0.9}{6}$ = 2.8

$$\sqrt{\frac{(0.5-2.8)^2+(0.3-2.8)^2+(0.5-2.8)^2+(0.4-2.8)^2+(0.2-2.8)^2+(0.9-2.8)^2}{6}}$$

=2,343%

En este caso, el margen de error generado por la desviación estándar es de 2.343%, lo que determina el menor margen de error que pueda existir en el intercambio de datos.

Rendimiento:

El algoritmo de cifrado debe realizar la tarea a la velocidad y debe tener el tiempo necesario para el cifrado o descifrado.

D: Tiempo de procesamiento de descifrado en segundos.

E: Tiempo de procesamiento de cifrado en segundos

$$R = \frac{D}{E}$$

2.6. Criterios éticos.

Se considera uno de los principios éticos más importantes en el ser persistente en participar en la investigación, comprendiendo de la información que se le ha otorgado.

Confiabilidad:

El proyecto de investigación tendrá como estrategia proteger los documentos elaborados por el investigador manteniendo las normas y valores de un profesional.

Derechos de autor:

El proyecto elaborado respeta la auditoría de cada investigación elaborada para respaldar la veracidad de la información, los cuales estarán citados y referenciados en el siguiente proyecto de investigación.

2.7. Criterios de Rigor Científico.

El proyecto de investigación estará sujeto a criterios de veracidad de la información, confiabilidad de los instrumentos y así mismo a los protocolos de investigación de la universidad.

2.7.1. Validez

Para validar nuestro proyecto de investigación se especificará en tablas de Operacionalización el cual nos ayudará a medir las variables y poder ser analizados por el investigador.

2.7.2. Fiabilidad

El proyecto de investigación tendrá resultados consistentes debido a que hará uso de variables y fórmulas establecidas que nos ayudará a reducir los errores y obtendremos datos más confiables para el desarrollo de la investigación.

2.7.3. Transferibilidad

El proyecto de investigación tendrá como parte de la investigación la obtención de varios reportes e informes de distintos investigadores para poder obtener similitudes de la investigación que se está elaborando para el tema de investigación.

2.7.4. Consistencia

El proyecto de investigación se fundamenta con pruebas consistentes y demostrables como son los artículos científicos mencionados en el siguiente informe de investigación.

III. RESULTADOS.

3.1. Resultados en Tablas y Figuras.

En esta investigación se evaluaron técnicas de cifrado de datos simétrica, asimétrica, de bloques e hibrido, debido a que cada técnica cuenta con su propia estructura de cifrado y descifrado de las claves, se determinó implementar los algoritmos de cifrado para realizar pruebas por cada técnica evaluando el tiempo de procesamiento, margen de error y rendimiento.

El tiempo de procesamiento estadísticamente se denota de la media aritmética que comprende de la sumatoria del tiempo obtenida en segundos entre el número de pruebas realizadas. Las pruebas que se realizaron se desarrollaron bajo las mismas condiciones para las evaluaciones a cada uno las técnicas de cifrado. En el cual estas consisten que son 300 datos transferidos y todas mediante 1 paquete.

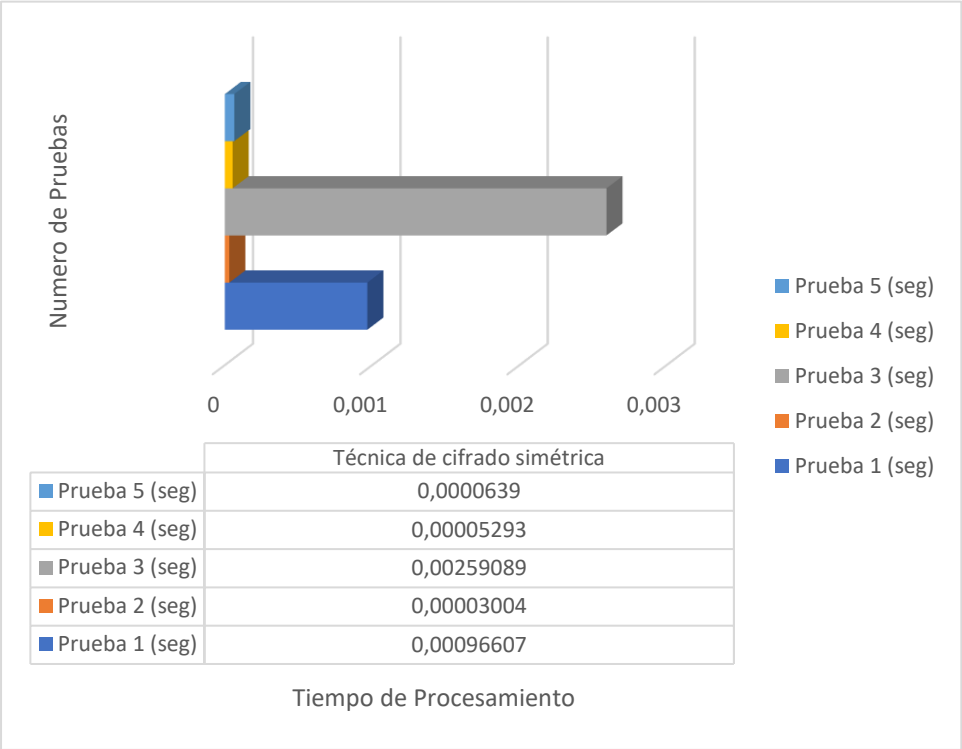


Figura 8. Tiempo de procesamiento de transferencia de datos de la técnica de cifrado simétrica. Fuente: Elaboración propia.

La figura 8. Muestra el tiempo de procesamiento de trasferencia de datos en segundos por cada prueba realizada de la técnica de cifrado simétrica, que se realizó mediante el algoritmo AES 256 GCM con 16 bits de longitud de claves. De esta forma se cuantifica el tiempo de procesamiento de datos en cada prueba. En el cual se observa que la prueba 2 es la que obtuvo el menor tiempo de procesamiento de transferencia de datos siendo este de 0.00003004 segundos.

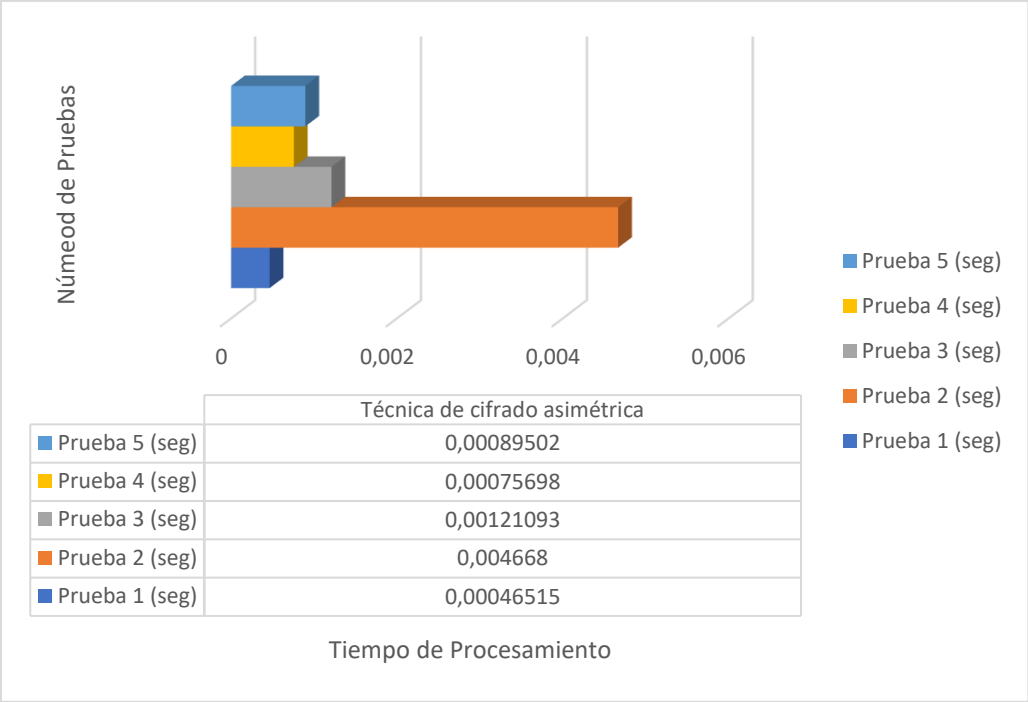


Figura 9. Tiempo de procesamiento de transferencia de datos de la técnica de cifrado asimétrica. Fuente: Elaboración propia.

En la figura 9. Se observa los resultados de tiempo de procesamiento en segundos de cada prueba realizada para la técnica de cifrado asimétrica que se implementó el algoritmo SHA 512 Y RSA con 2048 bits en longitud de claves. De esta manera se identifica que la prueba número 1 es la que obtuvo un tiempo de 0.00046515 seg. siendo calificada como la de menor tiempo de procesamiento en la transferencia de datos. Ya que luego se observa que el tiempo de procesamiento demoró milisegundos más.

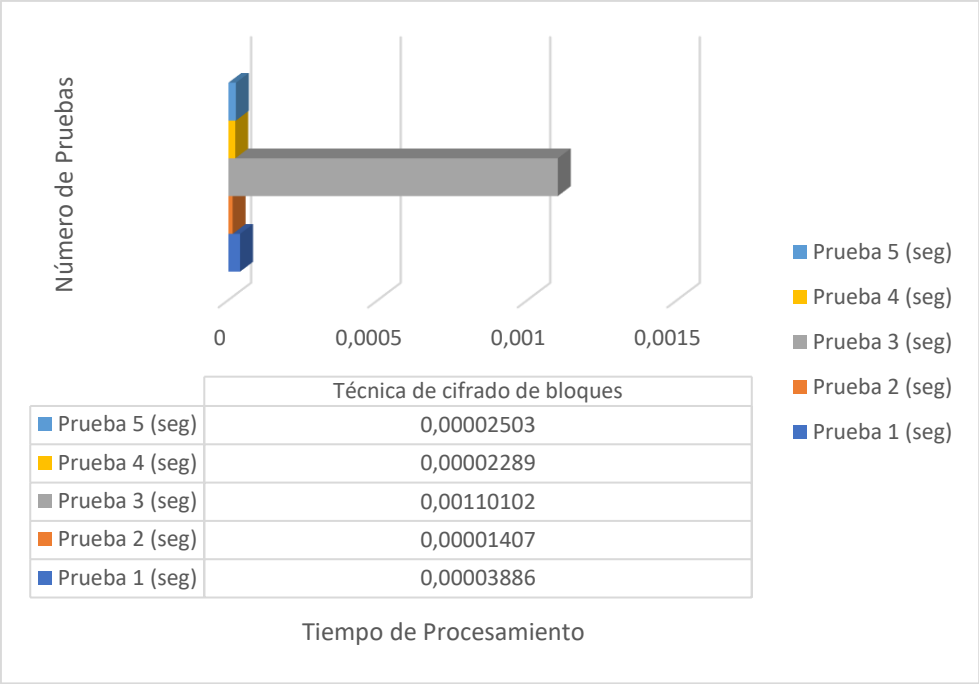


Figura 10 Tiempo de procesamiento de transferencia de datos de la técnica de cifrado de bloques. Fuente: Elaboración propia.

En la figura anterior se muestra los resultados obtenidos en cada prueba realizada para evaluación de la técnica de cifrado de bloques mediante el algoritmo AES 128 CBC con 16 bits de longitud de claves. En el cual tiempo fue obtenido en segundos y se identifica que en la prueba 2 es la que obtuvo 0.00001407 seg considerándose la prueba con menor tiempo obtenido, mientras que en la prueba 3 es la más demoró en el procesamiento de transferencia de datos ya que obtuvo 0.00110102 seg.

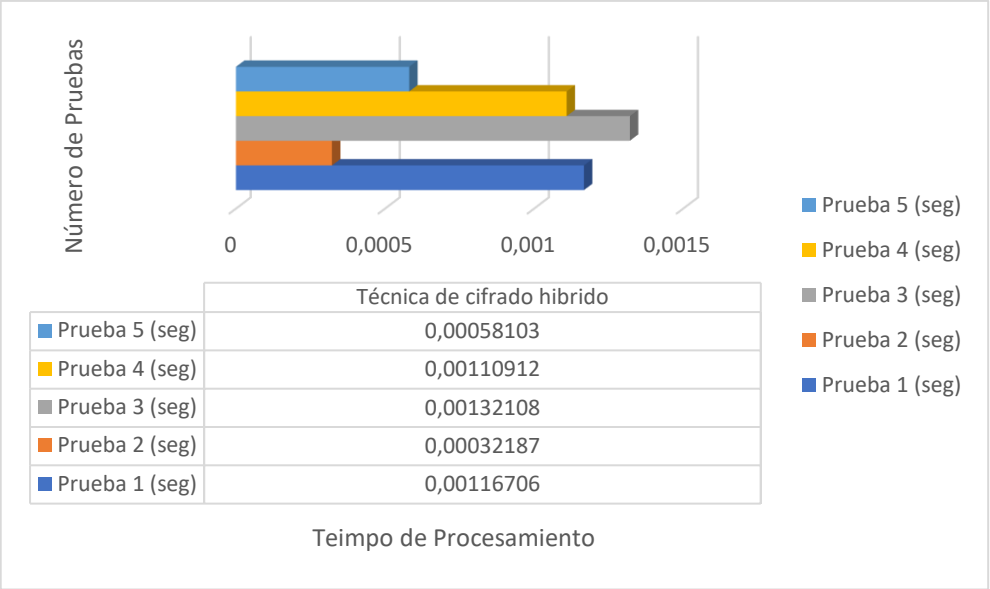


Figura 11 . Tiempo de procesamiento de transferencia de datos de la técnica de cifrado hibrido. Fuente: Elaboración propia.

En la figura 11. Nos muestra los resultados del comportamiento del tiempo en segundos del procesamiento en transferencia de datos de la técnica de cifrado hibrido, las pruebas realizadas se obtuvieron bajo implementación del algoritmo de cifrado SHA 256 con 2047 bits de longitud de claves. En el cual se manifiesta que la prueba con menor tiempo de procesamiento de datos es la prueba 2 con 0.00032187 segundos y la prueba 3 tomo unos milisegundos más obteniendo 0.00132108 segundos.

Margen de Error

El margen de error en la transmisión de datos es evaluado para determinar el porcentaje de error que se tiene por cada técnica de cifrado para la transmisión de datos. En cual se determinará cuál de las técnicas evaluadas (simétrica, asimétrica, bloques e hibrida) es la que obtiene menor margen de error garantizando así que es la más segura en transmisión de datos. cabe indicar que las pruebas se desarrollaron bajo las mismas condiciones para cada una de las técnicas, por lo que se preparó 1 paquete con 300 datos.

