RIELAC, Vol. 46 (Publicación Continua):e971 (2025) ISSN:18-15-5928

RIELAC
Revista de Ingeniería Electrónica, Automática y Comunicaciones

Autenticación eficiente en la capa de percepción IoT con pruebas de conocimiento cero

Ernesto Rafael Carbonell-Rigores, Aramays Aimet Morales-Duran, Roberto Sepúlveda-Lima, Wenny Hojas-Mazo

RESUMEN

La adopción de la Internet de las Cosas en aplicaciones críticas expone la necesidad de robustecer la seguridad en su capa de percepción, una de las más vulnerables. Este artículo presenta un modelo de amenazas para dicha capa, a través de la identificación de los ataques de reproducción, negación de servicio y captura de tráfico de red como los más críticos. Para contrarrestarlos, se propone una variante optimizada de un protocolo de autenticación basado en pruebas de conocimiento cero, que mejora la eficiencia y escalabilidad del protocolo original de Hecht. La solución introduce matrices elementales para reducir la complejidad computacional del protocolo y un mecanismo explícito para la gestión segura del secreto. Se valida experimentalmente en un sistema de control de acceso mediante códigos QR, por la simulación de un entorno real de la Internet de las Cosas. Los resultados demuestran que la variante propuesta es ligera, eficiente y adecuada para dispositivos con recursos limitados, especialmente en entornos web, ofrece un alto nivel de seguridad, al no revelar información sobre la clave secreta durante la autenticación. Además, un diseño de experimentos optimiza los parámetros del protocolo y minimiza el tiempo de ejecución sin comprometer la seguridad. El protocolo propuesto representa una mejora en seguridad y eficiencia para la autenticación en la capa de percepción de la Internet de las Cosas.

Palabras clave: internet de las cosas, autenticación, capa de percepción, pruebas de conocimiento cero, matrices elementales

ABSTRACT

The adoption of the Internet of Things in critical applications highlights the need to strengthen security in its perception layer, one of the most vulnerable. This article presents a threat model for this layer, identifying replay, denial-of-service, and network traffic capture attacks as the most critical. In order to counteract them, an optimized variant of an authentication protocol based on zero-knowledge proofs is proposed, improving the efficiency and scalability of the original Hecht protocol. The solution introduces elementary matrices to reduce protocol computational complexity and an explicit mechanism for secure secret management. It is experimentally validated in a QR code-based access control system, simulating a real Internet of Things environment. The results show that the proposed variant is lightweight, efficient, and suitable for resource-constrained devices, especially in web environments, offering a high level of security by not revealing information about the secret key during authentication. Furthermore, a design of experiments optimizes the protocol parameters, minimizing execution time without compromising security. The proposed protocol represents a significant improvement in security and efficiency for authentication in the Internet of Things perception layer.

Keywords: internet of things, authentication, perception layer, zero-knowledge proofs, elementary matrices

Efficient Authentication in IoT Perception Layer with Zero-Knowledge Proofs

1. - INTRODUCCIÓN

La Internet de las Cosas (IoT, por sus siglas en inglés) se ha consolidado como una de las áreas tecnológicas más dinámicas y prometedoras de los últimos años. Este término hace referencia a la interconexión de objetos inteligentes a través de Internet, que permite su comunicación directa y autónoma para ofrecer servicios sin la necesidad de interacción humana o entre

Recibido: 03/2025 Aceptado: 06/2025

personas y computadoras [1]. La versatilidad de la IoT la posiciona como una solución innovadora en múltiples sectores, desde el monitoreo ambiental y la gestión de recursos, hasta la automatización de procesos industriales y domésticos [2].

La IoT ha sido ampliamente adoptada en aplicaciones como el control automático de iluminación en función de las condiciones ambientales, el seguimiento remoto de pacientes y la notificación de emergencias, la optimización del riego agrícola mediante sensores y la automatización del hogar, entre otras [2-5]. Sin embargo, junto con esta rápida expansión surgen preocupaciones críticas relacionadas con la seguridad y confidencialidad de la información gestionada en estos entornos. Se estimaba que en 2021, el número de dispositivos IoT conectados alcanzaría los 11,3 mil millones, y las proyecciones actuales indican un aumento a 29 mil millones para 2030 [6]. Este crecimiento exponencial acentúa la gravedad de las amenazas de seguridad, especialmente en aplicaciones críticas, como el caso registrado en 2017 por la Administración de Drogas y Alimentos de los EE.UU, que retiró del mercado casi medio millón de marcapasos debido a vulnerabilidades que podrían ser explotadas para alterar su funcionamiento. Además, entre enero y junio de 2021, Kaspersky reportó aproximadamente 1510 millones de filtraciones en dispositivos IoT, un incremento significativo respecto a los 639 millones registrados en 2020.

La arquitectura IoT comprende tres capas y cada una enfrenta amenazas específicas, pero la capa de percepción, donde los activos de información se generan y se capturan, resulta especialmente vulnerable. Esta está expuesta a ataques como la manipulación de hardware o software (*tampering*), el monitoreo de activos de información mediante *sniffing* y los ataques de denegación de servicio (DoS, por sus siglas en inglés) que afectan la comunicación entre dispositivos [7, 8].

En este contexto, la autenticación emerge como un mecanismo fundamental para garantizar la legitimidad de entidades en la capa de percepción. Los esquemas de autenticación en IoT deben cumplir requisitos como baja potencia computacional, limitaciones de memoria y compatibilidad con múltiples protocolos de red [9]. Entre los protocolos existentes, las pruebas de conocimiento cero (ZKP, por sus siglas en inglés) destacan por su eficiencia y seguridad, ya que permiten verificar declaraciones sin revelar información secreta alguna inherente al protocolo [10].

En [11], se propuso un protocolo basado en pruebas de conocimiento cero, fundamentado en el uso de matrices de Hill y el Problema de la Descomposición Simétrica Generalizada (GSDP, por sus siglas en inglés) como función de una vía. Este enfoque aprovecha la aritmética de simple precisión para garantizar su aplicabilidad en dispositivos de bajas prestaciones computacionales, mientras que el GSDP asegura que la recuperación de información secreta sea computacionalmente intratable, o sea, que el ataque requiera fuerza bruta. Sin embargo, esta propuesta presenta limitaciones prácticas, como la falta de definición sobre la inclusión del secreto (por ejemplo, la clave del usuario) dentro del protocolo y la baja probabilidad de que las matrices generadas de forma aleatoria sean invertibles, lo que incrementa significativamente la complejidad computacional [11].

Las contribuciones científicas fundamentales que se hacen en este trabajo son:

1. Proponer una variante eficiente del protocolo de autenticación presentado en [11], optimizada para su implementación en dispositivos de bajas prestaciones en IoT.
2. Incorporar un mecanismo explícito para la gestión segura del secreto dentro del protocolo, garantizando el cumplimiento de la propiedad de conocimiento cero.
3. Reducir la complejidad computacional asociada a la generación de matrices invertibles, así como al producto, a la inversión y a la potenciación de matrices, mediante algoritmos optimizados.
4. Evaluar la implementación del protocolo propuesto en un entorno real de IoT, con métricas de rendimiento y seguridad específicas de la capa de percepción.

Estas contribuciones responden a la necesidad de mitigar las amenazas de seguridad en la mencionada capa de la IoT mediante soluciones eficientes y prácticas. Por lo tanto, el objetivo general del trabajo consiste en implementar una variante mejorada y eficiente del protocolo de autenticación propuesto en [11] para la capa de sensores en aplicaciones IoT.

2. - INTERNET DE LAS COSAS (IOT)

La Internet de las Cosas, también conocida como IoT, ha emergido como una tecnología transformadora que redefine la interacción entre el mundo físico y el digital. Este paradigma se fundamenta en una red de dispositivos interconectados, capaces de comunicarse entre sí y con la nube, facilitando la automatización y el control inteligente [1, 2]. A diferencia de la Internet tradicional, centrada en la comunicación entre personas, la IoT integra, en una red global, objetos cotidianos que van desde electrodomésticos y vehículos hasta maquinaria industrial. Esta integración permite que estos dispositivos, equipados con sensores y actuadores, recopilen datos de su entorno y respondan, de manera inteligente, a las necesidades de los usuarios, optimizando procesos y mejorando la eficiencia [1, 2]. Las capacidades disruptivas de la IoT han impulsado su adopción en

una amplia gama de sectores, generando un impacto significativo en la sociedad [2]. Entre las áreas de mayor influencia, se pueden destacar las ciudades inteligentes [5], la atención médica [3], el comercio minorista y la logística [12], así como la agricultura y la ganadería [4]. La creciente implementación de estas aplicaciones va de la mano de un aumento exponencial en el número de dispositivos y sensores inteligentes interconectados. Estos dispositivos, a menudo, discretos y poco perceptibles, operan de forma ubicua, comunicándose de manera inalámbrica y autónoma en cualquier momento y lugar [2]. La gestión efectiva de la complejidad inherente a esta tecnología requiere el establecimiento de una arquitectura robusta y bien definida, que facilite la interoperabilidad y la escalabilidad.

La arquitectura típica de un entorno IoT se estructura en tres capas fundamentales: la capa de aplicación, la capa de red y la capa de percepción, cada una con funciones y responsabilidades específicas [1]. La capa de percepción es la más cercana al mundo físico y está compuesta por una gran variedad de elementos heterogéneos, incluyendo sensores y actuadores. Estos dispositivos se caracterizan por una amplia diversidad en sus capacidades de computación, almacenamiento, comunicación y fuente de alimentación [1, 7-9]. Mientras que algunos, como los medidores inteligentes, poseen una capacidad de procesamiento considerable, que les permite realizar cálculos complejos, otros, como las bombillas inteligentes, tienen una capacidad computacional muy limitada, apenas suficiente para operaciones simples. En general, la mayoría de los dispositivos de la capa de percepción se caracterizan por sus recursos limitados, tanto en términos de capacidad de procesamiento como de energía disponible. Esta limitación restringe su capacidad para ejecutar tareas computacionalmente intensivas, lo que impone restricciones significativas en el diseño de aplicaciones y protocolos para esta capa.

