An Optimized Hybrid Encryption Framework for Smart Home Healthcare: Ensuring Data Confidentiality and Security

POPOOLA, Olusogo, RODRIGUES, Marcos <http://orcid.org/0000-0002-6083-1303>, MARCHANG, Jims <http://orcid.org/0000-0002-3700-6671>, SHENFIELD, Alex <http://orcid.org/0000-0002-2931-8077>, IKPEHAI, Augustine and POPOOLA, Jumoke

Published version

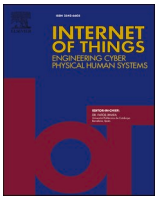
POPOOLA, Olusogo, RODRIGUES, Marcos, MARCHANG, Jims, SHENFIELD, Alex, IKPEHAI, Augustine and POPOOLA, Jumoke (2024). An Optimized Hybrid Encryption Framework for Smart Home Healthcare: Ensuring Data Confidentiality and Security. Internet of Things; Engineering Cyber Physical Human Systems, 27: 101314.

Internet of Things 27 (2024) 101314

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.sciencedirect.com/journal/internet-of-things



An optimized hybrid encryption framework for smart home healthcare: Ensuring data confidentiality and security

Olusogo Popoola [a,*], Marcos A Rodrigues [b], Jims Marchang [a], Alex Shenfield [b], Augustine Ikpehai [b], Jumoke Popoola [a]

[a] Department of Computing, Sheffield Hallam University, UK
[b] Department of Engineering and Math, Sheffield Hallam University, UK

ARTICLE INFO

Keywords:
Hybrid encryption
Smart home healthcare
Quantum resistance
Post-quantum cryptography (PQC)
Elliptic curve cryptography (ECC)
Internet of Things (IoT) security

ABSTRACT

This study proposes an optimized hybrid encryption framework combining ECC-256r1 with AES-128 in EAX mode, tailored for smart home healthcare environments, and conducts a comprehensive investigation to validate its performance. Our framework addresses current limitations in securing sensitive health data and demonstrates resilience against emerging quantum computing threats. Through rigorous experimental evaluation, we show that the proposed configuration outperforms existing solutions by delivering unmatched security, processing speed, and energy efficiency. It employs a robust yet streamlined approach, meticulously designed to ensure simplicity and practicality, facilitating seamless integration into existing systems without imposing undue complexity. Our investigation affirms the framework's capability to resist common cybersecurity threats like MITM, replay, and Sybil attacks while proactively considering quantum resilience. The proposed method excels in processing speed (0.006 seconds for client and server) and energy efficiency (3.65W client, 95.4W server), offering a quantum-resistant security level comparable to AES-128. This represents a security-efficiency ratio of 21.33 bits per millisecond, a 25.6% improvement in client-side processing speed, and up to 44% reduction in server-side energy consumption compared to conventional RSA-2048 methods. These improvements enable real-time encryption of continuous health data streams in IoT environments, making it ideal for IoT devices where AES-128's smaller footprint is advantageous. By prioritizing high-grade encryption alongside ease of use and implementation, the proposed framework presents a future-proof solution that anticipates the trajectory of cryptographic standards amid advancing quantum computing technologies, signifying a pivotal advancement in safeguarding IoT-driven healthcare data.

1. Introduction

The advent of smart home healthcare, underpinned by the Internet of Things (IoT), marks a transformative phase in patient care, enabling remote monitoring and management of health data. The proliferation of IoT devices and remote patient monitoring platforms necessitates robust security to safeguard confidential patient data. Connected sensors and wearables enable continuous health supervision but introduce privacy risks from intercepted data flows or compromised devices [1]. This technological evolution brings

* Corresponding author.
E-mail address: o.popoola@shu.ac.uk (O. Popoola).

https://doi.org/10.1016/j.iot.2024.101314

Available online 3 August 2024

significant challenges in securing sensitive patient data against increasing threats, particularly from the growing capability of quantum computing. The rapid advancement of quantum computing technologies poses a significant threat to traditional cryptographic systems, particularly those relying on the hardness of integer factorization and discrete logarithm problems, such as RSA (Rivest-Shamir--Adleman) and ECC (Elliptic Curve Cryptography) that secure most communication channels today. As quantum computers become more powerful, they could potentially break these widely-used encryption methods, jeopardizing the security of sensitive data across various domains, including healthcare. To address these challenges, we explore advanced cryptographic techniques including ECC, RSA, and AES.