Tabla 4

Margen de Error en la transmisión de la técnica de cifrado simétrica

TIEMPO DE PROCESAMIENTO (seg)					
1° Técnica de Cifrado Simétrica					
AES 256 GCM	**Prueba 1**	**Prueba 2**	**Prueba 3**	**Prueba 4**	**Prueba 5**
Tiempo de Procesamiento	0.00096607	0.00003004	0.00259089	0.00005293	0.00006390
Total, de Tiempo de Procesamiento	**0.00370383**				
Margen Error en la transmisión	**0.00140142%**				

Fuente: Elaboración propia.

$$\sigma = \sqrt{\frac{(x_1-\bar{x})^2+(x_2-\bar{x})^2+\cdots+(x_n-\bar{x})^2}{N}}$$

$$\frac{x = \ 0.00096607 + \ 0.00003004 + 0.00259089 + 0.00005293 + \ 0.00006390}{5}$$

$$x = \ 0{,}00074077$$

$$= \sqrt{\frac{(0.00096607-\bar{x})^2+(0.00003004-\bar{x})^2+ (0.00003004-\bar{x})^2+(0.00005293-\bar{x})^2+(0.00006390-\bar{x})^2}{5}}$$

$$= \sqrt{\frac{(0.0002253)^2 + (-0{,}00071073)^2 + \ (0{,}00185012)^2 + (-0.00068784)^2 + (-0.00067687)^2}{5}}$$

$$= \frac{\sqrt{0.0000491}}{5} \quad x = 0.00140142\%$$

Para la técnica de cifrado simétrica, se realizó cinco pruebas bajo el algoritmo de cifrado AES 256 GCM con 16 bits de longitud de claves. En el cual para obtener el margen de error se tomó en cuenta la suma total de tiempo de procesamiento y después del cálculo realizado se obtuvo 0.00140142% de margen de error.

Tabla 5

Margen de Error en la transmisión de la técnica de cifrado asimétrica

TIEMPO DE PROCESAMIENTO (seg)					
	2° Técnica de Cifrado Asimétrica				
SHA 512 Y RSA	**Prueba 1**	**Prueba 2**	**Prueba 3**	**Prueba 4**	**Prueba 5**
Tiempo de Procesamiento	0.00046515	0.004668	0.0012109 3	0.0007569 8	0.0008950 2
Total, de Tiempo de Procesamiento			**0.00799608**		
Margen Error en la transmisión			**0.05018000%**		

Fuente: Elaboración propia.

$$\sigma = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \cdots + (x_n - \bar{x})^2}{N}}$$

$$\frac{x = 0.00046515 + 0.00466800 + 0.00121093 + 0.00005293 + 0.00089502}{5}$$

$$x = 0.00159922$$

$$= \sqrt{\frac{(0.00046515-\bar{x})^2+(0.00466800-\bar{x})^2+(0.00121093-\bar{x})^2+(0.00005293-\bar{x})^2+(0.00089502-\bar{x})^2}{5}}$$

$$= \sqrt{\frac{(-0.000113407)^2+(0.00306878)^2+(0{,}0002509)^2+(-0.00154629)^2+(-0.0007042)^2}{5}}$$

$$= \frac{\sqrt{0.06296312}}{5}$$

$$x = 0.05018\%$$

En la evaluación realizada para la técnica de cifrado asimétrica se implementó el algoritmo de cifrado SHA 512 Y RSA con 2048 bits de longitud de claves. En el cual se obtuvo un 0.05018% de margen, por lo que se consideraría que la técnica asimétrica tiene 99.9498200% de seguridad en la transmisión de datos.

Tabla 6

Margen de Error en la transmisión de la técnica de cifrado de bloques

.

TIEMPO DE PROCESAMIENTO (seg)					
	3° Técnica de Cifrado de Bloques				
AES 128 CBC	**Prueba 1**	**Prueba 2**	**Prueba 3**	**Prueba 4**	**Prueba 5**
Tiempo de Procesamiento	0.00003886	0.00001407	0.00110102	0.00002289	0.00002503
Total, de Tiempo de Procesamiento			**0.00120187**		
Margen Error en la transmisión			**0.00043767%**		

Fuente: Elaboración propia.

$$\sigma = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \cdots + (x_n - \bar{x})^2}{N}}$$

$$\frac{x = \ 0.00003886 \ + \ 0.00001407 \ + 0.00110102 \ \ + 0.00002289 + \ 0.00002503}{5}$$

$$x = 0.00031984$$

$$= \sqrt{\frac{(0.00003886 - \bar{x})^2 + (0.00001407 - \bar{x})^2 + (0.00110102 - \bar{x})^2 + (0.00002289 - \bar{x})^2 + (0.00002503 - \bar{x})^2}{5}}$$

$$= \sqrt{\frac{(0.00028098)^2 + (0.00030577)^2 + \ (0.00078118)^2 + (0.00029695)^2 + (0.00029481)^2}{5}}$$

$$x = 0.00043767\%$$

Para la evaluación realizada para la técnica cifrado de bloques se implementó bajo el algoritmo de cifrado AES 128 CBC con 16 bits de longitud de claves. Por el cual para obtener el margen de error se tomó en cuenta el total de tiempo de procesamiento de cada prueba realizada, obteniendo así 0.00043767% de margen de error lo que se considera que la técnica de bloques garantiza 99.9498200% de seguridad en la transmisión de datos.

Tabla 7

Margen de Error en la transmisión de la técnica de cifrado híbrida

TIEMPO DE PROCESAMIENTO (seg) **4° Técnica de Cifrado Hibrido**					
SHA 256	**Prueba 1**	**Prueba 2**	**Prueba 3**	**Prueba 4**	**Prueba 5**
Tiempo de Procesamiento	0.00116706	0.00032187	0.00132108	0.00110912	0.00058103

Total, de Tiempo de Procesamiento	**0.00450016**
Margen Error en la transmisión	**0.00038166%**

Fuente: Elaboración propia.

$$\sigma = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \cdots + (x_n - \bar{x})^2}{N}}$$

$$\frac{x = \ 0.00116706 \ + \ 0.00032187 \ + 0.00132108 + 0.00110912 \ + \ 0.00058103}{5}$$

$$x = \ 0.000900032$$

$$= \sqrt{\frac{(0.00116706 \ - \bar{x})^2 + (0.00032187 - \bar{x})^2 + (0.00132108 \ - \bar{x})^2 + (0.00110912 - \bar{x})^2 + (0.00058103 - \bar{x})^2}{5}}$$

$$= \sqrt{\frac{(0.000267028)^2 + (0.000578162)^2 + \ (0.000421048)^2 + (0.000209088)^2 + (0.000319002)^2}{5}}$$

$x = 0.00038166\%$

Durante la evaluación realizada para la técnica de cifrado hibrido se implementó el algoritmo de SHA 256 con 2047 bits de longitud de claves, se tomó en cuenta el tiempo de procesamiento que se realizó a cada una de las pruebas por lo que de acuerdo con ello se obtuvo 0.00038166% de margen de error, por lo que se considera que la técnica de cifrado hibrido obtiene un 99.8798130% de seguridad en la transmisión de datos.

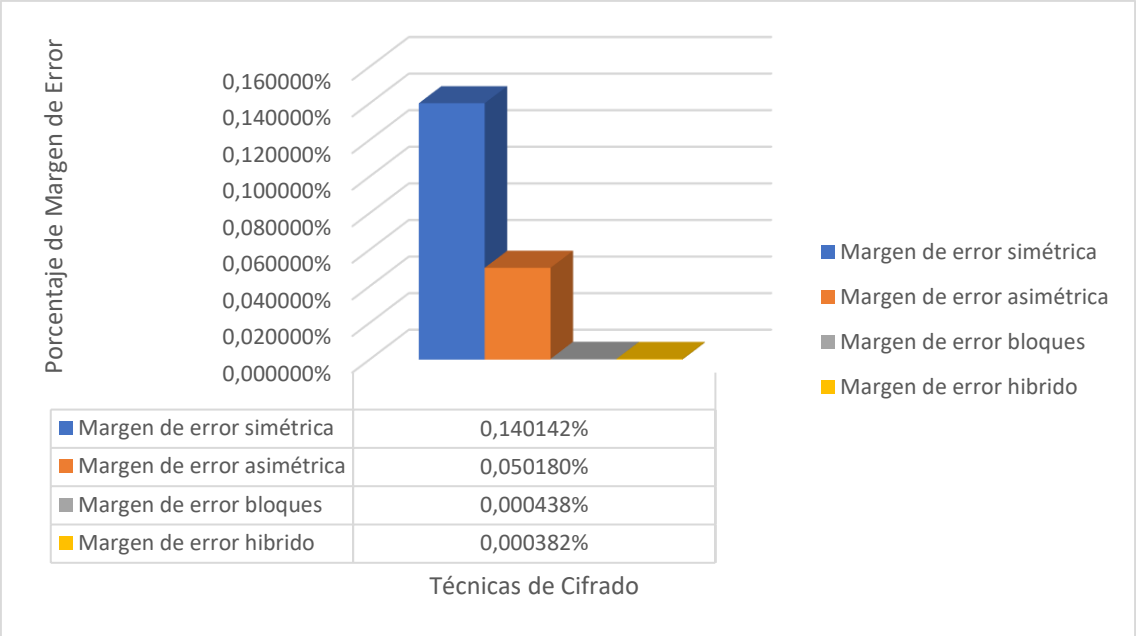


Figura 12 Margen de error de las técnicas de cifrado Fuente: Elaboración propia.

En la figura 12 “Resultado de la comparación de técnicas de cifrado” se observa evidentemente que el algoritmo híbrido proporciona menor margen de error en comparación con los algoritmos de cifrado globalmente conocidos. Y es que en seguridad de datos mientras menor sea el margen de error es más eficiente, por lo que según los datos obtenidos la técnica de cifrado hibrido nos garantiza un 99.8798130% en seguridad para la transmisión de datos.

El rendimiento es calculado para identificar la utilidad y velocidad de cada una de las técnicas de cifrado al desarrollar su tarea de cifrar y descifrar. Para ello, se desarrolla una división del tiempo de velocidad de descifrado y del tiempo de velocidad de cifrado.

Tabla 8

Descripción de las variables de la fórmula

Rendimiento de la técnica de cifrado simétrica	
D: Tiempo de velocidad del descifrado en segundos	E: Tiempo de velocidad del cifrado en segundos

$R = \frac{D}{E}$

Fuente: Elaboración propia.

Tabla 9

Rendimiento de la técnica de cifrado simétrica

1° Técnica de Cifrado Simétrica	**Tiempo de Velocidad del descifrado en seg.** Prueba 1	**Tiempo de velocidad del cifrado en seg.**
AES 256 GCM	0.00002098	0.00006890
Rendimiento	$R = \frac{D}{E}$ = 0.3044%	

Fuente: Elaboración propia.

Tabla 10.

Rendimiento de la técnica de cifrado asimétrica

2° Técnica de Cifrado Asimétrico	**Tiempo de Velocidad del descifrado en seg.** Prueba 1	**Tiempo de velocidad del cifrado en seg.** Prueba 1
SHA 512 Y RSA	0.00156903	0.00046015
Rendimiento	$R = \frac{D}{E}$ = 0.293%	

Fuente: Elaboración propia.

Tabla 11.

Rendimiento de la técnica de cifrado bloques

3° Técnica de Cifrado por bloques	Tiempo de Velocidad del descifrado en seg Prueba 1	Tiempo de velocidad del cifrado en seg Prueba 1
AES 128 CBC	0.00001788	0.00004387
Rendimiento	$R = \frac{D}{E}$ = 0,407%	

Fuente: Elaboración propia.

Tabla 12

Rendimiento de la técnica de cifrado híbrido

4° Técnica de Cifrado híbrido	Tiempo de Velocidad del descifrado en seg Prueba 1	Tiempo de velocidad del cifrado en seg Prueba 1
SHA 256	0.00140095	0.00038695
Rendimiento	$R = \frac{D}{E}$ = 0.276%	

Fuente: Elaboración propia.

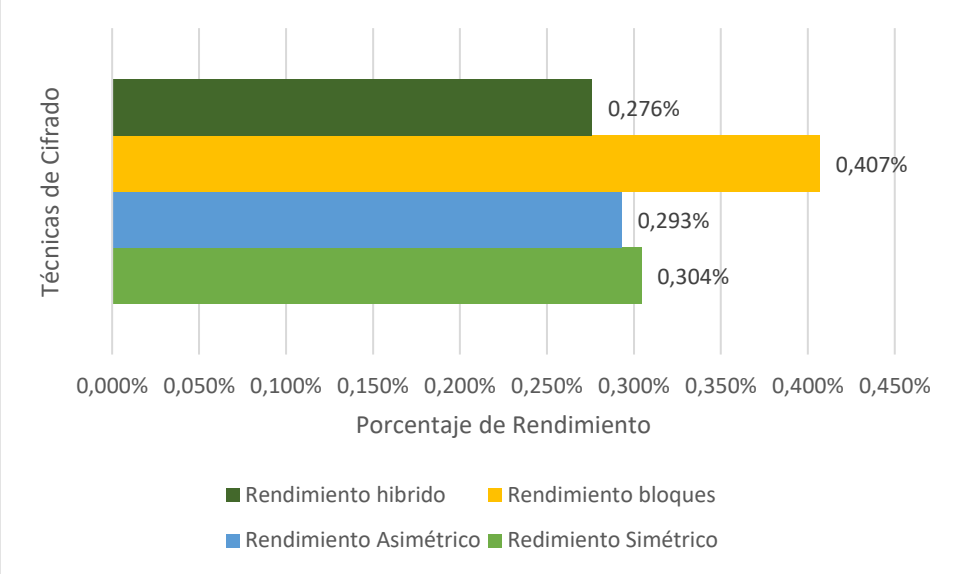


Figura 13 Rendimiento de las técnicas de cifrado. Fuente: Elaboración propia.

En la figura 13 “Resultado de la comparación de rendimiento de técnicas de cifrado” se observa evidentemente que el algoritmo híbrido proporciona menor

velocidad en el desarrollo de cifrado y descifrado en comparación con los algoritmos de cifrado globalmente conocidos.

Tabla 13

CUADRO COMPARATIVO DE EVALUACIÓN DE TÉCNICAS DE CIFRADO PARA EL INTERCAMBIO DE DATOS EN INTERNET DE LAS COSAS EN EL ÁMBITO DE LA SALUD

	Técnica de Cifrado Simétrica	**Técnica de Cifrado Asimétrica**	**Técnica de Cifrado Bloques**	**Técnica de Cifrado Híbrida**
Rendimiento	0.3044%	0.293%	0,407%	0.276%
Tiempo de procesamiento	0.00003004 seg	0.00046515 seg.	0.00001407 seg	0.00032187 seg
Margen de Error	0.00140142%	0.05018000%	0.00043767%	0.00038166%

Fuente: Elaboración propia.