La capa de red, por otro lado, actúa como un puente entre la capa de percepción y la capa de aplicación. Su función principal consiste en recibir la información recopilada por los sensores y actuadores de la capa de percepción y transmitirla de manera eficiente a través de una variedad de tecnologías de comunicación, como Bluetooth, WiFi y LTE, entre otras [1, 2]. La capa de red debe enrutar los datos hacia o desde diferentes dispositivos y aplicaciones, utilizando interfaces o puertas de enlace para interconectar redes heterogéneas y soportar diversos protocolos y tecnologías de comunicación. Finalmente, la capa de aplicación, también conocida como la capa de negocio, se sitúa en el nivel superior de la arquitectura IoT. Esta capa recibe los datos transmitidos desde la capa de red y los utiliza para proporcionar servicios específicos a los usuarios finales. Estos servicios pueden incluir el almacenamiento de datos para copias de seguridad, el análisis de datos para la predicción del estado futuro de los dispositivos físicos, y la generación de reportes e informes para la toma de decisiones [1, 2].

La seguridad en la Internet de las Cosas es una preocupación de máxima prioridad, especialmente considerando que su principal medio de comunicación es Internet, un entorno inherentemente abierto y susceptible a ciberataques de diversa índole [13]. Los desafíos de seguridad en el contexto de la IoT abarcan un amplio espectro de aspectos que van, desde la configuración inicial del sistema hasta el almacenamiento y la gestión de la información, pasando por la preservación de la confidencialidad, el control de acceso y la autenticación de dispositivos y usuarios [7-9].

La capa de percepción, al estar compuesta por dispositivos físicos directamente expuestos al entorno, es particularmente vulnerable a una serie de amenazas de seguridad. La naturaleza frecuentemente desatendida de estos dispositivos, así como su limitada capacidad de procesamiento y almacenamiento, los convierte en objetivos atractivos para los atacantes. Para clasificar y comprender mejor estas amenazas, se utiliza la taxonomía STRIDE, un modelo ampliamente adoptado en el ámbito de la seguridad informática [14], pero que adquiere particularidades críticas en el contexto de la capa de percepción de la IoT. La suplantación de identidad (Spoofing) consiste en falsificar cualquier elemento de identificación (direcciones MAC, credenciales o firmas digitales) para suplantar dispositivos legítimos, siendo especialmente relevante en IoT, debido a los frecuentes mecanismos de autenticación débiles [15]. La manipulación de datos (*Tampering*) implica la modificación maliciosa de datos en tránsito o almacenados, lo que puede corromper permanentemente mediciones de sensores o configuraciones críticas en dispositivos IoT con recursos limitados [14, 15]. El repudio se ve agravado por la habitual ausencia de sistemas robustos de registro de eventos en estos dispositivos [14]. La revelación de información (*Information disclosure*) ocurre cuando activos de información sensibles son expuestos mediante monitoreo no autorizado o acceso físico, riesgo acentuado en IoT por el almacenamiento de credenciales en el software de dispositivo integrado y comunicaciones a menudo no cifradas [14, 15]. La negación de servicio (*Denial of Service*) adquiere nuevas dimensiones en IoT, donde ataques pueden agotar deliberadamente la batería mediante solicitudes maliciosas frecuentes, saturar la memoria limitada, bloquear canales de comunicación inalámbricos o explotar protocolos de bajo consumo energético [14, 15]. Finalmente, la elevación de privilegios (*Elevation of privileges*) ocurre cuando se explotan interfaces de administración expuestas, software de dispositivo integrado obsoleto o configuraciones por defecto inseguras, para obtener acceso privilegiado a estos dispositivos [13, 15]. Un atacante podría comprometer el dispositivo mediante software malicioso diseñado para: (a) acceder a interfaces administrativas no autorizadas, o (b) ejecutar códigos de aprovechamiento de vulnerabilidades locales para escalar privilegios.

Para garantizar un nivel adecuado de seguridad en la capa de percepción de la IoT, es imperativo que los sistemas y dispositivos cumplan con una serie de requisitos o propiedades fundamentales. Estas propiedades se basan en el estándar ISO/IEC 27000 y son confidencialidad, integridad y disponibilidad [8, 9, 13]. La confidencialidad exige que la información sólo pueda ser accedida por entidades debidamente autorizadas, impidiendo el acceso a activos de información privados por

parte de usuarios no autorizados. La integridad, por su parte, garantiza que los datos no sean modificados de manera no autorizada durante su transmisión o almacenamiento, asegurando que el destinatario reciba exactamente la información que fue enviada por la fuente. La disponibilidad implica que los recursos y la información deben estar accesibles para las entidades legítimas en todo momento, independientemente del lugar y el momento en que se requieran. La autenticación se refiere al proceso de verificación de la identidad de los dispositivos y usuarios que interactúan en el sistema, siendo un elemento crucial para prevenir el acceso no autorizado y garantizar la legitimidad de las comunicaciones. La autorización, en tanto, determina los derechos y privilegios de cada usuario o dispositivo, especificando las acciones que pueden realizar (lectura, escritura, eliminación) y las reglas de control de acceso que permiten o deniegan permisos a los dispositivos IoT. Finalmente, el no-repudio asegura que las transacciones y comunicaciones realizadas en el sistema sean irrefutables, es decir, que ninguna de las partes involucradas pueda negar su participación en una comunicación o transacción.

2.4. – LA AUTENTICACIÓN COMO FOCO DE LA SEGURIDAD EN IOT

La autenticación constituye un requisito crucial de seguridad en el contexto de la Internet de las Cosas, ya que constituye la primera línea de defensa para garantizar la confidencialidad e integridad de la información que se transmite y almacena en los dispositivos y sistemas IoT [9]. Sin una autenticación robusta, los dispositivos son vulnerables al acceso no autorizado, la manipulación de activos de información y otros ataques maliciosos. Sin embargo, los esquemas de autenticación tradicionales, desarrollados para entornos informáticos convencionales, no son directamente aplicables a la IoT, debido a las características específicas de los dispositivos que componen la capa de percepción. Estos dispositivos se caracterizan por sus limitaciones en términos de capacidad de procesamiento, memoria y energía disponible [9].

Los sistemas de autenticación tradicionales basados en criptografía asimétrica fueron creados para computadoras centrales de los años 70-80, con capacidades muy superiores a las de los dispositivos IoT actuales. Estos algoritmos presentan tres problemas fundamentales en entornos IoT que disponen de recursos limitados [9]: (1) alto consumo energético (hasta 100 veces mayor que alternativas modernas, que afecta negativamente el rendimiento y la vida útil de la batería), (2) requerimientos de procesamiento excesivos para microcontroladores de bajo rendimiento, y (3) necesidad de memoria RAM frecuentemente superior a la disponible en estos dispositivos. Estas limitaciones persisten incluso en equipos IoT más potentes, haciendo preferibles esquemas criptográficos diseñados específicamente para hardware restringido. Además, los esquemas de autenticación basados en el uso de nombres de usuario y contraseñas son intrínsecamente vulnerables a diversos tipos de ataques, como los de fuerza bruta, el *phishing* y el robo de credenciales. Si bien la simplicidad de estos esquemas los hace fáciles de usar, también los convierte en un blanco fácil para atacantes [9]. Para que un mecanismo de autenticación sea efectivo y viable en el contexto de la Internet de las Cosas, debe cumplir con una serie de requisitos específicos que se derivan de las características y limitaciones de los dispositivos IoT [9]. Estos requisitos se encaminan a asegurar que el mecanismo de autenticación deba ser aplicable a todas las capas de la arquitectura IoT, garantizando la autenticación de dispositivos y usuarios, tanto en la capa de aplicación como en la capa de red y, especialmente, en la crítica capa de percepción.

Los protocolos de conocimiento cero, conocidos como ZKP por sus siglas en inglés, han surgido como una solución prometedora para la autenticación en entornos con recursos limitados, como es el caso de la Internet de las Cosas. Estos protocolos criptográficos permiten a una de las partes, denominada probador, demostrar a la otra parte, denominada verificador, que una afirmación es cierta, sin necesidad de revelar ninguna información adicional, más allá de la veracidad de la afirmación en sí [10]. Esta característica los hace especialmente atractivos para la IoT, donde la confidencialidad y la eficiencia son de suma importancia. Debido a su ligereza computacional y a su alto nivel de seguridad, los ZKP se perfilan como un método de autenticación eficaz para los dispositivos que operan en un escenario IoT [10, 16].

Para que un protocolo criptográfico pueda ser considerado como una prueba de conocimiento cero, debe satisfacer tres propiedades fundamentales [17]: la integridad, con la cual se establece que, ante un probador que dice la verdad y proporciona información válida, el verificador siempre quedará convencido de la veracidad de la afirmación dicha; la solidez, que garantiza que si el probador está mintiendo o intentando engañar al verificador, no podrá convencerlo de la veracidad de su afirmación, excepto con una probabilidad extremadamente baja, que puede ser despreciable en la práctica; y, finalmente, el conocimiento cero, la propiedad más característica de estos protocolos, que asegura que el verificador no aprende absolutamente nada, más allá del hecho de que la afirmación del probador es cierta. Es decir, el verificador no obtiene ninguna información adicional sobre el secreto del probador, ni siquiera de forma indirecta.