In response to this looming threat, researchers have proposed several post-quantum cryptography (PQC) schemes designed to withstand attacks from both classical and quantum computers. These include lattice-based key encapsulation mechanisms like NewHope, which offers efficiency and quantum resistance based on the 'ring learning with errors' problem, making it a suitable candidate for secure IoT communications and device key exchange [2,3]. The Isogeny-based cryptographic protocols such as Supersingular Isogeny Diffie-Hellman (SIDH) leverage the hardness of finding isogenies between supersingular elliptic curves to provide compact key sizes and strong security assumptions [4-6]. Code-based cryptographic schemes like McEliece, known for their long-term security against quantum attacks due to the difficulty of decoding random linear codes, have also gained attention despite requiring larger key sizes [7,8]. Additionally, multivariate quadratic (MQ) cryptography, which uses quadratic polynomials over finite fields, shows promise in resisting both classical and quantum attacks, making them a strong candidate for future-proof encryption solutions. [9,10]. While these PQC schemes offer potential solutions for future cryptographic needs, their practical implementation, especially in resource-constrained environments like Internet of Things (IoT) devices used in smart home healthcare environments (SHHE), presents significant challenges that need to be addressed. Thus, the urgent adoption of post-quantum cryptography (PQC) techniques robust against quantum attacks is vital. This motivates the analysis and benchmarking of promising PQC schemes like lattice-based NewHope, isogeny-based SIDH, code-based McEliece, and multivariate-quadratic QR alongside ECC and AES [11,12].

Lightweight cryptography emerged as an essential solution, ensuring the confidentiality, integrity, and availability of data amidst the peer-to-peer distribution in smart ecosystems. With the average smart home predicted to house over 50 internet-connected devices by 2025 [13] the urgency for robust, efficient hybrid encryption schemes that leverage RSA/AES, ECC/AES across various symmetric encryption modes, and perhaps post-quantum cryptography—becomes paramount, as these cryptosystems provide promising optimal approach to balance between security and operational efficiency. Despite their promises, a comprehensive exploration of the most effective configurations and encryption modes tailored for healthcare applications remains scant. This gap motivates a focused analysis of hybrid RSA/AES and ECC/AES encryption schemes, aiming to pinpoint configurations that best align with healthcare security requirements.

Resource-limited IoT devices ubiquitous in smart environments necessitate secure yet efficient cryptographic solutions tailored to their constraints. The core challenge involves striking a balance - deploying encryption strong enough to adequately safeguard sensitive patient health data while remaining lightweight to respect the processing capacity, battery life, and memory restrictions endemic in networked healthcare sensors and wearables. The key to securing smart healthcare ecosystems hinges on identifying and leveraging optimized encryption techniques providing sufficient cryptographic strength for privacy protections while minimizing associated overheads to align with the inherent limitations of IoT endpoints [14] As IoT proliferates in scale and scope across modern healthcare infrastructures, the imperative for lightweight confidentiality protocols offering robust defenses despite their leanness constitutes the crux. The analysis and comparative assessment of integrated encryption models balancing security protections and efficiency form a pivotal step towards this goal.

The primary contribution of this paper is the proposal of a tailored hybrid encryption framework specifically optimized for smart home healthcare environments. We adapt and refine existing cryptographic techniques, combining ECC-256r1 for asymmetric encryption with AES-128 in EAX mode for symmetric encryption, to address the unique constraints and security requirements of IoT-driven healthcare systems. We then rigorously investigate the performance of this adapted framework to demonstrate its effectiveness and efficiency in this specific context.

This study conducts an in-depth investigation into the aforementioned hybrid encryption models, rigorously evaluating their performance in terms of processing time, latency, bandwidth usage, and energy consumption. By undertaking comprehensive performance analyses of hybrid RSA/AES and ECC/AES transport encryption using experimental encryption modes, this research provides empirical evidence supporting their relevance and efficiency in ensuring secure data transmission. The findings demonstrate that ECC/AES, in particular, is highly effective for IoT-driven SHHE, balancing security, performance, and adaptability while mitigating overheads in encryption processes. The comparative assessment of these hybrid transport encryption schemes offers a detailed benchmark, guiding the selection of the most efficient cryptographic solutions for stakeholders within SHHE. Additionally, the study meticulously evaluates the cryptographic overheads across various data exchange scenarios, using IoT client-to-cloud server communication as a representative case study. This robust investigative approach confirms the practical applicability of ECC/AES in delivering remote healthcare securely and efficiently, thereby ensuring user privacy and addressing the multidimensional resource constraints inherent in IoT environments.