3.2. Discusión de resultados.

Tras haber aplicado las técnicas de cifrado para el intercambio de datos en IoT en Salud, cabe mencionar que la técnica de cifrado híbrida con el algoritmo SHA 256 es muy eficiente, con mejores resultados estadísticos en tiempo de procesamiento de cifrado y en rendimiento; resultados similares fueron obtenidos por (Sadikin & Wardhani, 2016) en su trabajo de investigación, donde señala que la técnica de cifrado híbrido en combinación con el algoritmo SHA 256, se implementa en la programación Java garantizando la integridad, eficiencia y no rechazo, por lo que al igual en este trabajo de investigación, se advierte que en el trascurso del tiempo, este algoritmo ha tenido bastante acogida.

Así mismo, (Ravi Raushan & Kakali , 2020) en su trabajo de investigación, propone una técnica de cifrado híbrido para la transmisión segura de datos, en donde la transferencia de segura de datos de salud es esencial para que, al almacenar el registro del paciente, se debe mantener a salvo el uso indebido y la modificación de datos, ya que puede ser rastreado fácilmente y es muy difícil de cifrar los datos con técnicas de cifrado de alto nivel. El método propuesto se basa en una nueva técnica de cifrado híbrido, donde se centran en una función de rotación de matriz, XoR y expansión, el proceso depende principalmente de la función de expansión, función de generación de claves redondas, luego las técnicas de cifrado se implementan en el microcontrolador Arduino de 8 bits, se procesan datos de tamaño de bloque de 2kb para cifrado y se muestran los resultados experimentales; analizando exactamente el caso propuesto, no afecta la diferencia estadística en el caso propuesto, por lo que se resalta además que el cifrado híbrido SHA en tiempos de velocidad en cifrar son casi similares, ya que se muestra un tiempo de cifrado de 0.02112 seg.

Con respecto a los indicadores, tienen en cuenta otros como la velocidad de memoria, por lo que se resalta más que se mide en este trabajo de investigación

el rendimiento como un indicador clave para evaluar la técnica de cifrado más eficiente en IoT entorno a Salud.

3.3. Demostración de Hipótesis

La Hipótesis que deseamos demostrar es la siguiente:

"La técnica de cifrado híbrida es la más eficiente para el intercambio de datos en ambientes IoT en el ámbito de la Salud."

Para la aprobación de la validez de la hipótesis planteada tenemos que recordar que:

La eficiencia de los recursos de un algoritmo, este suele medirse en función del parámetro: rendimiento, por lo tanto, para demostrar el rendimiento propuesto el algoritmo de cifrado debe realizar la tarea a la velocidad y debe tener el tiempo necesario para el cifrado o descifrado.

Para la medida teórica se midió el tiempo de procesamiento de descifrado en segundos y el tiempo de procesamiento de cifrado en segundos.

RENDIMIENTO

Rendimiento de la técnica de cifrado híbrido

4° Técnica de Cifrado híbrido	**Tiempo de Velocidad del descifrado en seg**	**Tiempo de velocidad del cifrado en seg**
	Prueba 1	Prueba 1
SHA 256	0.00140095	0.00038695
Rendimiento	$R = \frac{D}{E}$ = 0.276%	

Fuente: Elaboración propia.

En la tabla 12 se evidencia el rendimiento de ejecución del cifrado propuesto resultando evidentemente que el algoritmo híbrido proporciona menor velocidad en el desarrollo de cifrado y descifrado en comparación con los algoritmos de cifrado globalmente conocidos.

3.4. Aporte práctico.

Seleccionar las técnicas de cifrado de datos

Para la selección de las Técnicas de cifrado de datos, se realizó un proceso de revisión de artículos científicos en la base de datos IEEE Xplore y ProQuest, en donde se separa los términos utilizados en el PICOC para definir el ámbito de la revisión, a efectos de responder a las preguntas de investigación.

Tabla 13

Descripción de términos - PICOC

N°	PICOC	DESCRIPCIÓN
01	Population(P)	Artículos científicos relacionados sobre las técnicas de cifrado.
02	Intervention (I)	Realizar una revisión sistemática de la literatura sobre las técnicas de cifrado en el intercambio de datos en Salud.
03	Comparison (C)	Si se desea aplicar comparación.
04	Outcomes (O)	Técnicas de cifrado en el intercambio de datos en Salud.
05	Context (C)	Industria y empresas de Salud en el cifrado de datos.

Fuente: Elaboración propia.

Posteriormente, se formulan preguntas de investigación que tienen como objetivo, identificar las lagunas de conocimiento y las necesidades de investigación en el área concreta.

Tabla 14

Preguntas de Investigación

N°	Research Questions
01	¿Qué son las técnicas de cifrado?
02	¿Cuántos estudios importantes relacionados a las técnicas de cifrado en el intercambio de datos en Salud se han realizado?
03	¿Cuáles son los autores más reconocidos en el área de estudio?
04	¿Qué es el intercambio de datos?
05	¿Cómo funciona las técnicas de cifrado en el intercambio de datos?

Fuente: Elaboración propia.

Luego, se realizó una descripción para habilitar los campos con palabras claves, sinónimos y relacionado con intervención o población.

Tabla 15

Palabras claves de la Investigación.

Palabra Clave	**Sinónimos**	**Relacionados con**
Técnicas de cifrado	Encriptación	Intervención
Intercambio	Cambio	Población
Datos	Nota	Intervención
Salud	Sanidad	Población
Internet de las Cosas	Inteligencia Distribuida	Población

Fuente: Elaboración propia.

Después, de definir las palabras en español, se traducen las palabras en inglés, debido a que en las bases de datos científicas de IEEE Xplore y ProQuest, las publicaciones se encuentran en inglés.

Asimismo, se generó la cadena de búsqueda, utilizando los operadores booleanos (Y, O), comillas dobles para las palabras compuestas y paréntesis para separar lógicamente las palabras claves y sinónimos.

Tabla 16

Cadena de Búsqueda

Base de Datos	Búsqueda
IEEE Xplore	Título: (“Encryption techniques OR for IoT)
ProQuest	Título: (“encryption techniques” OR “for IoT”) AND (“encryption techniques” OR “For data)

Fuente: Elaboración propia

Posteriormente, se agrega criterios de inclusión y exclusión, ya que es importante los criterios de aceptación o de rechazo de cada artículo del conjunto de datos que se está manejando.

Tabla 17

Criterios de Inclusión y Exclusión

Criterios de Inclusión	Criterios de Exclusión
Alta conferencia	El artículo científico no es accesible
Artículo en inglés	No cumple objetivos

El artículo presenta una experiencia con resultados	El artículo no presenta un estudio aprobado

Fuente: Elaboración propia

Para la lista de verificación de evaluación de la calidad, se formulan preguntas de los artículos seleccionados.

Tabla 18

Preguntas de evaluación de calidad de artículos de investigación

N°	Preguntas
01	¿El artículo científico reúne la información necesaria para el entendimiento de cualquier lector?
02	¿Tiene un impacto significativo en el campo en el que se investiga?
03	¿Se detalla claramente los objetivos de la investigación?
04	¿Se usan técnicas estadísticas para analizar datos?
05	¿El problema está contextualizado correctamente?

Fuente: Elaboración propia

Se realizó una descripción de valor

Tabla 19

Descripción de valor de calificación.

Descripción	Peso
Si	2.5
Regular	1.5

No	0.5

Nota: Valores de calificación en base a la investigación para la selección de calidad de los artículos de investigación.

Fuente: Elaboración propia

Finalmente, se verifica los puntajes de evaluación de calidad

Máximo puntaje: 15.0

Puntuación de corte: 9.0

Estudios de encontrados

Tabla 20

Estudios encontrados en Base de datos

Fuente	**Estudios Encontrados**
Biblioteca digital IEEE	100
ProQuest	100

Fuente: Elaboración propia

Finalmente, para la elección de la investigación, considerando la investigación publicada de 2015 a 2021, también se requiere que estos artículos describan claramente el método de obtención de los resultados. De esta manera, se seleccionaron 4 técnicas de cifrado basadas en el mejor desempeño probado.

Identificar los criterios de evaluación de las técnicas de cifrados de datos.

A continuación, en la tabla 26. Se realizó un análisis de comparación de los algoritmos de criptografía simétrica: AES, 3DES, DES. En el cual se tomó en cuenta once factores de evaluación que son de mucha importancia e influyentes en la ejecución de los algoritmos de cifrado.

Tabla 21

Técnicas de cifrado de datos seleccionadas para la evaluación

N°	Nombre de Técnica	Autor
01	Técnica de Cifrado de bloques	Mohd, B. J., & Hayajneh, T. (2018). Lightweight Block Ciphers for IoT: Energy Optimization and Survivability Techniques. IEEE Access, 6, 35966–35978. doi:10.1109/access.2018.2848586
02	Técnica de Cifrado Simétrico	R. Hussain e I. Abdullah, "Revisión de diferentes técnicas de cifrado y descifrado utilizadas para la seguridad y privacidad de IoT en diferentes aplicaciones", *Conferencia internacional IEEE 2018 sobre ingeniería de redes de energía inteligente (SEGE)*, 2018, págs. 293-297, doi: 10.1109 / SEGE.2018.8499430.
03	Técnica de Cifrado Asimétrico	Bhandari, R., & Kirubanand, V. B. (2019). Enhanced encryption technique for secure iot data transmission. *International Journal of Electrical and Computer Engineering, 9*(5), 3732-3738. doi: http://dx.doi.org/10.11591/ijece.v9i5.pp3732-3738
04	Técnica de Cifrado híbrida	Vijayalakshmi, A. V., & Arockiam, L. (2018). *Hybrid security techniques to protect sensitive data in E-healthcare systems. 2018 International Conference on Smart*

Systems and Inventive Technology (ICSSIT). doi:10.1109/icssit.2018.8748298

Nota: Se seleccionaron las técnicas de cifrado más utilizadas en el intercambio de datos. (Sarin, A., Thanawala, D., Verma, S., & Prakash, C. 15 de octubre de 2020). Fuente: *(Chala, 2019)*

Identificar los criterios de evaluación de las técnicas de cifrados de datos.

A continuación, en la tabla 27. Se realizó un análisis de comparación de los algoritmos de criptografía simétrica: AES, 3DES, DES. En el cual se tomó en cuenta once factores de evaluación que son de mucha importancia e influyentes en la ejecución de los algoritmos de cifrado.

Tabla 22

Comparación de diferentes algoritmos de criptografía de clave simétrica

FACTORES	AES	3DES	DES
Longitud de la clave	128,192 y 256 bits	K1, K2 YK3 168 bits (k1yk2 168 bits (k1yk2 es mismo) 112bits	56 bits
Tipo de cifras	Simétrica-bloques de Cifrado	Simétrica-bloques de Cifrado	Simétrica-bloques de Cifrado
Tamaño de bloque	128, 192 y 256 Bits	64 BITS	64 bits
Año de desarrollo	2000	1978	1977
Posibles claves	2128, 2192 y 95 192	2 112 , y 2 168	257 957

Posibles teclas de caracteres ASCII imprimir	9515, 9524 Y 9532	95 1,4 y 95 21	
Imprimible Revisión toda la llave en 50 mil millones de claves	Para una clave de 128 Bits 5 x10 Años	Para Una Clave de 112 Bits 800 Dias	Para una clave de 56 bits 400 días
Ronda	10 (128-Bist) ,12 (192 Bits) , 14 (256 Bits)	48	16
Rendimiento (encriptación desencriptación)	4.174 / 6.452	3.45/ 5. 665	4.01/6347
Clave	Solo	Sola (dividida en 3 partes	Sola

Fuente: (Medina & Miranda, 2015)

En la tabla 28. Se describe un análisis comparativo de los algoritmos de criptografía asimétrica: RSA, ElGamal, ECC. Por lo que se determinó 4 propiedades que influyen en la evaluación y ejecución de cada algoritmo de cifrado.

Tabla 23

Comparación de diferentes algoritmos de criptografía de clave asimétrica.

Propiedades	RSA	El gamal	ECC
Tamaño de la clave	1024 bits	1024 bits	160 bits
Estructura del Algoritmo	Entero Factorización	Discreto Logarítmico	Curva elíptica

Desarrollador por	Rivest, Shamir, y Adleman,1977	Taher elgamal,1985	Neal Koblitz y víctor S. Miller,1985
Objetivo	Utilizaciones dos teclas dispares: pública y privada clave para cifrado y decodificación, por separado, envista de considerando de enorme prima números, descubriendo p y q para el mod n dado es simple sin embargo lo contrario es molesto, indefenso contra pequeños números	En vista de discreto problema de logaritmo, difícil de rastrear por el discreto logaritmo de a número dado tiempo de inspección span, produce distintivo texto cifrado para un dado texto plano, lento y crea texto cifrado largo	A la luz de la curva elíptica, aplicar discreto logaritmo ase centra en la curva elíptica, pequeño tamaño de clave, menos registrando poder, costoso."

Fuente: (Kumar, Chidananda, A Noola, Nirmal, & Sivakumar, 2021)

Diseñar un escenario de pruebas para evaluar la técnica de cifrado de datos seleccionada.

La atención médica, requiere la ayuda de técnicas de cifrado para ocultar los datos personales de los pacientes.

Con la llegada de la Pandemia de Covid-19 esta ha elaborado un estado en la que los ciudadanos, empresas, administraciones, y entidades con

personalidad jurídica, han de poner en práctica una serie de medidas de higiene, protección y distanciamiento.
Debido a la Pandemia de COVID-19, se ha visto la dificultad de cómo proteger a los trabajadores de este riesgo ocurrido, en situaciones de contacto físico.

Debido a toda la situación el sistema de control de acceso está siendo adoptado con mucha importancia por entidades públicas, por ejemplo, el Consejo General del Poder Judicial lo recomienda para el acceso de sedes judiciales cuando se reúnen las actuaciones con un umbral de 37,5°C, y también se utiliza por muchas entidades privadas para el control de acceso.

Para este trabajo se hará uso de la toda la información recolectada teniendo como resultado de que uno de los indicadores más frecuentes observados es la presencia de fiebre, que usualmente todo es superior a 125°C, por todo lo expuesto, se considera pertinente poner en marcha un sistema de control de accesos basado en toma de temperatura y transmisión de datos, para ello se realiza la creación de la página web, en el que se observa la data que se envía cuando se toma el sensor de temperatura y se muestra el cifrado en la página web y la temperatura desencriptada.

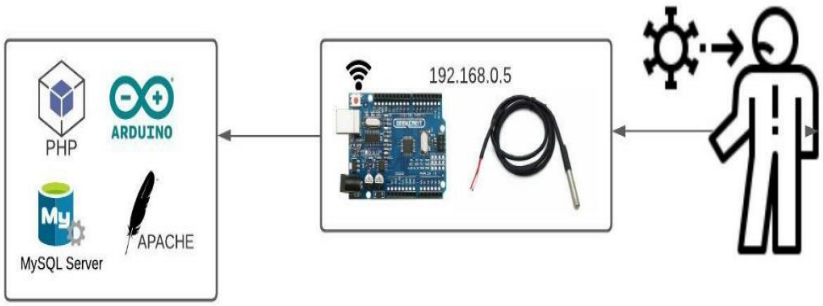


Figura 14. *Caso de estudio. Fuente: Elaboración propia.*

Para el diseño y construcción del escenario de pruebas se tiene en cuenta los siguientes componentes:

- **Sensor de Temperatura DS18B20**

  Le permite medir fácilmente temperaturas hasta 125ºC, y además está sellado en un paquete impermeable para que pueda sumergirse en líquido o protegerse de los elementos. Dado que es un sensor digital, la señal de lectura no se reducirá debido a la distancia del cableado. (BricoGeek, 2021)

Figura 15 . Sensor de temperatura. Fuente: Elaboración propia.