A pesar de sus indudables ventajas, los protocolos de conocimiento cero tradicionales presentan ciertas limitaciones que dificultan su aplicación directa en el contexto de la IoT. Estos protocolos se basan en la complejidad de problemas matemáticos clásicos, como el Problema de la Factorización de Enteros (IFP, por sus siglas en inglés) y el Problema del Logaritmo Discreto (DLP, por sus siglas en inglés), entre otros [16]. Si bien estos problemas son computacionalmente difíciles de resolver para las computadoras convencionales, su resolución implica un costo computacional significativo, lo que los hace poco prácticos para dispositivos con recursos limitados. Además, la seguridad de estos protocolos, frente a la computación cuántica, es un

tema de debate, ya que se han propuesto algoritmos cuánticos que, en teoría, podrían resolver estos problemas en tiempo polinomial, comprometiendo la seguridad de los ZKP tradicionales a largo plazo [16].

Con el objetivo de superar las limitaciones de los ZKP tradicionales, Hecht propone un protocolo de conocimiento cero innovador que se basa en el grupo de matrices de Hill (sus elementos son congruencias módulo *p*) con módulo *p=251*, un espacio algebraico no conmutativo [11]. El empleo de congruencias módulo *p* propicia que, para las operaciones sobre los elementos o entradas matriciales, sea suficiente aritmética de simple precisión. La seguridad de este protocolo reside en la dificultad computacional de resolver el Problema de la Descomposición Simétrica Generalizada (GSDP, por sus siglas en inglés) que es considerado computacionalmente intratable, incluso para las computadoras cuánticas, ya que no se han documentado, hasta la fecha, ataques efectivos y eficientes contra él [11].

El protocolo propuesto por Hecht presenta dos ventajas principales que lo hacen atractivo para su aplicación en la IoT. En primer lugar, ofrece un alto nivel de seguridad a largo plazo, por basarse en GSDP, incluso en un escenario en el que la computación cuántica sea una realidad. En segundo lugar, su diseño está orientado a la eficiencia computacional, lo que lo hace adecuado para su implementación en dispositivos con bajas prestaciones, como los comúnmente utilizados en la capa de percepción de la IoT.

A pesar de sus ventajas, el protocolo de Hecht presenta una serie de desafíos que dificultan su implementación práctica en entornos reales de IoT, especialmente en la capa de percepción. Una de las principales desventajas radican en la necesidad de generar aleatoriamente matrices invertibles, un proceso que no está garantizado dentro del propio protocolo. Hecht señala que la probabilidad de generar aleatoriamente una matriz invertible es de solo el 0.4%, lo que obliga a repetir el proceso de generación hasta obtener una matriz que cumpla con este criterio [11]. La verificación de la no singularidad de una matriz es, en sí misma, una operación computacionalmente costosa, con una complejidad de orden $O(n^3)$ en el mejor de los casos o, incluso, O(n!) en el peor, lo que la hace prohibitiva para dispositivos con recursos limitados. Otra limitación importante es la falta de un mecanismo explícito para la gestión del secreto (por ejemplo, una contraseña) dentro del protocolo. El mencionado protocolo no especifica en qué fase ni de qué manera se incorporaría el secreto del usuario, lo que plantea interrogantes sobre cómo se integraría dicho protocolo de autenticación en un sistema informático concreto. Un tercer desafío aparece con la recomendación de realizar 10 rondas o iteraciones para la verificación que lleva a cabo el protocolo, cifra que puede tener un impacto negativo en el rendimiento, especialmente en aplicaciones en tiempo real, debido a la complejidad de las operaciones matriciales que se ejecutan en cada ronda. Finalmente, el protocolo no ofrece garantías de que las matrices utilizadas para la generación de las claves privadas sean solo periódicas para potencias no menores que un valor Z□Z especificado, un aspecto que, según el propio Hecht, se deja como una recomendación para futuras investigaciones, pero que podría tener implicaciones para la seguridad del protocolo al aplicar GSDP. Vale aclarar que la matriz de Hill *M* se denomina periódica, si satisface que $M^{h+1}=M$, para un cierto entero positivo *h*. Se denomina período de la matriz *M* al menor *h* que hace periódica a *M*. En la propuesta original, Hecht sugiere fijar el mencionado valor *Z* en (al menos) 32, como margen razonable de seguridad, dado que el protocolo lo utiliza como cota superior de los exponentes a los que se potencian las matrices al aplicar GSDP. En adición, un valor de *h* inferior a *Z* pudiera disminuir la dificultad o intratabilidad de resolver GSDP [11].

3. - SOLUCIÓN PROPUESTA

La presente sección introduce la solución propuesta para fortalecer la seguridad en la capa de percepción de la IoT: un protocolo de autenticación basado en pruebas de conocimiento cero (ZKP) que se distingue por su eficiencia computacional y su robustez frente a ataques. Esta propuesta se fundamenta en una variante optimizada del protocolo ZKP presentado en [11], adaptada específicamente a las restricciones de recursos inherentes a los dispositivos que conforman la capa de percepción. El objetivo principal es proporcionar un mecanismo seguro y ligero de autenticación, que no comprometa el rendimiento de los dispositivos ni la experiencia del usuario. Para ilustrar la aplicabilidad práctica del protocolo, se describe el desarrollo e implementación de un sistema de control de acceso basado en códigos QR, que simula un escenario real de IoT y permite la validación experimental de la solución propuesta. A continuación, se detallan los aspectos clave del modelo de amenazas para la capa de percepción, seguido de una descripción exhaustiva del protocolo ZKP propuesto. Se incluye su arquitectura, las modificaciones realizadas para mejorar la eficiencia de la propuesta original de Hecht, y un análisis de su efectividad contra ataques.

3.1. - MODELO DE AMENAZAS PARA LA CAPA DE PERCEPCIÓN EN IOT

La seguridad en la capa de percepción es un aspecto crítico para garantizar la integridad y confiabilidad de todo el sistema IoT. Dada la naturaleza expuesta de los dispositivos en esta capa y su proximidad al entorno físico, resulta fundamental identificar y mitigar las amenazas potenciales que podrían comprometer su funcionamiento. En este contexto, se presenta un modelo de amenazas específico para la capa de percepción, basado en la metodología de Microsoft [18], que permite un

análisis sistemático de los riesgos y la definición de estrategias de mitigación efectivas. Este modelo se articula en torno a cuatro fases fundamentales: la recopilación de información general, la creación y análisis del modelo de amenazas, la revisión de amenazas, y la identificación de técnicas y tecnologías de mitigación. A continuación, se detalla cada una de estas fases, y se proporciona una descripción exhaustiva del proceso seguido y los resultados obtenidos.

La creación de un modelo de amenazas sólido requiere comprender a fondo el sistema analizado. En la capa de percepción de la IoT, se recopila información clave sobre casos de uso, recursos a proteger y límites del entorno. En esta capa intervienen tres actores principales: sensores (también conocidos como dispositivos de detección), que recopilan datos del entorno (por ejemplo, parámetros ambientales como temperatura, humedad, etc.), dispositivos controladores (denominados también dispositivos de borde), que gestionan la comunicación y el procesamiento de la información (como los dispositivos Raspberry Pi, que procesan los datos y filtran la información irrelevante), y actuadores, que ejecutan acciones automatizadas. Entre los actuadores más comunes se encuentran relés, indicadores luminosos, electroválvulas y motores [19]. Según estos roles, se identifican tres casos de uso: recopilación de datos, gestión de información y ejecución de acciones, respectivamente. Los dispositivos de borde, cuentan con mayores recursos y múltiples interfaces de comunicación, que permiten la integración y optimización del sistema [19].

Es importante destacar que en estos procesos se maneja una gran cantidad de información sensible. Los dispositivos de detección recopilan activos de información confidenciales que deben ser protegidos contra accesos no autorizados y manipulaciones. La transmisión de datos entre los dispositivos se realiza a través de la comunicación de máquina a máquina (M2M), sin intervención humana [1, 2]. La heterogeneidad es una característica intrínseca de los dispositivos IoT, que se manifiesta en los formatos de entrada y salida de la información, los protocolos de comunicación, la complejidad de los cálculos que realizan, entre otros aspectos [19]. Para que todos estos dispositivos heterogéneos puedan interactuar de manera efectiva, es fundamental que comprendan la información que se captura y transfiere, y que sean capaces de realizar cálculos y generar observaciones a partir del análisis de los activos de información obtenidos.

En la seguridad informática de la IoT, el principal activo de información a proteger en la capa de percepción es la información manejada por los dispositivos, ya que puede contener datos sensibles, cuyo acceso no autorizado, alteración o manipulación podrían comprometer el sistema. Dado que el funcionamiento de la IoT depende del flujo continuo de esta información, cualquier vulneración de su confidencialidad, integridad o disponibilidad podría causar fallos críticos. Debido a su constante transferencia y procesamiento en la red, la exposición de este activo de información es alta, y su valor clasifica como crítico.

La implementación de la tecnología IoT conlleva a una serie de desafíos que deben ser considerados. Entre las limitaciones más importantes se encuentran el ancho de banda y el consumo de energía [7]. Los dispositivos IoT son ligeros, de bajo consumo y con memoria limitada, lo que exige optimizar el ancho de banda y la energía. La arquitectura IoT es compleja, con dispositivos heterogéneos que deben interoperar, lo que dificulta su gestión a medida que crece la red. La detección de fallos es crucial para monitorear el estado de los dispositivos. Además, debido a las limitaciones de memoria, la seguridad en IoT requiere mecanismos criptográficos de bajo costo y mínima sobrecarga, ya que los algoritmos tradicionales no son viables.