Furthermore, it demonstrates the feasibility and effectiveness of lightweight yet robust encryption models that significantly reduce overheads while ensuring efficient and secure health data transmission, addressing the unique challenges posed by resource-constrained IoT devices in SHHE. The study explores the use of advanced data protection mechanisms to significantly enhance privacy preservation, addressing the dire need for an optimized encryption framework that caters to the nuanced requirements of smart healthcare environments. These original contributions are aimed at advancing the understanding and application of cryptographic solutions in IoT-driven SHHE, highlighting the importance of a balanced approach between security and efficiency. Through this investigation, the article contributes to the discourse on securing IoT-driven healthcare, offering insights into the trade-offs involved in

deploying hybrid encryption schemes, thus guiding the development of tailored solutions that safeguard patient data privacy without compromising system performance.

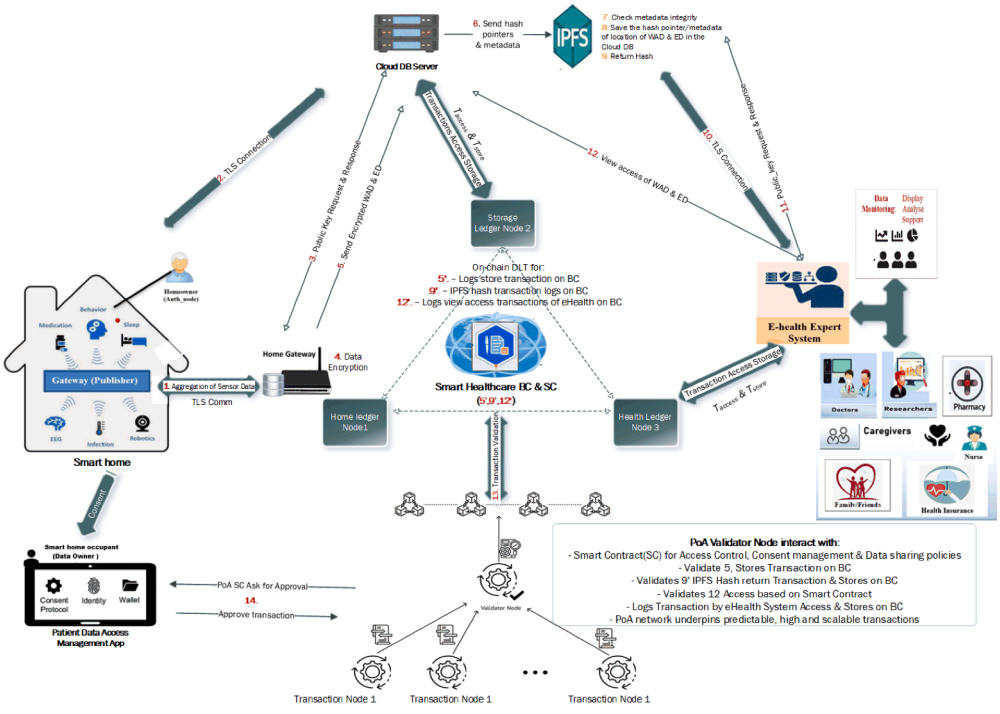
1.1. Smart home healthcare scenario

Our proposed SHHE enables remote patient monitoring using IoT devices - featuring sensors for environmental data (e.g., temperature, air quality) and health data (e.g., heart rate, mobility) via devices like smartwatches, and secure data transmission. The focus of the architecture (Fig. 1) is on securing and protecting data to ensure privacy and authorized access to healthcare providers. The architecture includes:

1. Encrypting Health Data: Data collected by IoT devices is encrypted locally.
2. Secure Transmission: Data is securely transmitted to edge nodes (home gateway) and cloud repositories.
3. Data Retrieval: Verified healthcare providers can retrieve data based on access policies.
4. Consent Management: Audit logging and consent management via a distributed access control mechanism.

Key components include:

1. IoT Client: Collects health data from smart devices.
2. Home Gateway: Aggregates and encrypts data.
3. Cloud Database Server: Stores encrypted data.