Características del cable:

Tubo de acero inoxidable de 6mm de diámetro por 30mm de largo
Largo: 91cm
Diámetro: 4mm
Contiene un sensor de temperatura DS18B20
Digital

- **Tarjeta Wifi (ESP8266)**

  Sistema de desarrollo ESP8266, interfaz USB, chip CH340. Tarjeta WiFi basada en chip ESP-8266, compatible con Arduino ™ y NodeMcu. Implementada para desarrollar aplicaciones para la IOT. (PlastimediaStudio, 2021)

Figura 16 . Tarjeta Wifi. Fuente: Elaboración propia.

Características:

WIFI NodeM ESP8266 + 32Mb

Incluye USB-TTL

11 pines GPIO

Cada pin GPIO puede funcionar como PWM, I2C, 1-Wire

Contiene antena PCB

Pulsador RST

Pulsador Flash

Conexión a red wifi (puede funcionar como punto de acceso o estación, alojar un servidor web)

Puede conectarse a internet para recuperar o cargar datos

Es programable por medio de la IDE de Arduino®

- **RESISTENCIA 500 OHM**

  la resistencia es una medida de la resistencia al flujo de corriente en un circuito. Se expresa en ohmios y se representa con la letra griega omega. (García, 2021)

Figura 17 Resistencia 500 OHM. Fuente: Elaboración propia.

Arquitectura diseño lógico

Conexión DS18B20 y Arduino

Representa el montaje físico del sistema, el cual está conformado por una tarjeta ESP32, un sensor de temperatura DS18B20, también de algunos componentes electrónicos necesarios para el circuito, tales como resistencias, una protoboard, un cable micro-usb y un led.

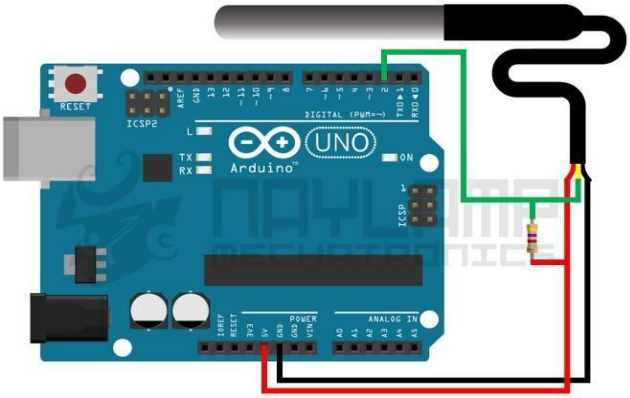


Figura 18. *Conexión DS18B20. Fuente: Elaboración Propia*

DISEÑO FÍSICO

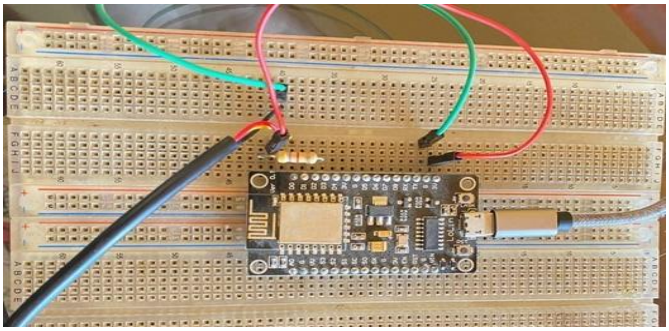


Figura 19 . Conexión Física del Arduino con los Sensores. Fuente: Elaboración Propia

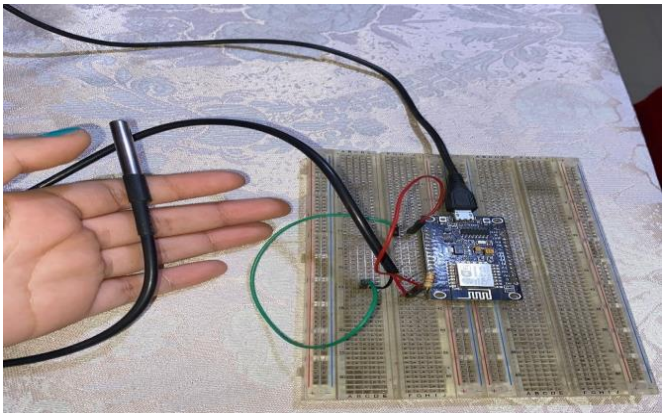


Figura 20. Conexión Física del Arduino con sensor temperatura. Fuente: Elaboración Propia

APLICAR LAS TÉCNICAS SELECCIONADA DE CIFRADO DE DATOS EN EL ESCENARIO DE PRUEBAS.

Programación ESP86266

Se enviará los datos mediante una red WIFI al respectivo canal, para ello en esta fase se configura la tarjeta de desarrollo ESP32 con todos estos datos de transmisión.

ARDUINO IDE 1.8.15

Para programar la tarjeta de desarrollo es necesario de un entorno de desarrollo (IDE), como en este caso, el IDE de código abierto Arduino.

Se importa las librerías necesarias para el uso de todos los elementos del sistema, como lo es la librería ESP8255Wifi.h, se necesitó dos variables para la red wifi y otra para el passowrd en la cual se va a conectar, se crea un objeto server de la clase wifi server en el puerto 80 para que funcione como un servidor web.

ARDUINOFINAL

```
#include <ESP8266WiFi.h>
const char* ssid     = "CECILIA"; // SSID
const char* password = "liskanajhy";                    // Password
const char* host = "192.168.1.38";                     // IP serveur - Server IP
const int   port = 80;                                 // Port serveur - Server Port
const int   watchdog = 5000;                           // Fréquence du watchdog - Watchdog frequency
unsigned long previousMillis = millis();
```

Figura 21. Arduino IDE- Importe librerías. Fuente: Elaboración Propia

Configurar red WIFI

Para configurar la red WiFi, se necesita conocer el ssid de nuestra red, cual es el nombre de esta y la contraseña en caso de tener una. Una vez obtenidos estos datos, se guarda como variables en código.

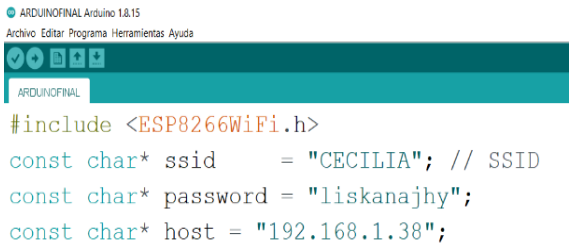


Figura 22. Arduino IDE-Wifi. Fuente: Elaboración Propia

Para la conexión a la red se utiliza la librería WiFi con un método Begin y se le pasa los argumentos de la red.

```
WiFi.begin(ssid, password);
```

Figura 23 . Arduino IDE-Wifi. Fuente: Elaboración Propia

En la primera parte del programa, se incluye librerías OneWire, ya que es la que implementa el protocolo 1-Wire, como DallasTemperature, que también se implementa en el código para enviar los comandos adecuados a los sensores y obtener la temperatura.

Librerías
#include **<OneWire**.h>
#include **<DallasTemperature**.h>

Función Setup ()

En este caso, lo primero que se hace en void. Setup inicia la comunicación serie, a 9600 bits de datos por segundo, entre la placa Arduino y el ordenador con la línea.

```
void setup(void) {  // inicia la comunicación serie
pinMode (pulsador, INPUT_PULLUP);
pinMode (led, OUTPUT);
```

```
delay(10);
Serial.println();
  Serial.print("Conectando a:\t");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED){
  delay (500);
  Serial.print(".");  }
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
```

Figura 24. Código fuente de función setup. Fuente: Elaboración Propia

Inmediatamente al instalar las librerías, se realiza la conexión anterior y se logra realizar las lecturas de temperatura, para eso usamos el siguiente sketch:

```
void enviardato (){
sensors.requestTemperatures();   //Se envía el comando para leer la temperatura
float t= sensors.getTempCByIndex(0); //Se obtiene la temperatura en ºC
float h= 28;
//float h = dht.readHumidity();
//float t = dht.readTemperature();
Serial.print("Humedad = ");
Serial.print(h);
Serial.print("%  ");
Serial.print("Temperatura = ");
Serial.print(t);
Serial.println("°C  ");
unsigned long currentMillis = millis();
Serial.println("enviando data ...  ");
if ( currentMillis - previousMillis > watchdog ) {
previousMillis = currentMillis;
WiFiClient client;
if (!client.connect(host, port)) {
Serial.println("Fallo al conectar"); return; }
```

Figura 25.Código fuente de obtener datos de temperatura. Fuente: Elaboración Propia

Después, se envía los datos por medio de la URL.

```
String url =
"http://192.168.1.38/ardu/guardar_data_arduino.php?temp=";
   url += t;    //variable a enviar
   url += "&hum=";
   url += h;    //variable a enviar
```

Figura 26.Código fuente de enviar datos Fuente: Elaboración Propia

```
    client.print(String("GET ") + url + " HTTP/1.1\r\n" +
                 "Host: " + host + "\r\n" +
                 "Connection: close\r\n\r\n");
  Serial.println("enviando data ...  ");
  unsigned long timeout = millis();
  while (client.available() == 0) {
    if (millis() - timeout > 5000) {
      Serial.println(">>> Client Timeout !");
      client.stop();
      return;
    }   }
```

Figura 27. Código fuente de enviar petición al servidor. Fuente: Elaboración Propia

```
// Leemos la respuesta del servidor
   while(client.available()){
     String line = client.readStringUntil('\r');
     Serial.println(line);
     digitalWrite(led, HIGH);
```

Figura 28Código fuente de leer respuesta del servidor. Fuente: Elaboración Propia

Posteriormente, se creó la base de datos en MySQL, para guardar los datos del sensor de temperatura y puedan ser cifrados.

	Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
☐	data_arduinos	Examinar Estructura Buscar Insertar Vaciar Eliminar	254	InnoDB	utf8_general_ci	272.0 KB	-
☐	estado	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_general_ci	16.0 KB	-
☐	historias	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_general_ci	16.0 KB	-
☐	pacientes	Examinar Estructura Buscar Insertar Vaciar Eliminar	5	MyISAM	utf8_unicode_ci	2.8 KB	-
☐	tipo_usuario	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	latin1_swedish_ci	16.0 KB	-
☐	usuario	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_general_ci	16.0 KB	-
	6 tablas	Número de filas	269	InnoDB	utf8mb4_general_ci	338.8 KB	0 B

Figura 29. Tablas de la base de datos. Fuente: Elaboración Propia

Se muestra, la creación de la base de datos y las tablas realizadas mediante el uso de comando SQL. En la siguiente imagen se muestra el código utilizado para la creación de las 6 tablas.

```
-- Base de datos: `Sistema_Monitoreo`
-- Estructura de tabla para la tabla `data_arduinos`
CREATE TABLE `data_arduinos` (
  `id_arduino` int(11) NOT NULL,
  `hume` varchar(255) DEFAULT NULL,
  `temp` varchar(255) DEFAULT NULL,
  `temp_hibrido` text DEFAULT NULL,
  `temp_asimetrico` text DEFAULT NULL,
  `temp_bloques` text DEFAULT NULL,
  `tiempo_simetrico` decimal(10,8) DEFAULT NULL,
  `tiempo_asimetrico` decimal(10,8) DEFAULT NULL,
  `tiempo_bloques` decimal(10,8) DEFAULT NULL,
  `tiempo_hibrido` decimal(10,8) DEFAULT NULL,
  `tiempo_desc_sincrono` decimal(10,8) DEFAULT NULL,
  `tiempo_desc_a_sincrono` decimal(10,8) DEFAULT NULL,
  `tiempo_desc_bloques` decimal(10,8) DEFAULT NULL,
  `tiempo_desc_hibrido` decimal(10,8) DEFAULT NULL,
  `fecha_registro` datetime DEFAULT NULL,
  `estado_idestado` int(11) DEFAULT NULL,
  `orden` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Figura 30Crear tabla de datos del obtenidos del Arduino. Fuente: Elaboración Propia

Creación de la tabla estado, historias, pacientes, tipo_usuario y usuario.

```
CREATE TABLE `estado` (
  `idestado` int(11) NOT NULL,
  `nombre` varchar(200) DEFAULT NULL,
  `valor` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Figura 31. Crear tabla de estado. Fuente: Elaboración Propia

```
CREATE TABLE `historias` (
  `idhistoria` int(11) NOT NULL,
  `idpaciente` int(11) NOT NULL,
  `fecha` datetime NOT NULL,
  `temperatura` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Figura 32.Crear tabla de historias. Fuente: Elaboración Propia

```
CREATE TABLE `tipo_usuario` (
  `idtipo_usu` int(11) NOT NULL,
  `nombre_tipousu` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Figura 33 Crear tabla de tipo de usuario. Fuente: Elaboración Propia

```
CREATE TABLE `usuario` (
  `idusuario` int(11) NOT NULL,
  `estado_idestado` int(11) NOT NULL,
  `idtipo_usu` int(11) DEFAULT NULL,
  `codusuario` varchar(100) DEFAULT NULL,
  `nomusuario` varchar(200) DEFAULT NULL,
  `email` varchar(255) DEFAULT NULL,
  `fecha_ingreso` datetime DEFAULT NULL,
  `contrasena` varchar(45) DEFAULT NULL,
  `orden` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Figura 34. Crear tabla de usuario. Fuente: Elaboración Propia

```
CREATE TABLE `pacientes` (
  `idpaciente` int(11) NOT NULL,
  `idusuario` int(11) DEFAULT NULL,
  `codigo` text COLLATE utf8_unicode_ci DEFAULT NULL,
  `tipo` int(11) DEFAULT 1 COMMENT '1. HOMBRE; 2. MUJER',
  `dni` text COLLATE utf8_unicode_ci DEFAULT NULL,
  `nombre` text COLLATE utf8_unicode_ci DEFAULT NULL,
  `apellidos` text COLLATE utf8_unicode_ci DEFAULT NULL,
  `nombre_completo` text COLLATE utf8_unicode_ci DEFAULT NULL,
  `nombre_rewrite` text COLLATE utf8_unicode_ci DEFAULT NULL,
  `temp` text COLLATE utf8_unicode_ci DEFAULT NULL,
  `telefono` text COLLATE utf8_unicode_ci DEFAULT NULL,
  `email` text COLLATE utf8_unicode_ci DEFAULT NULL,
  `cuenta` text COLLATE utf8_unicode_ci DEFAULT NULL,
  `iddpto` int(11) DEFAULT NULL,
  `idprvc` int(11) DEFAULT NULL,
  `iddist` int(11) DEFAULT NULL,
  `direccion` text COLLATE utf8_unicode_ci DEFAULT NULL,
  `clave` text COLLATE utf8_unicode_ci DEFAULT NULL,
  `pais` text COLLATE utf8_unicode_ci DEFAULT NULL,
  `ciudad` text COLLATE utf8_unicode_ci DEFAULT NULL,