Una vez recopilada la información básica sobre la capa de percepción y su entorno operativo, se procede a identificar las posibles amenazas que podrían afectar a los escenarios de uso definidos. Como resultado de este análisis, se identificó un conjunto de amenazas relevantes, clasificadas de acuerdo con la taxonomía STRIDE [14]. Las principales amenazas a la confidencialidad y confidencialidad en la capa de percepción de la IoT incluyen: Ataque de Reproducción, mediante el cual un adversario retransmite información legítima para obtener acceso; Negación de Servicio (DoS, por sus siglas en inglés), que bloquea la comunicación en la red IoT; Captura de un Nodo, que permite acceso físico y extracción de datos sensibles; Inyección de Código Malicioso, que compromete dispositivos mediante software malicioso; y Captura de Tráfico de Red, a través del cual la información transmitida es interceptada o manipulada. Estas amenazas comprometen la seguridad del sistema y requieren medidas de mitigación efectivas.

Luego de identificar las amenazas potenciales y las vulnerabilidades que podrían ser explotadas, se determina el nivel de riesgo asociado a cada una de ellas. La estimación de la probabilidad de un ataque, según la metodología propuesta por el NIST [20], se basa en la evaluación de tres factores: la motivación y la capacidad del atacante, el atractivo de la vulnerabilidad para el atacante, y la existencia y efectividad de los controles de seguridad implementados. Para cuantificar la probabilidad de que una vulnerabilidad sea explotada, se utiliza una escala cualitativa de tres niveles: Alta (3), Media (2) y Baja (1). El siguiente paso en la evaluación del riesgo consiste en estimar el impacto que tendría un ataque exitoso en el sistema IoT y en la organización, evaluando las consecuencias negativas que se derivarían del peor escenario posible. Para evaluar el nivel de impacto, se propone utilizar una escala cualitativa de cinco niveles: Bajo (1), Bajo-Medio (2), Medio (3), Medio-Alto (4) y Alto (5).

Teniendo en cuenta la información analizada, se ha evaluado el riesgo asociado a los posibles ataques a la capa de percepción de un escenario IoT. Para calcular el riesgo, se multiplica la probabilidad de ataque por el nivel de impacto (Riesgo =

Probabilidad de ataque * Impacto). Los valores de riesgo obtenidos se clasifican en tres niveles: Bajo (1-5), Medio (6-10) y Alto (11-15).

A continuación, se presenta la evaluación del riesgo para cada una de las amenazas identificadas: A01 - Ataque de Reproducción: Probabilidad de ataque: 3, Impacto: 5, Riesgo: 15 (Alto); A02 - Negación de Servicio: Probabilidad de ataque: 3, Impacto: 5, Riesgo: 15 (Alto); A03 - Captura de un Nodo: Probabilidad de ataque: 1, Impacto: 5, Riesgo: 5 (Bajo); A04 - Inyección de Código Malicioso: Probabilidad de ataque: 2, Impacto: 3, Riesgo: 6 (Medio); A05 - Captura de Tráfico de Red: Probabilidad de ataque: 3, Impacto: 5, Riesgo: 15 (Alto).

De acuerdo con el análisis de riesgos realizado, las amenazas de mayor impacto para la capa de percepción de la IoT son el ataque de reproducción (A01), el ataque de denegación de servicio (A02) y la captura de tráfico de red (A05). Para simular un escenario IoT donde se pueda verificar la efectividad de las medidas de mitigación, se propone el desarrollo de una aplicación con una arquitectura IoT representativa. Un ejemplo de aplicación práctica y ampliamente utilizada en la actualidad es un sistema de control de acceso mediante escaneo de códigos QR [21, 22, 23]. Para el desarrollo de los diferentes componentes del sistema de control de acceso, se ha seleccionado Node.js con el *framework* NestJS para el *backend* de la plataforma. Como sistema de gestión de bases de datos, se utilizará PostgreSQL. Para el *frontend* de la plataforma, se eligió React, mientras que para el desarrollo de la aplicación móvil del cliente (escáner de códigos QR) se empleó React Native.

La implementación de una variante del protocolo ZKP para autenticación en la capa de percepción de la IoT depende de las tecnologías utilizadas en el sistema de control de acceso. El lado del servidor se integra en el backend desarrollado con NestJS, usando JavaScript y TypeScript, mientras que el lado del cliente se implementa en React y React Native para la plataforma web y la aplicación móvil, respectivamente. Se busca una autenticación robusta y eficiente, con bajo consumo computacional, para mitigar amenazas en la IoT. Por ello, se propone una variante optimizada del protocolo ZKP de Hecht para la capa de sensores.

3.2. - PROTOCOLO ZKP PROPUESTO

En esta sección, se presenta una variante optimizada del protocolo de autenticación de conocimiento cero (ZKP) propuesto en [11], diseñada específicamente para entornos con recursos computacionales limitados, como la capa de percepción de la Internet de las Cosas. La propuesta se enfoca en la reducción de la complejidad computacional y la consecuente mejora de la eficiencia del protocolo original, manteniendo, al mismo tiempo, un alto nivel de seguridad. Además, se describe el despliegue de una aplicación de control de acceso que utiliza códigos QR para ilustrar la aplicación práctica del protocolo propuesto en un escenario real de IoT.

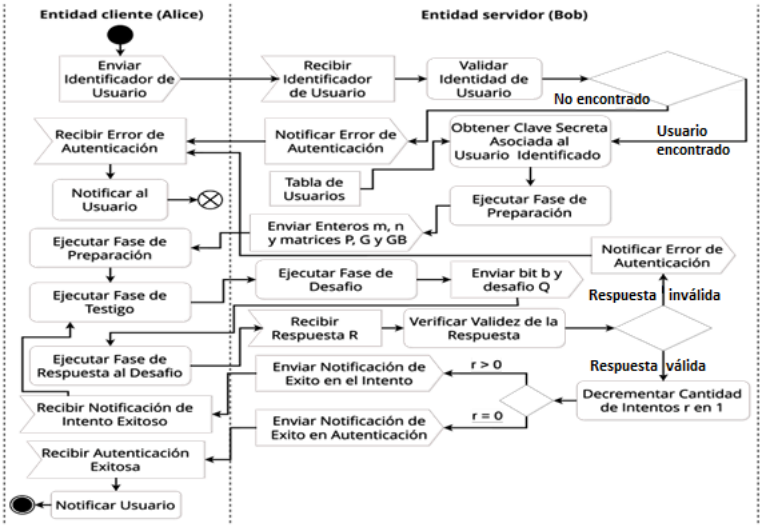

Figura 1

Diagrama de actividades del protocolo ZKP.

3.2.1. - ARQUITECTURA GENERAL DEL PROTOCOLO PROPUESTO

El protocolo propuesto sigue una arquitectura cliente-servidor, en la que el cliente (por ejemplo, un dispositivo IoT en la capa de percepción) busca autenticarse ante un servidor. La Fig. 1 ilustra el flujo principal del algoritmo.

El proceso comienza cuando la entidad cliente obtiene el identificador único del usuario y la clave secreta asociada. El método específico para obtener esta información depende del esquema de autenticación subyacente. Por ejemplo, en un esquema tradicional de nombre de usuario y contraseña, estos valores se obtendrían directamente de la entrada del usuario. El identificador único podría ser el nombre de usuario y la clave secreta sería la contraseña. Una vez que el cliente posee esta información, la envía al servidor, que verifica la existencia del usuario en el sistema. Si el usuario no se encuentra, se notifica un error y el proceso finaliza. En caso contrario, se inicia el protocolo de conocimiento cero. Finalmente, el servidor acepta o rechaza la identidad del usuario, basándose en el resultado del protocolo.

3.2.2. - DESCRIPCIÓN DEL PROTOCOLO ZKP PROPUESTO

La variante propuesta en este trabajo modifica el protocolo original presentado en [11], para superar sus limitaciones y facilitar su implementación en sistemas reales. Los principales cambios se centran en la fase de preparación del protocolo, sustituyendo algunas "matrices generales" por "matrices elementales". En álgebra lineal, se denomina "generales" a las matrices cuyos elementos (entradas) o sus respectivas magnitudes no siguen una disposición particular o patrón, ni se obtienen a partir de fórmulas que involucran estructuras algebraicas más simples. Por contraste, se denomina "estructuradas" a las que sí responden a tales patrones o fórmulas, como las diagonales, triangulares o elementales, por ejemplo; su estructura se explota para reducir el espacio para almacenarlas y el número de operaciones al manipularlas [24].
Una matriz elemental *A*, de orden *n*, conocida también como transformación elemental, es aquella que tiene la forma:

$$A=I-u*v^T, \tag{1}$$

en la que *I* es la matriz identidad de orden *n*, y *u* y *v* son vectores columna de orden *n* [24, 25]. Para su representación interna, en lugar de almacenar *n*x*n* elementos, solo se reserva espacio para los vectores *u* y *v*, es decir, solo *2*x*n* valores. Por otro lado, su inversa es, también, elemental y se obtiene mediante la expresión:

$$A^{-1}=I-(u*v^T)/(v^T*u-1). \tag{2}$$

El escalar $s=v^T*u$ se obtiene mediante una operación de orden O(*n*); verificar si *A* es no singular se circunscribe solo a comprobar si $s \neq 1$. En ese caso, al escalar $k=1/(s-1)$ se le denomina, en este trabajo, "factor de inversión" de la matriz elemental *A*, mediante el cual A^{-1} se obtiene con un esfuerzo computacional O(*n*), según:

$$A^{-1}=I-k*u*v^T. \tag{3}$$

El vector *v* de A^{-1} coincide con el de *A*, y el vector *u* de A^{-1} resulta del producto de *k* por el vector *u* de *A*. Aquí, se propone incluir el factor *k* como parte de la representación interna de cada matriz elemental. Por su parte, el producto de una matriz general *B* por la matriz elemental *A* se lleva a cabo mediante la resta matricial:

$$B*A=B*(I-u*v^T)=B-B*u*v^T, \tag{4}$$

que realiza dos productos matriz-vector ($B*u$ y $[B*u]*v^T$) y una resta de matrices; en total, se requiere un esfuerzo de orden $O(n^2)$, inferior al orden cúbico del producto de matrices generales. Resulta simple deducir el producto *A*B* y el esfuerzo que implica. Por último, la potencia *m* de *A* se obtiene mediante la expresión:

$$A^m = I - \frac{1-(1-s)^m}{s} * u * v^T, \tag{5}$$

que es una matriz elemental para la que el escalar $s=v^T*u$ no debe ser nulo. Luego, el vector *u* de $A'=A^m$ se obtiene de multiplicar el escalar $e=(1-(1-s)^m)/s$ por el vector *u* de *A* y el vector *v* de *A'* coincide con el de *A*. En suma, el proceso implica un costo de orden O(*n*), también inferior al cúbico implicado en la potenciación de las matrices generales.