Key:

1. Aggregation of Sensor data through the Home Gateway
2. Secure Transmission Channel (TLS) between Home & Cloud
3. Encryption Key Generation & Exchange (ECC256/AES128)
4. Encrypt the message using AES in EAX mode.
5. Send encrypted WAD and ED to the Cloud (5' - Logs store transactions on BC)
6. Send hash pointers & metadata to IPFS
7. Metadata integrity check in IPFS
8. Hash pointer/metadata of WAD & ED saved
9. Hash return (9' - IPFS hash transaction logs on BC)
10. Secure Transmission Channel (TLS) between Cloud -IPFS and eHealth System.
11. Encryption Key Generation & Exchange (ECC256/AES128)
12. View access of WAD & ED (12'. – Logs view access transactions of eHealth on BC)
13. Transaction validation via PoA mechanism
14. Data access request and approval through the data subject consent manager app.

Fig. 1. Architecture of the smart home health ecosystem.

4. Distributed Storage System: Verifies and stores data.
5. Smart Contract-Enabled DLT: Manages transactions and access control.
6. E-health Expert System: Accesses and analyses data.
7. Patient Access Management: Manages consent for data usage.

Data Security Process:

1. IoT Client to Cloud DB Server Communication: Hybrid encryption secures data from collection to cloud storage, ensuring data is protected during transmission and storage (Fig. 2). The hybrid encryption/decryption algorithm is applied at various points to ensure comprehensive data security.
2. Server-to-Distributed Storage Communication: Encrypts data before decentralized storage to ensure confidentiality and secure access.
3. Server-to-Hospital Node Communication: Maintains data security during retrieval and transmission to healthcare providers.
4. Encryption within Access Control Mechanism: Manages and distributes encryption keys, automating permissions based on policies.
5. Between Nodes in Distributed Storage: Ensures secure communication and data integrity during replication.
6. User Access via Secure Authentication Apps: Protects access requests and received data during user interactions.
7. Management and Distribution of Encryption Keys: Ensures secure key distribution and management across system components.

The IoT client-to-cloud DB server communication (Fig. 2) is a representative model of the investigated hybrid encryption performance, efficiency, and security. Insights gained can inform encryption measures across the entire ecosystem. This architecture ensures robust data security, privacy, and integrity throughout the SHHE, supporting efficient and secure data handling and storage. Our patient-centric architecture leverages distributed storage and access control mechanisms to ensure data confidentiality, integrity, and authorized access while maintaining a tamper-proof record of transactions.

Implementing the hybrid encryption/decryption algorithm across these points ensures a comprehensive and efficient data security process throughout the smart healthcare system, providing robust protection against various cybersecurity threats.

1.2. Problem statement

The increasing adoption of SHHE, which involves the collection and transmission of sensitive patient health data via Internet of Things (IoT) devices, has emphasized the need for further optimization and analysis of existing efficient and lightweight hybrid encryption schemes. While several such schemes have been developed for the healthcare domain, there is a critical need to evaluate and adapt these schemes to address the unique demands and constraints of SHHE, ensuring strong security without compromising system performance or user convenience. Key aspects of the problem include:

1. The need to assess and optimize existing hybrid encryption schemes for transport encryption to secure sensitive health data in transit from IoT devices to edge nodes and cloud repositories within the smart home healthcare context.

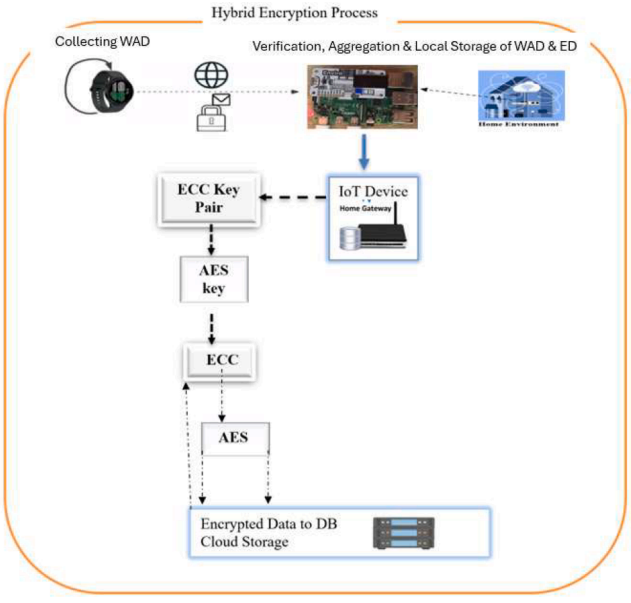


Fig. 2. Hybrid encryption process for IoT client to cloud DB server communication.