  `imagen` text COLLATE utf8_unicode_ci DEFAULT NULL,
  `observacion` text COLLATE utf8_unicode_ci DEFAULT NULL,
  `ocupacion` text COLLATE utf8_unicode_ci DEFAULT NULL,
  `orden` int(11) DEFAULT NULL,
  `estado` int(11) DEFAULT NULL,
  `fecha_registro` date DEFAULT NULL,
  `estado_idestado` int(11) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

Figura 35.Crear tabla de paciente. Fuente: Elaboración Propia

- En la tabla, data_arduinos, se muestran los datos cifrados, ya sea simétrico, asimétrico, por bloques y híbridos.

```
SELECT * FROM `data arduinos` ORDER BY `id arduino`
```

ASIMETRICA	SIMETRICA	BLOQUES	HIBIRDA
VT+0exbNxXM86ehgR/TiyO...	nLGgBfVAZEZCxsz+HLdeQo...	bmF5aGVsaXRlc2lzMTIzNO...	mWrECSsK8EbRwEZ/7ICJaH...
jU8eKX+0uNBAQr9zCCSZJv...	nLGgBfVAZEZCxsz+HLdeQo...	bmF5aGVsaXRlc2lzMTIzNO...	BDVcAf/ZI0DRVgvcai2hrO...
mrVU+rShzntGtyzTFmxURC...	nLGgBfVAZEZCxsz+HLdeQo...	bmF5aGVsaXRlc2lzMTIzNO...	r9woOeDGukydCuzJX6JcJU...
pF+fNYwDBFfTKAb+xI57MJ...	nLGgBfVAZEZCxsz+HLdeQo...	bmF5aGVsaXRlc2lzMTIzNO...	JHnK1S1fDKcVtCXvFT4nrH...
Z0RqioUyHFOtvW9GmG8IB4...	nLGgBfVAZEZCxsz+HLdeQo...	bmF5aGVsaXRlc2lzMTIzNO...	D0sLyF2Peq9v354Ns1TUhT...
OjCwVVk8krmqlpWHDFfU3h...	nLGgBfVAZEZCxsz+HLdeQo...	bmF5aGVsaXRlc2lzMTIzNO...	GL1OVfgF09dyW9/XiaPdV6...
JfUSvwRwS4F7ZS7oBTLoFq...	nLGgBfVAZEZCxsz+HLdeQo...	bmF5aGVsaXRlc2lzMTIzNO...	hkI2TpMwtvwPEf5vWUMBey...
KnlAMl5uat9ZvDqIta4vux...	nLGgBfVAZEZCxsz+HLdeQo...	bmF5aGVsaXRlc2lzMTIzNO...	v3pPAZcma20DYzqcRquiOm...

Figura 36.Tablas de datos obtenidos del Arduino. Fuente: Elaboración Propia.

En Sublime Text, en guardar_data_arduino.php, se desarrolla el cifrado simétrico con el algoritmo AES 256 GCM.

```
/* CIFRADO SIMETRICO: AES 256 GCM */

                $time_inicio_simetrico = microtime(TRUE);
                $aesKey = 'nayhelitesis';
                $aesIv = '1234567812345678';
                $ciphertext = encrypt_aes_256_gcm($_GET["temp"], $aesKey, $aesIv);
                $_POST["temp"]= $ciphertext; /* compra minado completado */

                $time_end_simetrico = microtime(TRUE);
                $_POST["tiempo_simetrico"]=$time_sin = round($time_end_simetrico - $
                    time_inicio_simetrico, 8);
```

Figura 37. Código de algoritmo de cifrado AES 256 GCM. Fuente: Elaboración Propia

Después, dentro de la carpeta Class, se encuentra el archivo PHP: FUCTIONS, en donde se tiene la función para cifrar, y obtiene dos parámetros que es el mensaje y el segundo es la llave o la contraseña que se utilizará; luego se inicializa el vector para generar el cifrado, utilizando la función Openssl_Encrypt, en el que se coloca el mensaje a cifrar, el algoritmo del cifrado, la llave y la cadena de bytes que se obtiene en la línea anterior.

```php
/* Funcion SIMETRICA: encriptar y desencriptar: AES 256 GCM */

function encrypt_aes_256_gcm(string $plaintext, string $key, string $iv = '', string $
    aad = ''): string
{
    $ciphertext = openssl_encrypt($plaintext, 'aes-256-gcm', $key, OPENSSL_RAW_DATA, $
        iv, $tag, $aad, 16);

    if (false === $ciphertext) {
        throw new UnexpectedValueException('No se pudo cifrar la entrada $ texto sin
            formato, verifique su $ key y $ iv si son correctos o no.');
    }

    return base64_encode($ciphertext . $tag);
}
```

Figura 38. Código fuente de cifrado de la técnica simétrica. Fuente: Elaboración Propia

Se tiene la función descifrar, en donde se obtiene parámetros del mensaje que ya está cifrado y la contraseña para poder descifrarlos. Openssl_decrypt funciona.

```php
function decrypt_aes_256_gcm(string $ciphertext, string $key, string $iv = '', string $
    aad = ''): string
{
    $ciphertext = base64_decode($ciphertext);
    $authTag = substr($ciphertext, -16);
    $tagLength = strlen($authTag);

    /* Manually checking the length of the tag, because the `openssl_decrypt` was
    mentioned there, it's the caller's responsibility. */
    if ($tagLength > 16 || ($tagLength < 12 && $tagLength !== 8 && $tagLength !== 4)) {
        throw new RuntimeException('Las entradas `$ texto cifrado` están incompletas,
            la longitud de los bytes debe ser uno de 16, 15, 14, 13, 12, 8 o 4.');
    }

    $plaintext = openssl_decrypt(substr($ciphertext, 0, -16), 'aes-256-gcm', $key,
        OPENSSL_RAW_DATA, $iv, $authTag, $aad);

    if (false === $plaintext) {
        throw new UnexpectedValueException('No se pudo descifrar el $ cifrado de
            entrada, verifique su $ key y $ iv si son correctos o no.');
    }

    return $plaintext;
}
```

Figura 39.Código fuente de descifrado de la técnica simétrica. Fuente: Elaboración Propia

En guardar_data_arduino.php, se desarrolla el cifrado asimétrico con el algoritmo SHA 512 y RSA, en donde tenemos microtime para que devuelva la marca de tiempo actual con micro-segundos, en la siguiente línea se tiene RsaUtils que consiste en utilizar diferentes claves de cifrado y claves de descifrado.

```php
/* CIFRADO ASIMETRICO:  SHA 512 y RSA */

                $time_inicio_a_simetrico = microtime(TRUE);
                $en  = RsaUtils::getInstance();
                $data = $_GET["temp"];
                $encrypt_asimetrico = $en->publicEncrypt( $data );
                $_POST["temp_asimetrico"]= $encrypt_asimetrico; /* compra minado
                completado */

                $time_end_a_simetrico = microtime(TRUE);
                $_POST["tiempo_asimetrico"]=$time_sin = round($time_end_a_simetrico - $
                    time_inicio_a_simetrico, 8);
```

Figura 40. Código fuente algoritmo SHA 512 y RSA – Cifrado asimétrico. Fuente: Elaboración Propia

Después, dentro de la carpeta Class, se encuentra el archivo PHP con el nombre de ASIMÉTRICO, en donde se utiliza la clave privada Rsa y la ubicación de almacenamiento de la clave pública-privada.

```php
class RsaUtils
{

        //Llave pública
    private $private;
        //Llave privada
    private $public;
        // Ubicación de almacenamiento de la clave pública-privada
    private $keyPath;

    private $openssl;

    private static $_instance;
    private function __construct()
    {
        $this->keyPath =  'C:\xampp\htdocs\ardu\class'.DIRECTORY_SEPARATOR;
                /*Ruta del conf. openssl */
        $this->openssl = 'C:\xampp\apache\conf\openssl.cnf';
```

Figura 41.Código fuente de ubicación de almacenamiento de la clave pública y privada. Fuente: Elaboración Propia

Se inicializa clave pública-privada y se comprueba si existe un fichero o directorio

```php
private function _init()
{
    if( !file_exists( $this->keyPath.'private_key.pem'  ) || !file_exists( $this->
        keyPath.'public_key.pem' ) )
    {
        $this->initialization();
    }

    return true;
}
```

Figura 42. Código fuente de inicialización de fichero o directorio. Fuente: Elaboración Propia

En esta función con el Openssl_pkey_new, se genera un nuevo par clave privada y clave pública, el componente público de la clave se puede obtener usando openssl_pkey_get_public().

```php
protected function initialization()
{
    $config = array(
        'digest_alg'        => 'sha512',
        'private_key_bits'  => 2048,
        'private_key_type'  => OPENSSL_KEYTYPE_RSA,
        'config'            => $this->openssl,
    );
    try{
                    // Crea un par de claves
        $res = openssl_pkey_new( $config );
                    // Generar clave privada
        openssl_pkey_export($res, $privkey, null, $config);
                    // Generar clave pública
        $pubKey = openssl_pkey_get_details($res)[ 'key' ];
        file_put_contents($this->keyPath.'private_key.pem', $privkey );
        file_put_contents($this->keyPath.'public_key.pem', $pubKey );

        $this->private = openssl_pkey_get_private( $privkey );
        $this->public = openssl_pkey_get_public( $privkey );
    }catch ( \Exception $e ){

    }

    return true;
}
```

Figura 43Código fuente de inicialización de clave pública y privada. Fuente: Elaboración Propia.

En esta función se analiza la clave Key y se prepara para usarla con otras funciones, pero también se extrae la clave privada.

```
private function _getPrivateKey()
{
    $strPrivateKey = $this->keyPath.'private_key.pem';
    $key = file_get_contents( $strPrivateKey );

    return openssl_pkey_get_private( $key );
}
```

Figura 44. Código fuente de análisis de clave key. Fuente: Elaboración Propia.

Se extrae la clave pública y se prepara para usarla con otras funciones.

```
private function _getPublicKey()
{
    $strPublicKey = $this->keyPath.'public_key.pem';
    $key = file_get_contents( $strPublicKey );

    return openssl_pkey_get_public( $key );
}
```

Figura 45. Código fuente de extracción de clave key. Fuente: Elaboración Propia.

En esta parte de programación, se obtiene clave pública

```
public function getPublicKey()
{
    $this->public =  $this->_getPublicKey();

    return $this->public;
}
```

Figura 46 .Código fuente para obtener clave pública. Fuente: Elaboración Propia.

En la función public, se obtiene clave privada

```
public function getPrivateKey()
{
    $this->private =  $this->_getPrivateKey();

    return $this->private;
}
```

Figura 47.Código fuente para obtener clave privada. Fuente: Elaboración Propia.

En esta función, se observa el cifrado de clave privada, en donde json_enconde retorna la representación JSON del valor dado.

```
public function privEncrypt( $data = null )
{
    if( is_null( $data ) )
    {
        return null;
    }
    if ( ( is_array( $data ) || is_object( $data ) ) && !empty( $data ) )
    {
        $data = json_encode( $data );
    }

    return openssl_private_encrypt( $data, $encrypted, $this->_getPrivateKey() ) ?
        base64_encode( $encrypted ) : null;
}
```

Figura 48. En esta función, se tiene el cifrado de clave pública y se utiliza el json_encode para poder retornar la representación Json del valor dado.

```
public function publicEncrypt( $data = null )
{
    if( is_null( $data ) )
    {
        return null;
    }
    if ( ( is_array( $data ) || is_object( $data ) ) && !empty( $data ) )
    {
        $data = json_encode( $data );
    }

    return openssl_public_encrypt( $data, $encrypted, $this->_getPublicKey() ) ?
        base64_encode( $encrypted ) : null;
}
```

Figura 49. Código fuente cifrado de la clave pública.

Fuente: Elaboración Propia.

En esta función se observa el descifrado de clave privada, en donde se desencripta la información data la cual fue previamente encriptada mediante openssl_public_decrypt y almacena el resultado en decrypted.

```
public function privDecrypt( $encrypted = null )
{
    if( is_null( $encrypted ) || !is_string( $encrypted ) || strlen( $encrypted )
        <= 0 )
    {
        return null;
    }

    return ( openssl_private_decrypt( base64_decode( $encrypted ), $decrypted, $
        this->_getPrivateKey() ) ) ? $decrypted : null;
}
```

Figura 50.Código fuente descifrado de la clave pública. Fuente: Elaboración Propia.

En esta función se observa, el descifrado de clave pública, openssl_public_decrypt desencripta la información data que fue previamente encriptada mediante openssl_private_encrypt y se puede usar esta función para verificar si el mensaje fue escrito por el propietario de la clave privada.

```
public function publicDecrypt( $encrypted = null )
{
    if( is_null( $encrypted ) || !is_string( $encrypted ) || strlen( $encrypted )
        <= 0 )
    {
        return null;
    }

    return ( openssl_public_decrypt( base64_decode( $encrypted ), $decrypted, $this
        ->_getPublicKey() ) ) ? $decrypted : null;
}
```

Figura 51. Código fuente descifrado de la clave privada. Fuente: Elaboración Propia.

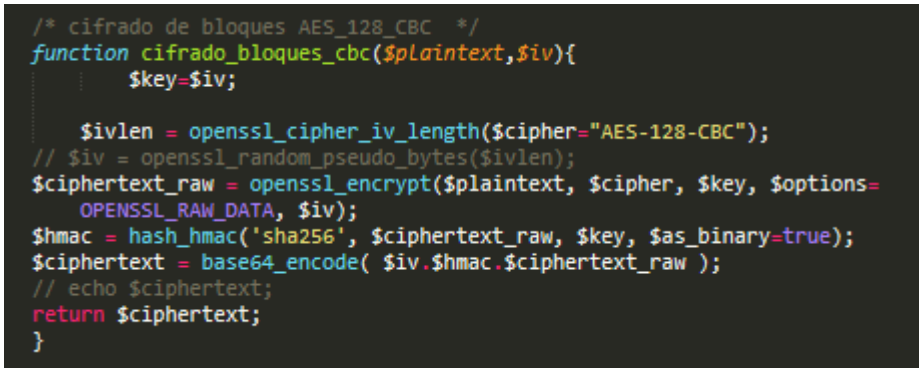
En guardar_data_arduino.php, se desarrolla el cifrado por bloques con el algoritmo AES 128 CBC, en donde en la primera línea se devuelve la fecha unix actual con microsegundos.

```
/* CIFRADO BLOQUES AES_128_CBC  */

                $time_inicio_bloques = microtime(TRUE);
                $iv = 'nayhelitesis1234';  /* key_iv */
                $_POST["temp_bloques"] = cifrado_bloques_cbc($_GET["temp"], $iv);

                $time_end_bloques = microtime(TRUE);
                $_POST["tiempo_bloques"]=$time_sin = round($time_end_bloques - $
                    time_inicio_bloques, 8);
```

Figura 52Código fuente de cifrado de bloques.