A continuación, se describe en detalle el protocolo aquí propuesto. Como en la propuesta original, se denotan por *Alice* y *Bob* a las entidades que en él intervienen. Todas las matrices y vectores manipulados son de Hill, con orden 8 y módulo 251. Se comienza por la **Inicialización** (Lado del Servidor – Bob):

1. Alice envía su identificación a Bob (nombre de usuario o identificador del dispositivo). Bob la recibe y verifica que Alice se encuentra registrada en el sistema.
2. Bob busca la clave secreta que tiene asociada a Alice (la contraseña o token secreto del dispositivo) y utiliza un algoritmo de cifrado para transformar dicha información en una matriz elemental, la cual se denota por *E*.
3. Bob genera las matrices *P*, *G* y *Db* como matriz elemental, general y diagonal, respectivamente. Para obtener *P*, se generan dos vectores "aleatorios" no nulos, *u* y *v*; entre los cambios aquí sugeridos, se propone calcular la suma de los

productos de cada par de entradas (u_i, v_i), en la medida que se generan, imponiendo que $u_i \ \bar{v}_i$ $(1 \leq i \leq 8)$, y que la generación de la última entrada de v sea iterada, hasta que la suma total de dichos productos sea distinta de 1. Por su parte, las entradas de *Db* deben ser no nulas y distintas dos a dos. Además, se generan "aleatoriamente", del intervalo [2, *Z*] (con Z=32), los enteros positivos m y n, que deben ser diferentes. Bob define, también, la cantidad r de veces que se deberá iterar el protocolo, antes de quedar satisfecho con la identidad de Alice.

4. Bob calcula la matriz G_b, por medio de la ecuación:

$$G_b = E^m * G * E^n \tag{6}$$

Para implementar esta, y las restantes ecuaciones que aquí se presentan, se emplea el denominado enfoque sistemático, en virtud del cual, en lugar de diseñar un algoritmo para aplicar la ecuación tal y como se presenta, se le debe descomponer y simplificar al máximo posible, entre otras, a través de sustituciones y la aplicación de propiedades, en aras de no repetir cálculos ya realizados y, en consecuencia, realizar la menor cantidad de operaciones. Por ejemplo, para la implementación de (6), se utiliza (5), según:

$$Gb = \left(I - \frac{1-(1-s)^m}{s} * u * v^T\right) * G * \left(I - \frac{1-(1-s)^n}{s} * u * v^T\right) \tag{7}$$

expresión que se descompone y simplifica, para diseñar el algoritmo con la ecuación ya reducida.

5. Bob calcula su clave privada B y su clave pública *GB*; se introduce una modificación, según la cual se sustituye el uso de G por la matriz G_b en la ecuación para la clave pública, como se muestra a continuación:

$$B = P * Db * P^{-1} \tag{8}$$
$$GB = B^m * G_b * B^n. \tag{9}$$

6. Bob envía la información pública (P, G, m, n y *GB*) a Alice.

A continuación se pasa a la fase de **preparación** (Lado del Cliente – Alice):

7. Alice recibe la información pública de Bob y utiliza un mecanismo de cifrado para transformar su información secreta (que desea demostrar a Bob) en una matriz elemental, la cual se denota por F. Además, genera la matriz diagonal *Da*, con entradas no nulas y distintas entre sí.

Se pasa, entonces a la fase de **testigo** (Lado del Cliente – Alice):

8. Alice calcula la matriz G_a, que se calcula por la ecuación:

$$G_a = F^m * G * F^n \tag{10}$$

9. Alice obtiene su clave privada A, para la que genera la matriz diagonal *Da*, con entradas no nulas y distintas dos a dos, calcula su clave pública *GA* y el testigo S, para el cual se genera un entero positivo "aleatorio" l del intervalo [2, *Z*], distinto de m y de n, según:

$$A = P * Da^m * P^{-1} \tag{11}$$
$$GA = A^m * G_a * A^n \tag{12}$$
$$S = A^l * GA * A^{-m} \tag{13}$$

10. Alice envía a Bob la clave pública *GA* y el testigo S.

Seguidamente, se pasa a la etapa de **desafío** (Lado del Servidor – Bob):

11. Bob recibe el testigo y la llave pública de Alice y genera un bit b (0 ó 1) aleatorio y define uno de dos desafíos:
    - Si $b = 0$, entonces genera al azar una matriz elemental H y calcula $Q = B^m * H * B^n$. (14)
    - Si $b = 1$, entonces calcula $Q = B^m * S * GA * B^n$. (15)

Bob envía el desafío Q y el bit b a Alice.

Posteriormente, se pasa a la fase de **respuesta** (Lado del Cliente – Alice):

12. Alice recibe el desafío de Bob y calcula una de las dos respuestas posibles y se la envía a Bob:
    - Si $b = 0$, entonces $R = S^{-m} * Q * S^{-n}$. (16)
    - Si $b = 1$, entonces $R = A^{-l} * Q * A^{-n}$. (17)

A continuación, se pasa a la **verificación** (Lado del Servidor – Bob):

13. Bob recibe la respuesta de Alice y verifica la consistencia de ésta a través de la veracidad de una de estas expresiones:
    - Si $b = 0$, entonces controla si $S^m * R * S^n * G_a = Q * G_b$. (18)
    - Si $b = 1$, entonces controla si $B^{-m} * R * B^{-n} = GB * G_b$. (19)
14. Si se verifica la veracidad de la respuesta recibida, Bob acepta la identidad del usuario en esa iteración. En ese caso, Bob decide si quedó satisfecho o no, chequeando la cantidad *r* de iteraciones. Si *r*> 0, Bob dispone iterar, *r* veces, los pasos desde el **testigo** hasta la **verificación**. Si r = 0, Bob acepta la identidad del usuario, lo notifica a Alice y termina el protocolo. En caso de una respuesta inválida, Bob rechaza la identidad del usuario y lo notifica a Alice, quien debe comenzar el proceso desde el inicio, si el usuario aún desea autenticarse.

3.2.3. - REDUCCIÓN DE LA COMPLEJIDAD COMPUTACIONAL

La introducción de matrices elementales, en la fase de preparación de esta variante, reduce el costo computacional del protocolo ZKP original. Mientras la propuesta original [11] presenta una complejidad de orden $O(n^3)$ u O(n!), debido a la generación de matrices, la inversión, el producto y la potenciación matriciales, esta variante reduce el esfuerzo al orden $O(n^2)$, ya que se aprovecha la estructura de dichas matrices en la implementación de muchas operaciones en él involucradas.

3.2.4. - ANÁLISIS DE EFECTIVIDAD CONTRA ATAQUES BASADOS EN EL TRÁFICO DE RED

En el análisis de seguridad desarrollado en [11], se define que "un atacante tiene una probabilidad $p = ½$ de ser aceptado en cada intercambio pero, al cabo de *r* intercambios la probabilidad total de éxito se vuelve $p = (1/2)^r$". En ese caso el autor definió $p = ½$ como probabilidad, ya que, para $b = 0$, no se puede garantizar la legitimidad del usuario. Por tanto, la única garantía de éxito en la autenticación es para el camino $b = 1$. Con las modificaciones propuestas para la versión original, las probabilidades mencionadas cambian, ya que se garantiza la legitimidad del usuario tanto para $b = 0$ como para $b = 1$, aunque este último continúa proveyendo un mayor nivel de seguridad ante ataques. A continuación, se explica el por qué de ese razonamiento. Según [26], la probabilidad de que ocurra un evento dada la probabilidad de ocurrencia de otros que son mutuamente excluyentes entre sí se define como:

$$P(A \cup B) = P(A) + P(B) \quad (20)$$

Para ello, se definen como eventos:

- A: Bob acepta a un Alice ilegítimo para b = 0.
- B: Bob acepta a un Alice ilegítimo para b = 1.

Para buscar $P(A)$ se utiliza la ecuación puntualizada en [26] que tiene en cuenta los casos favorables (CF) y posibles (CP):

$$P(A) = CF/CP \quad (21)$$

Para que se cumpla el evento A, el atacante debe generar una matriz *Ga* que coincida con la matriz G_b, que contiene el servidor. Estas matrices son de orden 8 (64 elementos) y módulo 251. Por lo tanto, existe solo 1 caso favorable y 25064 casos posibles, lo cual se traduce en $P(A) = 1/250^{64}$.