Fuente: Elaboración Propia.

Después, dentro de la carpeta Class, se encuentra el archivo PHP con el nombre de Functios de Aes 128 cifrado de bloques cbc, en donde se inicializa con el vector iv, y en donde opsenssl_encryp, encripta los datos con un método y una clave, devuelve una cadena codificada en base64.



```php
/* cifrado de bloques AES_128_CBC  */
function cifrado_bloques_cbc($plaintext,$iv){
        $key=$iv;

    $ivlen = openssl_cipher_iv_length($cipher="AES-128-CBC");
// $iv = openssl_random_pseudo_bytes($ivlen);
$ciphertext_raw = openssl_encrypt($plaintext, $cipher, $key, $options=
    OPENSSL_RAW_DATA, $iv);
$hmac = hash_hmac('sha256', $ciphertext_raw, $key, $as_binary=true);
$ciphertext = base64_encode( $iv.$hmac.$ciphertext_raw );
// echo $ciphertext;
return $ciphertext;
}
```

Figura 53. Código fuente cifrado de la técnica de bloques. Fuente: Elaboración Propia.

En esta parte la función, se hizo la decodificación de los datos codificados en base64, y el vector de inicialización (IV) que tiene que ser el mismo al cifrar y descifrar y en la siguiente línea openssl_cipher_iv_length se utiliza para obtener la longitud del iv de Cipher.

```php
function des_cifrado_bloques_cbc($ciphertext,$iv){
    $c = base64_decode($ciphertext);
        $key=$iv;

    // $iv = substr($c, 0, $ivlen);
    // $iv = '1234567812345678';
    $ivlen = openssl_cipher_iv_length($cipher="AES-128-CBC");
    $hmac = substr($c, $ivlen, $sha2len=32);
    $ciphertext_raw = substr($c, $ivlen+$sha2len);
    $original_plaintext = openssl_decrypt($ciphertext_raw, $cipher, $key, $options
        =OPENSSL_RAW_DATA, $iv);
    $calcmac = hash_hmac('sha256', $ciphertext_raw, $key, $as_binary=true);
    if (hash_equals($hmac, $calcmac))// timing attack safe comparison
    {
            // echo $original_plaintext."\n";
    }

    return $original_plaintext;
}
```

Figura 54. Código fuente descifrado de la técnica de bloques.

Fuente: Elaboración Propia.

En guardar_data_arduino.php, se desarrolla el cifrado híbrido con el algoritmo SHA 256, se observa que la primeria línea muestra la fecha unix actual con microsegundos del tiempo de inicio híbrido.

Después encrypt_hibrido se utiliza para cifrar el contenido indicado del búfer con el parámetro clave.

```
/* CIFRADO HIBRIDO: SHA 256  */
                $time_inicio_hibrido = microtime(TRUE);
                $en_hibri  = Cls_cifrado_hibrido::getInstance();
                $data = $_GET["temp"];
                $encrypt_hibrido = $en_hibri->publicEncrypt( $data );
                $_POST["temp_hibrido"]= $encrypt_hibrido; /* compra minado
                completado */

                $time_end_hibrido = microtime(TRUE);
                $_POST["tiempo_hibrido"]=$time_sin = round($time_end_hibrido - $
                    time_inicio_hibrido, 8);
```

Figura 55. Código fuente de cifrado hibrido- SHA 256.

Fuente: Elaboración Propia.

Después, dentro de la carpeta Class, se encuentra el archivo PHP con el nombre de híbrido, en donde se utiliza la clave pública y la ubicación de almacenamiento de la clave pública-privada.

```php
class Cls_cifrado_hibrido
{

        //Llave pública
    private $private;
        //Llave privada
    private $public;
        // Ubicación de almacenamiento de la clave pública-privada
    private $keyPath;

    private $openssl;

    private static $_instance;
    private function __construct()
    {
        $this->keyPath =  'C:\xampp\htdocs\ardu\class'.DIRECTORY_SEPARATOR;
                /*Ruta del conf. openssl */
        $this->openssl = 'C:\xampp\apache\conf\openssl.cnf';

        $this->_init();
    }

    public static function getInstance()
    {
        if( null == self::$_instance )
        {
            self::$_instance = new self();
        }
        return self::$_instance;
    }
```

Figura 56. Código fuente cifrado hibrido.

Fuente: Elaboración Propia.

En esta función, se inicializa la clave pública-privada y se comprueba si existe un fichero o directorio.

```php
private function _init()
{
    if( !file_exists( $this->keyPath.'private_key_hibrido.pem'  ) || !
        file_exists( $this->keyPath.'public_key_hibrido.pem' ) )
    {
        $this->initialization();
    }

    return true;
}
```

Figura 57. Código fuente inicialización de clave pública-privada. Fuente: Elaboración Propia.

En la función initialization, con el openssl_pkey_new se genera una clave privada nueva par clave privada y clave pública y con el openssl_pkey_export se obtiene una representación de una clave exportable a una cadena PEM codificada y la almacena.

```php
protected function initialization()
{
    $config = array(
        'digest_alg'        => 'sha256',
        'private_key_bits'  => 2048,
        'private_key_type'  => OPENSSL_KEYTYPE_RSA,
        'config'            => $this->openssl,
    );
    try{
                    // Crea un par de claves
        $res = openssl_pkey_new( $config );
                    // Generar clave privada
        openssl_pkey_export($res, $privkey, null, $config);
                    // Generar clave pública
        $pubKey = openssl_pkey_get_details($res)[ 'key' ];
        file_put_contents($this->keyPath.'private_key_hibrido.pem', $privkey )
            ;
        file_put_contents($this->keyPath.'public_key_hibrido.pem', $pubKey );

        $this->private = openssl_pkey_get_private( $privkey );
        $this->public = openssl_pkey_get_public( $privkey );
    }catch ( \Exception $e ){

    }

    return true;
}
```

Figura 58. Código fuente de función de inicialización.

Fuente: Elaboración Propia.

En esta función, se extrae clave privada, y el file_get_contents transmite un fichero completo a una cadena.

```php
private function _getPrivateKey()
{
    $strPrivateKey = $this->keyPath.'private_key_hibrido.pem';
    $key = file_get_contents( $strPrivateKey );

    return openssl_pkey_get_private( $key );
}
```

Figura 59. Código fuente función para extraer clave privada. Fuente: Elaboración Propia.

En esta función, se extrae clave publica y es preparada para usarla.

```php
private function _getPublicKey()
{
    $strPublicKey = $this->keyPath.'public_key_hibrido.pem';
    $key = file_get_contents( $strPublicKey );

    return openssl_pkey_get_public( $key );
}
```

Figura 60 .Código fuente función para extraer clave pública. Fuente: Elaboración Propia.

```php
public function getPublicKey()
{
    $this->public =  $this->_getPublicKey();

    return $this->public;
}
```

Figura 61.Código fuente función para obtener clave pública.

Fuente: Elaboración Propia.

```php
public function getPrivateKey()
{
    $this->private =  $this->_getPrivateKey();

    return $this->private;
}
```

Figura 62.Código fuente función para obtener clave privada.

Fuente: Elaboración Propia.

En esta función, se observa el cifrado de clave privada, en donde Json_enconde retorna la representación del valor dado y con el openssl_private_encryt se encripta la información con la clave privada.

```php
public function privEncrypt( $data = null )
{
    if( is_null( $data ) )
    {
        return null;
    }
    if ( ( is_array( $data ) || is_object( $data ) ) && !empty( $data ) )
    {
        $data = json_encode( $data );
    }

    return openssl_private_encrypt( $data, $encrypted, $this->_getPrivateKey()
        ) ? base64_encode( $encrypted ) : null;
}
```

Figura 63 Código fuente función para encriptar clave privada.

Fuente: Elaboración Propia.

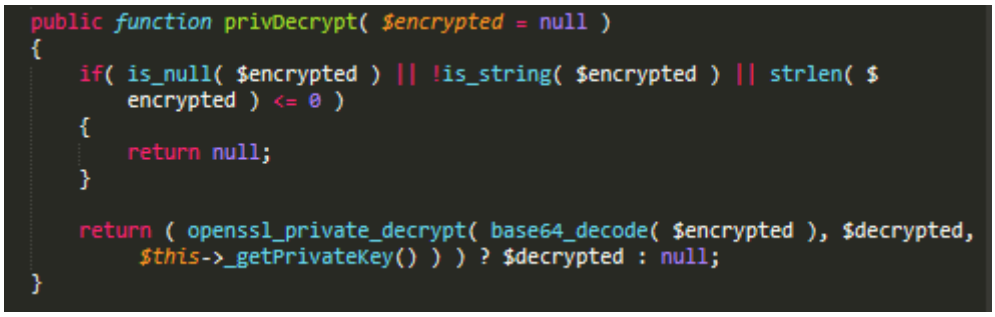
En esta función, se observa el cifrado de clave pública, y se comprueba si la variable es null, y finalmente retorna con el openssl_public_encrypt y es almacenado como una cadena codificada en base64.

```php
public function publicEncrypt( $data = null )
{
    if( is_null( $data ) )
    {
        return null;
    }
    if ( ( is_array( $data ) || is_object( $data ) ) && !empty( $data ) )
    {
        $data = json_encode( $data );
    }

    return openssl_public_encrypt( $data, $encrypted, $this->_getPublicKey() )
        ? base64_encode( $encrypted ) : null;
}
```

Figura 64.Código fuente función para encriptar clave pública.

Fuente: Elaboración Propia.

En esta función, se realiza el descifrado de clave privada y es almacenado como una cadena codificada en base64.



```
public function privDecrypt( $encrypted = null )
{
    if( is_null( $encrypted ) || !is_string( $encrypted ) || strlen( $
        encrypted ) <= 0 )
    {
        return null;
    }

    return ( openssl_private_decrypt( base64_decode( $encrypted ), $decrypted,
         $this->_getPrivateKey() ) ) ? $decrypted : null;
}
```

Figura 65. Código fuente función para desencriptar clave privada. Fuente: Elaboración Propia.

En esta función se realiza el descifrado de clave pública y retorna con el openssl_public_decrypt, para desencriptar la información data que fue previamente encriptada

```
public function publicDecrypt( $encrypted = null )
{
    if( is_null( $encrypted ) || !is_string( $encrypted ) || strlen( $
        encrypted ) <= 0 )
    {
        return null;
    }

    return ( openssl_public_decrypt( base64_decode( $encrypted ), $decrypted,
        $this->_getPublicKey() ) ) ? $decrypted : null;
}
```

Figura 66.Código fuente función para desencriptar clave pública.

Fuente: Elaboración Propia

En la base de datos de MYSQL, se inserta las cuatro técnicas de cifrado (simétrica, asimétrica, híbrida y finalmente de bloques) programadas en la base de datos.

```
/* Inserta en Base Datos */
                $campos=array('temp','temp_asimetrico','temp_bloques','temp_hibrido
                    ','tiempo_simetrico','tiempo_asimetrico','tiempo_bloques','
                   tiempo_hibrido','estado_idestado','orden','fecha_registro');
                $bd->inserta_(arma_insert('data_arduinos',$campos,'POST'));


                $rpta=1;
```

Figura 67.Insertar las cuatro técnicas a la base de datos. Fuente: Elaboración Propia.

Finalmente, en esta fase se realizó la creación de la página web, en el que se tiene que entrar con la siguiente URL: http://192.168.1.38/ardu/

En la siguiente imagen, se muestra el login de nuestro sistema web y se ingresa como administrador.

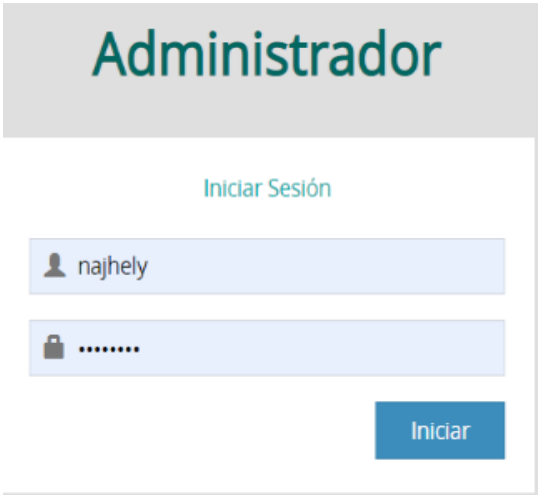


Figura 68.Login de inicio de sesión. Fuente: Elaboración Propia.

En esta imagen, se muestra la sección de usuarios de los dos administradores creados, para poder hacer inicio a la sesión de cuenta

.

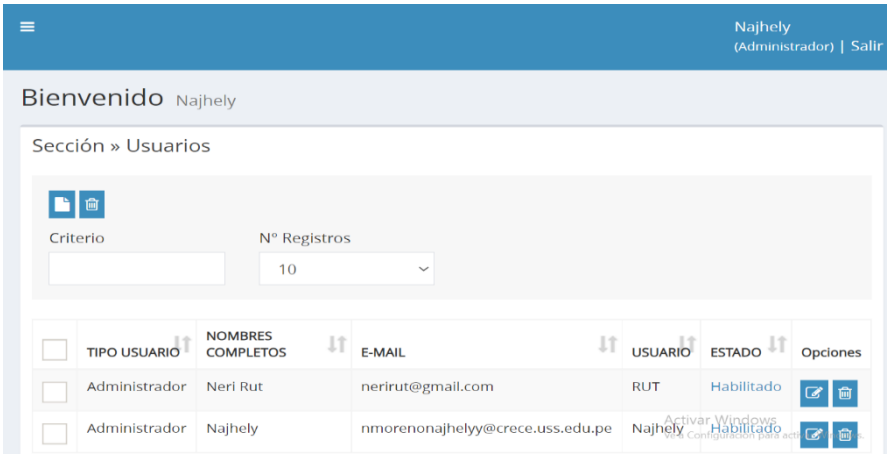


Figura 69. Home de inicio de sesión.

Fuente: Elaboración Propia.

En esta fase se observa la data, que se envía cuando se toma el sensor de temperatura y se muestra el cifrado en la página web. Finalmente, se registra cada paciente, y se muestra la temperatura desencriptada con las cuatro técnicas.

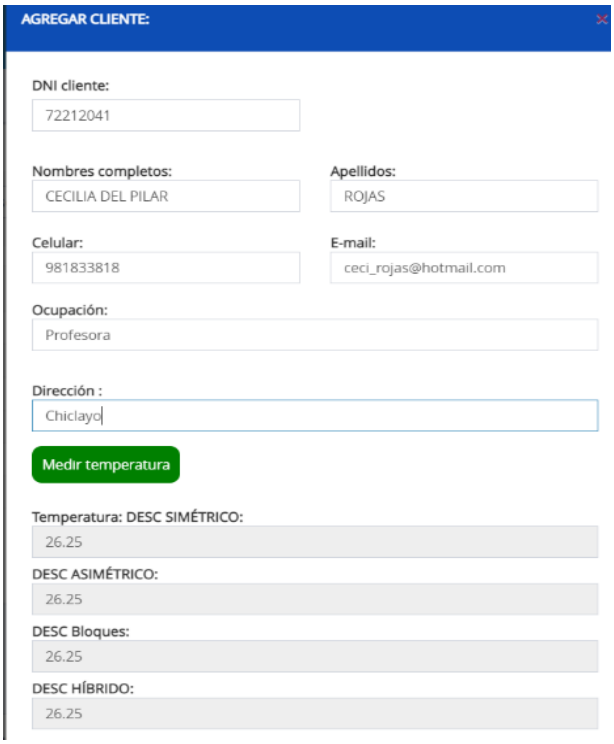


Figura 70. Toma de temperatura de paciente.

Fuente: Elaboración Propia

Evaluar los resultados obtenidos

Para la evaluación de resultados se utilizaron los siguientes indicadores:

Tiempo de Procesamiento

Usamos la siguiente fórmula:

$$x = \frac{X_1+X_2+X_3+\cdots+x_n}{N}$$

Media Aritmética: La media aritmética es el valor logrado al sumar todos los datos y dividir el resultado entre el número total de datos. Su sintaxis es la siguiente:

$$x = \frac{X_1 + X_2 + X_3 + \cdots + x_n}{N}$$

Donde: X1= T1; T1=TF-TI

X1: Prueba 1;

T1: Tiempo 1 para la prueba 1.

TF: Intervalo final.

TI: Intervalo inicial.

Con la conexión establecida del Arduino a la base de datos MySQL se procedió a realizar el tiempo de procesamiento.

Tabla 24

Tiempo de Procesamiento de la técnica simétrica

1° Técnica de Cifrado Simétrica	**Tiempo de Procesamiento (seg)**				
	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5

AES GCM	**256**	0.00096607 Seg	0.00003004 seg	0.00259089 seg	0.00005293 seg	0.00006390 Seg

Fuente: Elaboración propia

$$x = \frac{0.00096607 + 0.00003004 + 0.00259089 + 0.00005293 + 0.00006390}{5}$$

$$x = 0{,}00074077$$

Tabla 25

Tiempo de Procesamiento de la técnica asimétrica

2° Técnica de Cifrado Asimétrica	**Tiempo de Procesamiento (seg)**				
	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5
SHA 512 Y RSA	0.00046515 seg	0.00466800 seg	0.00121093 seg	0.00075698 seg	0.00089502 seg

Fuente: Elaboración propia

$$x = \frac{0.00046515 + 0.00466800 + 0.00121093 + 0.00005293 + 0.00089502}{5}$$

$$x = 0.00159922$$

Tabla 26

Tiempo de Procesamiento de la técnica por bloques.

3° Técnica de Cifrado por bloques	**Tiempo de Procesamiento (seg)**				
	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5
AES 128 CBC	0.00003886	0.00001407	0.00110102	0.00002289	0.00002503

Fuente: Elaboración propia

$$x = \frac{0.00003886 + 0.00001407 + 0.00110102 + 0.00002289 + 0.00002503}{5}$$

$$x = 0.00031984$$

Tabla 27

Tiempo de Procesamiento de la técnica por híbrido.

4° Técnica de Cifrado hibrida	Tiempo de Procesamiento (seg)				
	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5
SHA 256	0.00116706	0.00032187	0.00132108	0.00110912	0. 00058103

Fuente: Elaboración propia

$$\frac{x = 0.00116706 + 0.00032187 + 0.00132108 + 0.00110912 + 0.00058103}{5}$$

$$x = 0.000900032$$

Margen de Error en la transmisión

Para este indicador se utilizará la desviación estándar.

Usamos la siguiente fórmula:

Desviación Estándar: Esta fórmula se operará para establecer el margen de error en la seguridad de la transmisión de los datos.

$$\sigma = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \cdots + (x_n - \bar{x})^2}{N}}$$

Para el margen de error de las 4 técnicas, se realizó el procedimiento de la fórmula de desviación estándar.

Se utilizó las primeras 5 pruebas, sus tiempos en segundos y se restó por el total del tiempo del procesamiento total.

Tabla 28

Margen de Error en la transmisión de Técnica simétrica

Técnicas	Margen de Error
Margen de error Técnica Simétrica	0.00140142%
Margen de error técnica asimétrica	0.05018%
Margen de error técnica por bloques	0.00043767 %
Margen de error técnica híbrida	0.00038166%

Fuente: Elaboración Propia.

Tabla 29

Resultado de rendimiento por cada técnica de cifrado

Técnicas de cifrado	*Rendimiento*
Técnica de cifrado simétrica	*0.3044%*
Técnica de cifrado asimétrica	*0.293%*
Técnica de cifrado por bloques	*0,407%*
Técnica de cifrado híbrido	*0.276%*

Fuente: Elaboración Propia.

IV. CONCLUSIONES Y RECOMENDACIONES

4.1. **Conclusiones.**

La selección de técnicas de cifrado de datos para el intercambio de datos en IoT, se realizó mediante los artículos científicos, realizadas con una búsqueda eficiente de cinco palabras claves. Además, se observó que las técnicas de cifrado elegidas como cifrado simétrico, asimétrico, por bloques e híbrido, representan el 100% de técnicas de cifrado seleccionadas en base a la revisión bibliográfica en las bases de datos de IEEE Xplore y ProQuest.

En la presente investigación, se ha tenido que identificar los criterios de evaluación como la longitud de claves, tipo de cifras, tamaño de bloques, posibles claves, ronda, estructura del algoritmo, de las técnicas de cifrado de datos, con el fin que se cumplan todos los requisitos de mucha importancia como longitud de clave, posibles claves, su autor y rendimiento de encriptación.

En la presente investigación se realizó un diseño de escenario en el campo de la Salud, haciendo uso de un sensor de temperatura DS18B20, para que los datos transmitidos sean cifrados en la base de datos MySQL y así poder mostrarlos en la página web, cuando cada paciente sea registrado, con lo que podemos concluir que, al ser cifrados estos datos, se puede comprobar cuál de las técnicas es el mejor rendimiento tiene como la híbrida que obtuvo 0.276%.

Al finalizar la investigación se concluye que, al aplicar las técnicas de cifrado en el escenario de pruebas, se realizó las tomas de temperatura, y se logró cifrar en la base de datos, gracias a ello se observó que la técnica de cifrada híbrida es la que obtuvo menos margen de error dando un 0.000318166%.

Al finalizar la investigación se concluye que, en el campo de la salud, la técnica más eficiente para el intercambio de datos es el cifrado Híbrido con el algoritmo SHA 256, ya que, al intercambiar los datos, es el cifrado más eficiente en cuanto

a tiempo de procesamiento total con un resultado de 0.00450016 segundos y rendimiento 0.276%.

4.2. Recomendaciones.

Se recomienda para la búsqueda de referencia bibliográfica la combinación de 4 palabras claves para la formación de las cadenas de búsqueda: Encyption AND techniques AND for AND IoT.

Se recomienda realizar las pruebas de técnicas de cifrado, evaluando con el sensor de temperatura DS18B20 bajo los mismos indicadores como, tiempo de procesamiento, margen de error en el intercambio de datos, longitud de claves, número de bits y finalmente rendimiento.

Se recomienda que debido a la pandemia COVID 19, se tome en cuenta el uso de estas técnicas de cifrado, para que, los datos transmitidos de los pacientes no sean expuestos y se encuentren bien protegidos.

REFERENCIAS.

Access Quality. (12 de Enero de 2021). *El Cifrado de Datos (Tipos y Soluciones)*. Obtenido de Access Quality: https://www.accessq.com.mx/cifrado-de-datos/

Afsoon , Y., & Seyed , M. J. (2017). Improving the Security of Internet of Things using. *IEEE*, 1-5.

Bhandari, R., & Kirubanand, V. B. (2019). Enhanced encryption technique for secure iot data transmission. *International Journal of Electrical and Computer Engineering.* Malaysia, Yogyakarta: Computer Engineering.

Bodur, H., & Resul, K. (June de 2015). *Secure SMS Encryption Using RSA Encryption Algorithm on Android Message Application.* Obtenido de Researchgate: https://www.researchgate.net/publication/298298027_Secure_SMS_Encryption_Using_RSA_Encryption_Algorithm_on_Android_Message_Application

BricoGeek. (12 de Noviembre de 2021). *Sensor DS18B20 estanco*. Obtenido de Sensor DS18B20 estanco: https://tienda.bricogeek.com/sensores-temperatura/510-sensor-ds18b20-estanco.html

Chaudhary, R. R., & Chatterjee, K. (2020). An Efficient Lightweight Cryptographic Technique For IoT based E-healthcare System. *IEEE*, 1-5.

Cisco. (2016). *Tráfico en Internet de smartphones superará al de PCs para 2020*. Obtenido de Tráfico en Internet de smartphones superará al de PCs para 2020: https://www.tynmagazine.com/trafico-en-internet-de-smartphones-superara-al-de-pcs-para-2020/

Corak, B., Guzel, M., Murt, S., & Ozdemir, S. (2018). Análisis comparativo de las comunicaciones de IoTC protocolos. *InternetWord*.

Cryptobook. (2021). *Criptografía de curva elíptica (ECC).* Obtenido de Cryptobook: https://cryptobook.nakov.com/asymmetric-key-ciphers/elliptic-curve-cryptography-ecc

El-Latif, A., Abd-El-Atty, B., Mazurczyk, W., & Member, S. (2020). Secure Data Encryption Based on Quantum Walks. *IEEE*, 1-14.

Er, S. K., & attri, E. v. (2017). Revisión sobre técnicas de identificación por radiofrecuencia y su uso para asegurar IOT. India, Udaipur.

Essex, A. (15 de May de 2020). *Block Cipher.* Obtenido de Security Encyclopedia: https://www.hypr.com/black-cipher/

García, J. (16 de Noviembre de 2021). *Qué es la resistencia eléctrica*. Obtenido de qué es la resistencia eléctrica: http://www.asifunciona.com/electrotecnia/ke_resistencia/ke_resistencia_1.htm

GeeksforGeeks. (20 de Octubre de 2021). *Algoritmo de cifrado ElGamal.* Obtenido de GeeksforGeeks: https://www.geeksforgeeks.org/elgamal-encryption-algorithm/

Geng, H. (12 de mayo de 2017). *¿El IoT debe importarle a las Pymes?* Obtenido de ¿El IoT debe importarle a las Pymes?: https://destinonegocio.com/pe/emprendimiento-pe/el-iot-debe-importarle-a-las-pymes/

IEEE. (27 de Mayo de 2015). *Hacia una definición de Internet de las cosas (IoT).* Obtenido de IEEE: https://iot.ieee.org/definition.html

InternetSociety. (3 de Octubre de 2015). *La internet de las cosas- una breve reseña*. Obtenido de la internet de las cosas - una breve reseña: https://www.internetsociety.org/wp-content/uploads/2017/09/report-InternetOfThings-20160817-es-1.pdf

Jimenez, H., Rodriguez, R., & Tiparra, J. (1978). *Diagnóstico de TEA.* Madrid: Latinoamérica SA.

Khari, M. G. (2019). Securing Data in Internet of Things (IoT) Using. *IEEExplore.* New York: Signal Processing and the Internet of Things.

Kumar, R., Khan, P., & Kumar, S. (2019). A Cellular Automata-based Healthcare Data Encryption Technique for IoT Networks. *Ieexplore*, 1-4.

Lin, C., Cheng, Y., & Chuan, K. K. (19 de Octubre de 2017). Lower Power Data Transport Protection for Internet. *IEEExplore*, 1-3. doi:10.1109/desec.2017.8073865

López, A. (04 de Abril de 2021). *Todo sobre Criptografía: Algoritmos de clave semétrica y asimétrica*. Obtenido de Redes Zone: https://www.redeszone.net/tutoriales/seguridad/criptografia-algoritmos-clave-simetrica-asimetrica/

Manju, K., Kumar, G., & Amir, H. G. (27 de Marzo de 2019). Securing Data in Internet of Things (IoT) Using. *Ieee transactions on systems, man, and cybernetics: systems, 50*(73 - 80), 73 - 80. doi:10.1109/tsmc.2019.2903785

Medina, Y., & Miranda, H. (21 de Junio de 2015). *Comparación de Algoritmos basados en la Criptografía simétrica DES, AES, 3DES.* Obtenido de Science: https://dialnet.unirioja.es/descarga/articulo/5286657.pdf

Mejia, I., Ramirez, R., Jimenez, H., & Rosas, J. (2019). A new method a architecture entreprise. *Conference IEEE bussines*, 200-215.

Mejia, I., Tuesta, M., & Forero, M. (2020). A new method of enterprise archicture small organizations. *Computer Science Techology*, 150-170.

Micrsoft Azure. (2020). *Protocolos y tecnologías de IoT.* Obtenido de Micrsoft Azure: https://azure.microsoft.com/es-es/overview/internet-of-things-iot/iot-technology-protocols/

OSI. (10 de Julio de 2019). *Oficinaa de Seguridad de Internauta*. Obtenido de OSI: https://www.osi.es/es/actualidad/blog/2019/07/10/sabias-que-existen-distintos-tipos-de-cifrado-para-proteger-la-privacidad

Oualha, N. (06 de Septiembre de 2018). Reinforcing IoT-Enforced Security Policies. *Conference on Trust, Security and Privacy in Computing*, 1-6. doi:10.1109

Palma, J., & Marín, R. (2008). *Inteligencia Artificial.* Madrid: McGrawHill. doi:978-84-481-5618-3

PlastimediaStudio. (13 de Noviembre de 2021). *Tarjeta NodeMcu V3 para ESP8266*. Obtenido de Tarjeta NodeMcu V3 para ESP8266: https://www.didacticaselectronicas.com/index.php/comunicaciones/wi-fi/tarjeta-nodemcu-v3-para-esp8266-esp8266-wifi-nodemcu-nodem-wemos-d1-internet-iot-sistemas-tarjetas-de-desarrollo-con-de-wifi-wi-fi-para-esp8266-esp-8266-nodemcu-detail

Ponkanen, S. (09 de April de 2020). *Aes-gcm - what could go wrong?* Obtenido de ssh.random: https://blog.ssh.com/tech/aes-gcm-what-could-go-wrong

Quental. (2016). *Tecnología IoT en el Sector Hospitalario*. Obtenido de Tecnología IoT en el Sector Hospitalario: https://www.quental.com/media/files/Informe-Tecnologia-IoT-en-el-Sector-Hospitalario.pdf

Rahman, H., Ahmed, N., & Hussain, I. (2016). Comparación de técnicas de agregación de datos en Internet de las cosas (IoT). *Iexplore*, 1-5.