Para que se cumpla el evento B, el atacante debe generar una matriz que represente a la clave privada *A*. Esta, como se enunció antes, se calcula con información pública y compartida que puede ser vulnerada y con una matriz diagonal *Da* de 8 elementos (diagonales), la cual sí es información secreta y solo conocida por Alice. Además, el posible atacante debe generar otra matriz que represente a G_a, que debe coincidir con la matriz G_b, que contiene el servidor. Esta matriz es de orden 8 (64 elementos) y módulo 251. Por lo tanto, la probabilidad de que Bob acepte a Alice en una iteración con b = 1 está dada por la probabilidad de que genere una matriz *A* y una matriz G_a correctas y se define como [26]:

$$P(B) = P(C \cap D) = P(C) * P(D), \quad (22)$$

siendo C el evento de generar clave privada válida y D, el evento de generar G_a válida. Sustituyendo los valores, se obtiene:

$$P(C \cap D) = (CF \text{ de } C / CP \text{ de } C) * (CF \text{ de } D / CP \text{ de } D) = 1/2^{64} * 1/250^{64} = 1/(125^{64} * 2^{128}) \quad (11)$$

$$P(A \cup B) = P(A) + P(B) = 1/250^{64} + 1/(125^{64} * 2^{128}) \quad (12)$$

Como se puede observar, la probabilidad de éxito con una sola iteración es ínfima, incluso menor que el 0.00001 %, por lo tanto, teóricamente, la variante propuesta presenta un mayor nivel de seguridad con respecto a la variante original. Esta premisa propicia, además, que se requiera una menor cantidad de intentos (con respecto a la versión anterior) para garantizar el nivel de seguridad requerido. Por lo tanto, teóricamente, también debe disminuir el gasto de recursos computacionales y el tiempo de ejecución del protocolo con la variante propuesta.

3.2.5. - DESPLIEGUE DEL SISTEMA DE CONTROL DE ACCESO

Para evaluar la aplicabilidad del protocolo propuesto en un escenario real de IoT, se ha desarrollado un sistema de control de acceso basado en el escaneo de códigos QR. Este sistema simula un entorno IoT básico y permite la integración del protocolo ZKP para la autenticación de usuarios. El sistema consta de un servidor de base de datos, un servidor web y una aplicación móvil. El servidor de base de datos almacena la información persistente del sistema, mientras que el servidor web maneja las funcionalidades del sistema y proporciona una interfaz web para su administración. La aplicación móvil permite a los usuarios escanear códigos QR para solicitar acceso. La Fig. 2 muestra el diagrama de despliegue del sistema.

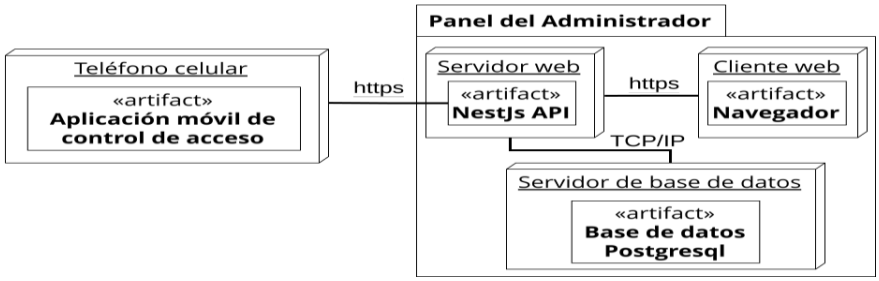

Figura 2

Diagrama de despliegue del sistema de control de acceso.

El despliegue del cliente web, del servidor web y del servidor de base de datos se realiza utilizando el proveedor de *hosting* gratuito render.com, que permite el despliegue de aplicaciones web estáticas, servicios web y bases de datos PostgreSQL, cumpliendo con los requerimientos del sistema.

4. - VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

En esta sección se presenta la validación experimental de la solución propuesta, con el objetivo de evaluar su eficiencia y seguridad en un entorno práctico. La validación se centra en el análisis del protocolo de autenticación ZKP implementado, tanto en su versión web como en la aplicación móvil, considerando aspectos como el tiempo de ejecución y la resistencia a ataques. Además, se realiza un diseño de experimentos para determinar la configuración óptima de los parámetros del protocolo que minimiza el tiempo de ejecución sin comprometer la seguridad. Finalmente, se verifica la propiedad de conocimiento cero del protocolo, demostrando que la información intercambiada durante el proceso de autenticación no permite la recuperación de la clave secreta del usuario. Los resultados obtenidos en esta sección son cruciales para demostrar la viabilidad y robustez de la solución propuesta en un contexto real de aplicación.

4.1. - ANÁLISIS DE EFICIENCIA DEL PROTOCOLO IMPLEMENTADO

La eficiencia de un algoritmo está intrínsecamente ligada a la cantidad de recursos que consume para resolver un problema específico; a menor cantidad de recursos empleados, mayor es la eficiencia. Se considera que un algoritmo es eficiente si su consumo de recursos se encuentra dentro de la media o por debajo de los niveles considerados aceptables para el contexto en el que se aplica. La eficiencia se puede medir de diversas maneras, siendo la complejidad temporal y espacial las dos métricas más comúnmente utilizadas. Este estudio se enfoca en la complejidad temporal, o sea, en el tiempo de ejecución del protocolo de autenticación ZKP implementado.

Para las mediciones del tiempo de ejecución, se emplearon dos unidades experimentales: un ordenador portátil Lenovo Ideapad 330-15AST y un dispositivo móvil Samsung A03s. El ordenador portátil cuenta con 8 GB de memoria RAM, un procesador AMD A9-9425 de 2 núcleos a 3.10 GHz, una velocidad promedio de subida de 9.39 Mbps y una velocidad promedio de descarga de 39.96 Mbps. El dispositivo móvil posee 4 GB de memoria RAM, un procesador Mediatek Helio P35 de 8 núcleos a 2.3 GHz, una velocidad promedio de subida de 18.75 Mbps y una velocidad promedio de descarga de 37.7

Mbps. El ordenador portátil se utilizó para medir el tiempo de autenticación en la plataforma del administrador, mientras que el dispositivo móvil se empleó para medir el mismo proceso en la aplicación móvil de control de acceso por QR desarrollada.

Es importante destacar que el tiempo de ejecución (TE) de un algoritmo puede verse afectado por diversos factores externos, entre los que se incluyen: las tareas del sistema operativo que se ejecutan en segundo plano; la velocidad del procesador, el número de núcleos y el conjunto de instrucciones que puede ejecutar; la cantidad de memoria RAM y caché disponibles, así como la velocidad de acceso a ellas. Adicionalmente, en el caso de algoritmos que implican intercambio de información a través de Internet o servicios web, como es el caso del protocolo ZKP implementado, también pueden influir factores como la calidad de la conexión a Internet y el rendimiento del servidor de "hosting" donde se encuentra desplegada la solución.

Para controlar la posible influencia de estos factores externos, se realizaron cinco ejecuciones del protocolo para cada valor de *r* (número de intentos o rondas del protocolo). En [11] se sugirió un valor de $r = 10$. Sin embargo, en este estudio se realizaron mediciones con valores de *r* en el conjunto {2, 4, 6, 8, 10}, con el fin de evaluar el impacto de este parámetro en el tiempo de ejecución. Además, se registraron los valores de los enteros m y n, ya que estos parámetros podrían influir en el tiempo de ejecución, debido a que se utilizan como exponentes en potenciación matricial.

4.1.1. - MEDICIONES DE TIEMPO DE EJECUCIÓN EN LA WEB

Como parte de la validación, se midió el tiempo de ejecución del protocolo ZKP implementado en el panel del administrador (cliente: navegador web). Los resultados obtenidos para las diferentes cantidades de intentos (*r*), junto con los valores de *m* y *n* utilizados, se presentan en la Tabla 1.

Tabla 1.

Tiempos de Ejecución para Distintas Cantidades de Intentos en la Web.

Cantidad de Intentos															
2			4			6			8			10			
m	n	TE (seg)	m	n	TE (seg)	m	n	TE (seg)	m	n	TE (seg)	m	n	TE (seg)	
2	8	2.947	27	5	3.654	23	17	2.551	28	14	5.018	14	11	3.592	
10	12	2.089	3	27	2.643	13	5	3.767	13	10	2.098	24	3	4.13	
23	2	1.666	23	8	1.356	12	13	2.274	24	15	4.403	2	20	3.001	
15	14	1.647	23	25	2.007	10	7	3.495	7	20	3.171	11	15	2.394	
2	3	1.192	3	27	2.777	4	12	2.15	29	19	4.091	27	4	3.254	

Como se puede observar en la Tabla 1, el tiempo de ejecución del protocolo, en la plataforma del administrador, varía entre aproximadamente 1 y 5 segundos. Sin embargo, no se puede determinar a priori qué factores influyen de manera determinante en dichos tiempos de ejecución.

4.1.2. - MEDICIONES DE TIEMPO DE EJECUCIÓN EN LA APK

Se realizaron mediciones del tiempo de ejecución del protocolo ZKP implementado en la aplicación móvil de control de acceso por QR. Los resultados obtenidos para las diferentes cantidades de intentos (*r*), junto con los valores de *m* y *n* utilizados, se presentan en la Tabla 2.

Tabla 2

Tiempos de Ejecución para Distintas Cantidades de Intentos en la apk.