Ramírez, H. (12 de Mayo de 2020). *Qué es la criptografía asimétrica y cómo funciona*. Obtenido de Grupo Atico 34: https://protecciondatos-lopd.com/empresas/criptografia-asimetrica/

Ravi Raushan, K., & Kakali , C. (2020). An Efficient Lightweight Cryptographic Technique For IoT based E-healthcare System. *International Conference on Signal Processing and Integrated Networks*, 991-995. Obtenido de https://ieeexplore.ieee.org/document/9071421

RedHat. (12 de mayo de 2019). *INTERNET OF THINGS (IOT)*. Obtenido de internet of things (iot): https://www.redhat.com/es/topics/internet-of-things/what-is-iot

Rojas, K. (2018). Identificación de efectos negativos de la TEA en el aprendizaje. *IEEE conference Techology children especial*, 200-215.

Rosales, G. (30 de 10 de 2020). *CrYptografia en Zoom.* Obtenido de Yanapti: https://yanapti.com/2020/cryptografia-en-zoom/

Salazar, J., & Silvestre, S. (08 de Noviembre de 2018). *Internet de las cosas.* Obtenido de Techpedia: https://core.ac.uk/download/pdf/81581111.pdf

Sarin, A., Thanawala, D., Verma, S., & Prakash, C. (15 de Octubre de 2020). Implementation of New Approach to Secure IoT Networks with Encryption and Decryption Techniques. *Communication and Networking Technologies (ICCCNT)*, 1-7. doi:10.1109/ICCCNT49239.2020.9225279

Sarmilla, K. B., & Manisekaran, S. V. (31 de Octubre de 2019). A Study on Security Considerations in IoT Environment and Data Protection Methodologies for Communication in Cloud Computing. *IEEExplore*, 1-6. doi:10.1109/c

Shafiq, M., Tian, Z., Bashir, A., & Du, X. (2020). CorrAUC: a Malicious Bot-IoT Traffic Detection Method in IoT Network Using Machine Learning Techniques. *IEEE xplore*, 3242 - 3254.

Sharma, C., Shri, U., Vaishno, M., Shri, U., & Vaishno, M. (12 de mayo de 2018). *Pila de protocolo de comunicación para sistemas de IoT restringidos 0-5*. Obtenido de Pila de protocolo de comunicación para sistemas de IoT restringidos 0-5: https://doi.org/10.1109/IoT-SIU.2018.8519904

Shea, S., & Wigmore, I. (2020). *IoT security (internet of things security)*. Obtenido de IoT Agenda: https://internetofthingsagenda.techtarget.com/definition/IoT-security-Internet-of-Things-security#:~:text=IoT%20security%20is%20the%20technology,%2C%20animals%20and%2For%20people.

Siddhart, S., & Devgon, R. (2019). Analysis of Encryption and Lossless Compression Techniques for Secure Data Transmission. *Ieexplore*, 1-5.

Singh, K., Johari, R., Singh, K., & Tyagi, H. (2019). Mercurial Cipher: A new cipher technique and comparative analysis with classical cipher techniques. *International Conference on Computing, Communication, and Intelligent Systems*, 1-6. doi:10.1109 / ICCCIS48478.2019.8974473

Susana, G. (31 de Enero de 2020). *El número de usuarios de Internet en el mundo crece un 7% y alcanza los 4.540 millones (2020)*. Obtenido de Marketing 4ecommerce: https://marketing4ecommerce.net/usuarios-internet-mundo/

SZNAJDLEDER, P. (2012). *Java a fondo - estudio del lenguaje y desarrollo de aplicaciones - 2a ed.* México: Alfaomega.

TEC. (5 de Mayo de 2020). *Caminar con éxito hacia la Industria 4.0: Capítulo 14 – Dispositivos (I) Internet de las cosas (IoT)*. Obtenido de Tecnología para los Negocios - Camara Valencia: https://ticnegocios.camaravalencia.com/servicios/tendencias/caminar-con-exito-hacia-la-industria-4-0-capitulo-14-dispositivos-i-internet-de-las-cosas-iot/

Vermesan, O. (Septiembre de 2011). *Internet of Things Strategic Research Roadmap.* Obtenido de El Clúster de Proyectos de Investigación Europeos, Tech. Reps: http://internet-of-things-research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2011.pdf

Xilinx. (2020). *Modo CBC.* Obtenido de Xilinx: https://xilinx.github.io/Vitis_Libraries/security/2020.1/guide_L1/internals/cbc.html

ANEXOS.

Anexo 1. Resolución de aprobación del proyecto de investigación

UNIVERSIDAD SEÑOR DE SIPÁN

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

RESOLUCIÓN N°0445-2021/FIAU-USS

Pimentel, 27 de mayo de 2021

VISTO:
El Acta de reunión N°1305-2021 del Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS remitida mediante oficio N°0227-2021/FIAU-IS-USS de fecha 19 de mayo de 2021, y;

CONSIDERANDO:
Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a letra dice: "La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";

Que, de conformidad con el Reglamento de grados y títulos en su artículo 21° señala: "Los temas de trabajo de investigación, trabajo académico y tesis son aprobados por el Comité de Investigación y derivados a la Facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El periodo de vigencia de los mismos será de dos años, a partir de su aprobación. En caso un tema perdiera vigencia, el Comité de Investigación evaluará la ampliación de la misma.

Que, de conformidad con el Reglamento de grados y títulos en su artículo 24° señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; es individual o en pares para obtener un título profesional. Asimismo, en su artículo 25° señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C.".

Que, según documentos de Vistos el Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS acuerdan aprobar los temas de las Tesis a cargo de los estudiantes del curso de Investigación I que se detallan en el anexo de la presente Resolución.

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

SE RESUELVE:

ARTÍCULO 1°: APROBAR, el tema de la Tesis perteneciente a la línea de investigación de INFRAESTRUCTURA, TECNOLOGÍA Y MEDIO AMBIENTE, a cargo de los estudiantes del Programa de estudios de INGENIERÍA DE SISTEMAS según se detalla en el anexo de la presente Resolución.

ARTÍCULO 2°: ESTABLECER, que la inscripción del Tema de la Tesis se realice a partir de emitida la presente resolución y tendrá una vigencia de dos (02) años.

ARTÍCULO 3°: DEJAR SIN EFECTO, toda Resolución emitida por la Facultad que se oponga a la presente Resolución.

REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE

Dr. Mario Fernando Ramos Moscol
Decano - Facultad de Ingeniería, Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.

MBA. María Noelia Sialer Rivera
Secretaria Académica / Facultad de Ingeniería, Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.

Cc: Interesado, Archivo

UNIVERSIDAD SEÑOR DE SIPÁN

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

RESOLUCIÓN N°0445-2021/FIAU-USS

Pimentel, 27 de mayo de 2021

ANEXO

N°	AUTOR (ES)	TEMA DE TESIS
1	RIMARACHIN ESCRIBANO NERI RUT NIÑO MORENO NAJHELY YAMILETT	EVALUACIÓN DE TÉCNICAS DE CIFRADO PARA EL INTERCAMBIO DE DATOS DE INTERNET DE LAS COSAS EN EL ÁMBITO DE LA SALUD
2	GUEVARA CHAMBERGO JHON DENNIS BOBADILLA CAMPOS ROLANDO MARTIN	DESARROLLO DE UNA METODOLOGÍA DE GESTIÓN DE RIESGOS AD HOC BASADA EN MARCOS INTERNACIONALES Y BUENAS PRÁCTICAS PARA UNA EMPRESA MANUFACTURERA PERUANA
3	CIEZA CELIS JESUS ABELARDO OJEDA ROMERO ANTHONNY JHONATAN	EVALUACIÓN DEL DESEMPEÑO DE LOS ESQUEMAS DE SEGURIDAD DE RED PARA COMBATIR VULNERABILIDADES EN REDES INALÁMBRICAS BASADAS EN EL PROTOCOLO WPA2
4	MENDOZA FERRÉ ESPERANZA NATALY CABRERA SANCHEZ KEVIN ALONSO	COMPARACIÓN DEL RENDIMIENTO DE TECNOLOGÍAS DE VIRTUALIZACIÓN PARA EL DESPLIEGUE DE APLICACIONES CON ARQUITECTURA DE MICROSERVICIOS
5	TEMOCHE GOMEZ LENNIN BILLEY	DESARROLLO DE UN MÉTODO PARA DETECTAR CON EFICIENCIA LAS VULNERABILIDADES INFORMÁTICAS DE ATAQUE CROSS-SITE SCRIPTING UTILIZANDO TÉCNICAS DE APRENDIZAJE AUTOMÁTICO
6	CASTRO MEDINA MIGUEL ANGEL	IMPLEMENTACIÓN DE UNA METODOLOGÍA AD HOC DE GESTIÓN DE LA SEGURIDAD DE LA INFORMACIÓN PARA UNA EMPRESA EDITORA DE DIARIO REGIONAL PERUANO
7	MURO ESPINOZA JUAN JOSE	DESARROLLO DE UNA METODOLOGÍA AD HOC DE GESTIÓN DE LA SEGURIDAD DE LA INFORMACIÓN PARA UN INSTITUTO SUPERIOR PEDAGÓGICO PERUANO
8	DIAZ ZAVALA ROXANA KARINA FRIAS VASQUEZ LADY	DESARROLLO DE UNA METODOLOGÍA AD HOC DE GESTIÓN DE LA SEGURIDAD DE LA INFORMACIÓN PARA UNA UNIDAD DE GESTIÓN EDUCATIVA PERUANA
9	CARRASCO BORDA APARICIO	DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110
10	OTERO MORALES JAVIER LIZARDO AQUINO SOSA NOELIA STEPHANY	DESARROLLO DE UN MODELO DE PROCESOS BASADO EN NORMAS DE PEQUEÑAS ORGANIZACIONES PARA MEJORAR LA CONSTRUCCIÓN DE SOFTWARE EN UN ÁREA DE DESARROLLO DE GOBIERNO MUNICIPAL
11	CALDERON YNOÑAN PAMELA DEL CARMEN PRIETO NEIRA FRANCK ALBERSON	DESARROLLO DE UN MÉTODO BAJO EL ENFOQUE ÁGIL EN ENTORNOS DE EXPERIENCIA DE USUARIO UI/UX PARA ASEGURAR LA USABILIDAD WEB
12	FLORES TINEO HUGO GALVANI DOLORIER POMA RONY RAUL	EVALUACIÓN DE LA USABILIDAD EN ENTORNOS VIRTUALES DE APRENDIZAJE PARA USUARIOS DE LAS ZONAS RURALES DEL PERÚ UTILIZANDO LA NORMA ISO/IEC 25010
13	CHANCAFE CASTRO JULIO JOEL	DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE PARA UNA MUNICIPALIDAD BASADO EN ISO/IEC 29110
14	SALAZAR DAVILA GIANFRANCO STEVEN	COMPARACIÓN DE TÉCNICAS DE VALIDACIÓN DE REQUISITOS DE SOFTWARE PARA MEDIR LA INFLUENCIA EN EL ÉXITO DE LOS PROYECTOS DE DESARROLLO EN PEQUEÑAS EMPRESAS PERUANAS
15	RIOJA MESIA CHARLES SEGUNDO FERNANDEZ RIOJA JUAN NICANOR	IMPLEMENTACIÓN DE UN MODELO DE GESTIÓN DE INCIDENCIAS BASADO EN ITIL PARA MEJORAR EL SERVICIO DE TI EN UNA MUNICIPALIDAD DISTRITAL DE LA REGIÓN LAMBAYEQUE
16	ALFARO PAJARES JUAN PEDRO	EVALUACIÓN DEL DESEMPEÑO DE PROCESOS DE NEGOCIO GESTIONADOS POR BPM EN UNA EMPRESA CONSTRUCTORA PERUANA
17	MONSALVE FERNANDEZ LENIN ESTALIN	IMPLEMENTACIÓN DE UN MODELO DE GESTIÓN DE SERVICIOS DE TI BASADO EN ITIL PARA MEJORAR LA GESTIÓN DE LOS SERVICIOS DE LA DIRECCIÓN DE TECNOLOGÍA DE UN GOBIERNO REGIONAL PERUANO
18	PEREZ CAMPOS DE QUIROZ BETTY MAGALY	EVALUACIÓN DEL DESEMPEÑO DE PROCESOS DE NEGOCIO GESTIONADOS POR BPM EN UNA MICRO EMPRESA PERUANA DESARROLLADORA DE SOFTWARE
19	MONTJOY PITA BRUNO	DESARROLLO DE UN SISTEMA DE RECOMENDACIÓN AUTOMÁTICA PARA EL TRATAMIENTO DE LAS PLAGAS EN CULTIVOS DE ARROZ DE LAS VARIEDADES QUE SE PRODUCEN EN LA REGIÓN LAMBAYEQUE
20	CRUZ FLORES JOSE ANTONIO CHAVEZ ANGULO GERMAN NEPTALI	IMPLEMENTACIÓN DE ARQUITECTURA EMPRESARIAL BASADO EN METODOLOGÍA ÁGIL PARA ALINEAR LAS TECNOLOGÍAS DE INFORMACIÓN CON LOS OBJETIVOS DE NEGOCIO DE UN ESTABLECIMIENTO PERUANO DE SALUD BUCAL

Facultad de Ingeniería, Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.

Anexo 2. Registro electrónico – Ficha técnica

1° Técnica de Cifrado Simétrica	Tiempo de Procesamiento (seg)				
	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5
AES 256 GCM	0.0009660 7	0.0000300 4	0.0025908 9	0.0000529 3	0.0000639 0
Tiempo Procesamiento Total	0.00370383 seg				
Transferencias	300	300	300	300	300
Total, de Transferencias	300				
Paquetes	1	1	1	1	1
Longitud de Claves	16 bits				

2° Técnica de Cifrado Asimétrica	Tiempo de Procesamiento (seg)				
	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5
SHA 512 Y RSA	0.0004651 5	0.0046680 0	0.0012109 3	0.0007569 8	0.0008950 2
Tiempo Procesamiento Total	0.00799608				
Transferencias	300	300	300	300	300
Total, de Transferencias	300				
Paquetes	1	1	1	1	1

Longitud de Claves	2048 bits

3° Técnica de Cifrado	**Tiempo de Procesamiento (seg)**				
	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5
AES 128 CBC	0.00003886	0.00001407	0.00110102	0.00002289	0.00002503
Tiempo Procesamiento Total	0.00120187seg				
Transferencias	300	300	300	300	300
Total, de Transferencias	300				
Paquetes	1	1	1	1	1
Longitud de Claves	16 bits				

4° Técnica de Cifrado	**Tiempo de Procesamiento (seg)**				
	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5
SHA 256	0.00116706	0.00032187	0.00132108	0.00110912	0.00058103
Tiempo Procesamiento Total	0.00450016 seg				
Transferencias	300	300	300	300	300
Total, de Transferencias	300				

Paquetes	1	1	1	1	1
Longitud de Claves	2,047 bits				