Cantidad de Intentos															
2			4			6			8			10			
m	n	TE (seg)	m	n	TE (seg)	m	n	TE (seg)	m	n	TE (seg)	m	n	TE (seg)	
15	16	3.169	13	5	6.209	6	13	6.135	17	7	10.631	27	3	9.099	
4	10	2.724	26	31	6.862	17	24	6.285	25	18	16.051	17	7	6.265	
11	4	2.1	30	22	4.25	28	5	4.74	18	3	11.781	29	22	6.347	
17	28	1.622	2	28	4.261	3	20	4.713	27	25	12.271	29	30	6.708	
22	21	1.698	29	17	4.142	18	5	5.431	29	25	9.766	21	28	6.636	

Como se puede observar en la Tabla 2, el tiempo de ejecución en la aplicación móvil varía entre aproximadamente 4 y 16 segundos. Al igual que en el caso de la plataforma del administrador, no es posible determinar a priori los factores que influyen de manera determinante en estos tiempos de ejecución.

4.1.3. - ANÁLISIS DE LOS RESULTADOS

Las mediciones de los tiempos de ejecución de la variante del protocolo ZKP implementada revelan que, a pesar de que en [11] se infiere que el protocolo "...funcionaría en tiempos no limitantes en ambientes de baja capacidad computacional... como los teléfonos celulares...", esta afirmación no se cumple completamente en la práctica, al menos en el contexto de una aplicación móvil. Si bien la variante implementada en este trabajo incorpora optimizaciones en varias operaciones críticas para el funcionamiento del protocolo, el tiempo de ejecución en la aplicación móvil, tanto para la autenticación de usuarios como de dispositivos, oscila entre 4 y 16 segundos.

Este retardo en la autenticación de dispositivos puede ser tolerable para el usuario, ya que es un proceso que se realiza en segundo plano y no afecta directamente su interacción con la aplicación. Sin embargo, en el caso de la autenticación de usuarios, un tiempo de ejecución de esta magnitud sí impacta negativamente en la experiencia del usuario, pudiendo llevar a la decisión de no utilizar este protocolo ZKP en aplicaciones móviles, a menos que la demora en el proceso de autenticación no sea un factor crítico para la funcionalidad de la aplicación en cuestión.

4.2. - DISEÑO DE EXPERIMENTOS PARA OPTIMIZAR EL PROTOCOLO

El diseño de experimentos (DoE, por sus siglas en inglés) es una herramienta estadística que permite investigar simultáneamente los efectos de múltiples variables de entrada (factores) sobre una variable de salida (respuesta). Estos experimentos consisten en una serie de corridas o pruebas en las que se realizan cambios intencionales en los valores de los factores. En cada corrida se recolectan datos sobre la variable de respuesta y, posteriormente, se analizan para identificar las condiciones del proceso y los valores de los factores que optimizan los resultados [27].

4.2.1. - METODOLOGÍA DEL DISEÑO DE EXPERIMENTOS

Con el objetivo de identificar los factores que ejercen influencia significativa en el tiempo de ejecución del protocolo ZKP propuesto, se llevó a cabo un diseño experimental. Se tomó, como variable de respuesta, el tiempo de ejecución del protocolo y se analizaron los siguientes factores: *r*, que representa la cantidad de intentos o rondas para validar la identidad del probador; así como *m* y *n*, que son enteros utilizados como exponentes en el cálculo de matrices durante la ejecución del protocolo. Estos valores deben ser diferentes y encontrarse en el rango de 2 a 32, según [11].

Se definieron dos niveles para cada factor: para *r*, un nivel bajo de 3 y un nivel alto de 8; para *m* y *n*, un nivel bajo de 8 y un nivel alto de 24. Se optó por un diseño factorial completo para evaluar todas las posibles combinaciones de los niveles de los factores. Dado que se tienen tres factores, con dos niveles cada uno, inicialmente se generaron $2^3 = 8$ tratamientos. Sin embargo, se eliminaron los tratamientos en los que *m* coincide con *n*, ya que el protocolo especifica que estos valores deben ser diferentes. Por lo tanto, el número final de tratamientos fue de 6. Para cada tratamiento se realizaron tres réplicas, obteniendo un total de 18 ejecuciones. Por otro lado, el orden de los tratamientos se toma al azar para garantizar la independencia de las observaciones.

4.2.2. - RESULTADOS DEL DISEÑO DE EXPERIMENTOS

Los resultados del diseño de experimentos se presentan en la Fig. 3. Las columnas etiquetadas como "r", "m" y "n" representan los factores con sus respectivos niveles en cada corrida, mientras que la columna "Tiempo de Ejecución (s)" muestra el tiempo de ejecución obtenido en cada experimento.

Como se puede observar en la Fig. 3, los tiempos de ejecución varían entre aproximadamente 1 y 3 segundos para las diferentes corridas. Para determinar qué factores ejercen influencia significativa en estos resultados, se construyó un diagrama de Pareto (Fig. 4), que muestra los efectos de los factores y sus interacciones en la variable de respuesta [28].

	OrdenEst	OrdenCorrida	PtCentral	Bloques	r	m	n	Tiempo de ejecución(seg)
1	8	2	1	1	8	24	8	1.765
2	24	3	1	1	8	24	8	2.586
3	23	4	1	1	3	24	8	1.383
4	20	5	1	1	8	24	8	2.176
5	3	6	1	1	3	24	8	2.066
6	5	9	1	1	3	8	24	1.115
7	11	11	1	1	3	24	8	1.388
8	19	16	1	1	3	24	8	1.384
9	4	17	1	1	8	24	8	2.527
10	22	18	1	1	8	8	24	3.016
11	12	19	1	1	8	24	8	2.679
12	14	20	1	1	8	8	24	3.177
13	13	21	1	1	3	8	24	1.450
14	21	23	1	1	3	8	24	2.416
15	6	24	1	1	8	8	24	2.622

Figura 3

Resultados del diseño de experimentos.

El diagrama de Pareto (Fig. 4) revela que el factor *r* (cantidad de intentos o rondas del protocolo) es el único que tiene un efecto estadísticamente significativo sobre el tiempo de ejecución. Los demás factores (*m*, *n* y sus interacciones) no tienen un impacto considerable en la variable de respuesta.

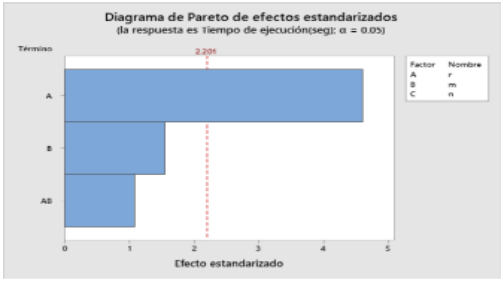

Figura 4

Diagrama de Pareto para los factores que influyen en el tiempo de ejecución.

4.3. - ANÁLISIS DE LA PROPIEDAD DE CONOCIMIENTO CERO

El análisis de varianza realizado mediante el DoE arrojó, como resultado, que el factor más influyente en el tiempo de ejecución del protocolo es la cantidad de intentos (*r*) y que, para alcanzar la configuración óptima que minimiza el tiempo de ejecución, se debe establecer *r* en su valor bajo (*r* = 3). Una vez establecida la configuración óptima, se procede a realizar las pruebas para validar la propiedad de conocimiento cero (ZKP) de la variante aquí implementada. Hecht ofrece un simulador que permite demostrar que la prueba de identidad, en [11], satisface dicha propiedad, y este simulador encaja en la variante presentada en este trabajo. El objetivo de estas pruebas consiste en demostrar que, a partir de la información transmitida a través de la red durante la ejecución del protocolo, no es posible obtener ninguna información sobre la clave secreta del usuario.

Para llevar a cabo estas pruebas, se utilizaron, como datos de entrada, el nombre de usuario "ara" y la contraseña "ara0005". Como se explicó en los pasos 2. y 7. del protocolo (sección 3.2.2), tanto Bob como Alice emplean un mecanismo de cifrado para manipular la contraseña de Alice. Específicamente, se utiliza el algoritmo SHA256, que genera una cadena hexadecimal de 64 caracteres. Posteriormente, esta cadena se convierte, de alguna manera, en una matriz elemental invertible de orden 8, donde cada carácter se representa por su valor numérico en código ASCII. Para demostrar que la variante implementada cumple con la propiedad ZKP, se debe probar que no es posible obtener la matriz anterior (que representa la contraseña cifrada) a partir de la información que se intercambia a través de la red durante la ejecución del protocolo.

Ahora, se compara la matriz obtenida a partir de la contraseña del usuario (información secreta) con la matriz *GA* (información pública), que se calcula a partir de la matriz secreta y se envía al verificador (Bob) durante el protocolo. En esta comparación no existió relación evidente entre la matriz secreta y la matriz pública *GA*. Como se mencionó anteriormente, la seguridad del protocolo se basa en el empleo de GSDP, que es un problema de complejidad NP. Por otro lado, a partir de las matrices *S* y *R* tampoco se puede obtener dicha clave secreta. Esto significa que, en la práctica, el único método conocido para obtener la matriz secreta, a partir de la información pública (*GA*, *m*, *n*, *S* y *R*) es mediante un ataque de fuerza bruta, que implica la exploración sistemática de un espacio de búsqueda exponencialmente grande. Por lo tanto, se puede concluir que la variante

implementada del protocolo ZKP cumple con la propiedad de conocimiento cero, ya que no es posible obtener información sobre la clave secreta a partir de la información que se transmite a través de la red.

5.- CONCLUSIONES

Este trabajo aborda la acuciante necesidad de seguridad en la capa de percepción de la Internet de las Cosas, proponiendo una variante optimizada del protocolo de autenticación basado en pruebas de conocimiento cero de Hecht [11], que se destaca por su eficiencia y robustez. Las contribuciones fundamentales incluyen la reducción significativa de la complejidad computacional mediante el uso de matrices elementales, la incorporación de un mecanismo explícito para la gestión segura del secreto, garantizando la propiedad de conocimiento cero, la optimización en la generación de matrices invertibles, y la validación experimental en un entorno IoT simulado mediante un sistema de control de acceso por códigos QR. Los resultados de la validación demuestran que la variante propuesta es viable para dispositivos de bajas prestaciones, con tiempos de ejecución considerablemente menores, en comparación con el protocolo original, especialmente en entornos web. Además, se identificaron, como amenazas de mayor riesgo, el ataque de reproducción, la negación de servicio y la captura de tráfico de red en la capa de percepción, amenazas que la solución propuesta mitiga eficazmente. Se abren, además, importantes líneas de investigación futura, como el fortalecimiento del cifrado, la optimización de operaciones con matrices, el análisis de la periodicidad de estas, la validación de dispositivos utilizados en la Internet de las Cosas en una gama más amplia, la incorporación de autenticación multifactor, un análisis formal de seguridad, la investigación de la resistencia a ataques cuánticos y la evaluación de vulnerabilidades a ataques de canal lateral (consumo energético, tiempo de ejecución y emisiones electromagnéticas), dado su impacto en implementaciones criptográficas reales. Esta investigación proporciona una solución eficiente y segura para la autenticación en la capa de percepción de la IoT, superando las limitaciones de propuestas anteriores y sentando las bases para futuras investigaciones que consoliden la seguridad en este ámbito crítico.

REFERENCIAS

1. Ahmid M., Kazar O., Barka E. Internet of Things Overview: Architecture, Technologies, Application, and Challenges. In: Boulila W, Ahmad J, Koubaa A, Driss M, Farah IR, editors. Decision Making and Security Risk Management for IoT Environments. Advances in Information Security; 2024. 106 p. 1-19. Springer, Cham. doi: 10.1007/978-3-031-47590-0_1.
2. Vishwakarma A.K., Chaurasia S., Kumar K., Singh Y.N., Chaurasia R. Internet of things technology, research, and challenges: a survey. Multimed Tools Appl.; 2024. 2:1-36. doi: 10.1007/s11042-024-19278-6.
3. Karunarathne S.M., Saxena N., Khan M.K. Security and privacy in IoT smart healthcare. IEEE Internet Computing; 2021. 25(4): 37-48.
4. Kumar V., Sharma K.V., Kedam N., Patel A., Kate T.R., Rathnayake U. A comprehensive review on smart and sustainable agriculture using IoT technologies. Smart Agricultural Technology; 2024. 8: 100487. doi: 10.1016/j.atech.2024.100487.
5. Maghfiroh H., Slamet Saputro J., Shanaza Andiany B., Hermanu C., Anwar M., Nizam M., et al. Smart Home System With Battery Backup and Internet of Things. Journal of Applied Engineering and Technological Science; 2023. 5(1): 42-57. doi: 10.37385/jaets.v5i1.1969.
6. Statista. Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2021, with forecasts from 2022 to 2030. [2022 Aug 06]; Disponible en: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/.
7. Tewari A., Gupta B.B. Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework. Future generation computer systems; 2020. 108: 909-920. doi: 10.1016/j.future.2018.04.027.
8. Pawlicki M., Pawlicka A., Kozik R., Choraś M. The survey and meta-analysis of the attacks, transgressions, countermeasures and security aspects common to the Cloud, Edge and IoT. Neurocomputing; 2023. 6:126533. doi: 10.1016/j.neucom.2023.126533.
9. Rao P.M., Deebak B.D. A comprehensive survey on authentication and secure key management in internet of things: Challenges, countermeasures, and future directions. Ad Hoc Networks; 2023. 146:103159. doi: 10.1016/j.adhoc.2023.103159.
10. Feige U., Fiat A., Shamir A. Zero-knowledge proofs of identity. Journal of cryptology; 1988. 1(2): 77-94.
11. Hecht P. A Zero-Knowledge authentication protocol using non commutative groups. Actas del VI Congreso Iberoamericano de Seguridad Informática CIBSI; 2011.

12. Zivkovic C., Guan Y., Grimm C. IoT Platforms, Use Cases, Privacy, and Business Models. Springer, Cham; 2021.
13. Chaurasia N., Kumar P. A comprehensive study on issues and challenges related to privacy and security in IoT. e-Prime- Advances in Electrical Engineering, Electronics and Energy; 2023. 4: 100158. doi: 10.1016/j.prime.2023.100158.
14. Microsoft. The STRIDE Threat Model; 2009. [2022 Oct 06]; Disponible en: https://learn.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)?redirectedfrom=MSDN.
15. Sadhu P.K., Yanambaka V.P., Abdelgawad A. Internet of things: Security and solutions survey. Sensors; 2022. 22(19):7433. doi: 10.3390/s22197433.
16. Menezes A.J., Van Oorschot P.C., Vanstone S.A. Handbook of applied cryptography. CRC press; 1997.
17. Rubinstein-Salzedo S. Cryptography. Cham, Switzerland: Springer; 2018.
18. Microsoft. Security Development Lifecycle (SDL); 2022. [2022];
19. Disponible en: https://learn.microsoft.com/en-us/compliance/assurance/assurance-microsoft-security-development-lifecycle.
20. Bansal S., Kumar D. IoT ecosystem: A survey on devices, gateways, operating systems, middleware and communication. International Journal of Wireless Information Networks; 2020. 27(3):340-64. doi: 10.1007/s10776-020-00483-7.
21. Quinn S., Quinn S., Ivy N., Barrett M., Feldman L., Witte G., et al. Identifying and estimating cybersecurity risk for enterprise risk management. US Department of Commerce, National Institute of Standards and Technology; 2021.
22. Satanasaowapak P., Kawseewai W., Promlee S., Vilamat A. Residential access control system using QR code and the IoT. International Journal of Electrical and Computer Engineering (IJECE); 2021. 11(4):3267-3274.
23. Fauzi A.F.M., Mohamed N.N., Hashim H., Saleh M.A. Development of Web-Based Smart Security Door Using QR Code System. IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS). Shah Alam, Malaysia; 2020. p. 13-17. doi: 10.1109/I2CACIS49202.2020.9140200.
24. Meyer C.D. Matrix analysis and applied linear algebra. Society for Industrial and Applied Mathematics; 2023.
25. Golub G.H., Van Loan C.F. Matrix Computations. The John Hopkins University Press. Baltimore; 1996.
26. Sun X. Aggregations of Elementary Transformations. Technical Report DUKE-TR-1996-03. Duke University; 1996.
27. Walpole R., Myers R., Myers S. Probabilidad y estadística para ingeniería y ciencias. Pearson educación; 2012. 162: p. 157.
28. Antony J. Design of experiments for engineers and scientists. Elsevier; 2023.

CONFLICTO DE INTERESES

Ninguno de los autores manifestó la existencia de posibles conflictos de intereses que debieran ser declarados en relación con este artículo.

CONTRIBUCIONES DE LOS AUTORES

Ernesto Rafael Carbonell-Rigores, Conceptualización, Curación de datos, Análisis formal, Validación, Diseño experimental, Investigación, Metodología, Recursos, Software, Supervisión, Validación, Visualización, Redacción-borrador original, Redacción-revisión y edición.
Aramays Aimet Morales-Duran, Conceptualización, Curación de datos, Validación, Diseño experimental, Investigación, Recursos, Software, Validación, Visualización y Redacción-borrador original.
Roberto Sepúlveda-Lima, Conceptualización, Curación de datos, Análisis formal, Investigación, Metodología, Recursos, Supervisión, Validación, Visualización, Redacción-borrador original, Redacción-revisión y edición
Wenny Hojas-Mazo, Conceptualización, Metodología, Visualización, Redacción-borrador original, Redacción-revisión, edición.

AUTORES

Ernesto Rafael Carbonell-Rigores, https://orcid.org/0000-0001-5722-103X, Graduado de Licenciatura en Matemáticas (Universidad Técnica de Dresde, 1988). Es profesor Asistente de la Facultad de Ingeniería Informática en la Universidad Tecnológica de La Habana José Antonio Echeverría, Cujae. Intereses de investigación centrados en álgebra computacional, prueba de conocimiento cero, criptografía poscuántica. Email: ernesto@ceis.cujae.edu.cu.

Aramays Aimet Morales-Duran, https://orcid.org/0009-0004-4497-0962, Graduada de Ingeniería Informática (Cujae, 2022). Es trabajadora de Guajiritos SRL. Sus intereses de investigación se centran en criptografía, soluciones informáticas que optimicen o mejoren procesos en sectores fundamentales de la sociedad como medicina, educación y otros. Email: aramaysm@gmail.com.

Roberto Sepúlveda-Lima, https://orcid.org/0000-0002-9451-6395, Graduado de Ingeniero (Cujae, 1981) y Doctor en Ciencias Técnicas (Cujae, 1998). Profesor Titular de la Universidad Tecnológica de La Habana José Antonio Echeverría. Cujae. Sus intereses de investigación se centran en los sistemas bioinspirados, la ciberseguridad, la ingeniería de software y el diseño de hardware para ser aplicado como herramientas combinadas. Email: rsepulvedalima@gmail.com.

Wenny Hojas-Mazo, https://orcid.org/0000-0002-8298-3439, Graduado de Ingeniería Informática (Cujae, 2012), Máster en Informática Aplicada (Cujae, 2017) y Doctor en Ciencias Técnicas (Cujae-Universidad de Alicante, 2024). Es profesor Asistente de la Facultad de Ingeniería Informática en la Universidad Tecnológica de La Habana José Antonio Echeverría, Cujae. Sus intereses de investigación se centran en el procesamiento del lenguaje natural, la inteligencia artificial, aprendizaje automático, ingeniería de software y sistemas distribuidos. Email: whojas@ceis.cujae.edu.cu.