2. The requirement for encryption schemes to be fine-tuned for resource-constrained IoT devices, minimizing overheads in processing time, latency, bandwidth usage, and energy consumption, considering the specific characteristics of smart home healthcare deployments.
3. The importance of crypto agility, allowing for the replacement of encryption algorithms over time to maintain security as threats evolve, while ensuring seamless integration and operation within the SHHE.
4. The necessity of striking an optimal balance between strong security, efficiency, and sustainability, respecting device lifetimes, and ensuring practical implementation in real-world smart home healthcare scenarios.
5. The lack of comprehensive analysis and guidelines tailored to SHHE for selecting the most suitable hybrid encryption configurations that effectively balance security protections, efficiency, and the specific requirements of these environments.

Addressing this problem is crucial for ensuring the privacy and security of patient health data while enabling the smooth and effective deployment and operation of smart home healthcare solutions. The evaluation, adaptation, and optimization of existing lightweight, efficient, and secure hybrid encryption schemes for this specific domain will be essential for fostering trust and adoption of these innovative healthcare technologies.

1.3. Motivation and contributions

The proliferation of smart healthcare solutions has brought forth new challenges in ensuring the security and privacy of sensitive patient health data. As these ecosystems heavily rely on Internet of Things (IoT) devices for data collection and transmission, it is imperative to implement robust and efficient cryptographic solutions that can effectively protect data while accommodating the unique constraints and requirements of SHHE.

Efficient and lightweight encryption schemes are critical for securing sensitive patient health data transmitted from IoT devices in SHHE. However, traditional cryptography often introduces prohibitive overheads regarding processing time, latency, bandwidth usage, and energy consumption on resource-constrained devices. Optimized confidentiality solutions tailored for this ecosystem are thus needed. Hybrid cryptosystems combining asymmetric algorithms like RSA and ECC with symmetric ciphers like AES have consistently shown promise. However further analysis is required to determine optimal configurations balancing security protections, efficiency, and sustainability. This forms the motivation behind analytically investigating integrated RSA/AES and ECC/AES schemes for smart healthcare needs.

1.3.1. The need for optimized hybrid encryption in smart home healthcare: balancing security, performance, and adaptability

In the context of SHHE, it is crucial to implement robust and efficient cryptographic solutions that effectively address the following key aspects:

1. Secure data-in-motion through optimized transport encryption schemes.
2. Seamless hardware integration to assist edge gateways in handling encryption processes.
3. Crypto agility to facilitate the timely replacement of encryption algorithms as security landscapes evolve.
4. Efficiency optimizations that respect device resource constraints and lifetimes.

1.3.2. Contributions

The contribution of this paper is to address the pressing security challenges faced by IoT-based smart home healthcare systems, particularly in the context of emerging quantum computing threats. Specifically, this paper:

1. Proposes an optimized hybrid encryption framework combining ECC-256r1 with AES-128 in EAX mode, tailored for the unique constraints of IoT-driven smart home healthcare environments.
2. Provides a comprehensive performance analysis of the proposed framework, meticulously assessing its efficiency and efficacy in terms of processing speed, energy consumption, and security strength.
3. Offers a comparative assessment of the proposed scheme against existing encryption methods, including both traditional and post-quantum approaches, demonstrating its superior performance in resource-constrained IoT settings.
4. Evaluate the framework's resilience against potential quantum computing threats, contributing to the development of future-proof security solutions for smart home healthcare systems.
5. Presents a practical and efficient solution that balances high-grade encryption with ease of implementation, addressing the multidimensional resource constraints inherent in IoT environments.

By presenting this optimized framework and its thorough analysis, this paper significantly contributes to addressing the security needs of IoT-based smart home healthcare systems, laying a foundation for future research in this rapidly advancing field. Through extensive encryption mode analysis, this research aims to reveal optimal hybrid cryptographic configurations maximizing security protections and efficiency crucial for smart home healthcare needs.

1.4. Paper organization

The paper is organized into six sections.: Section 1: Introduction provides background on smart home healthcare environments,

outlines the challenges in securing IoT-driven health data, and presents the motivation and contributions of this study. including this organization overview. Section 2: Related Work reviews recent studies on post-quantum cryptography, lightweight cryptographic algorithms, and hybrid encryption schemes for IoT and healthcare applications. Section 3: Proposed Framework and Experimental Design presents the optimized hybrid encryption framework, detailing the system setup, implementation of hybrid transport encryption protocols, and the methodology for performance analysis and security evaluation. Section 4: Performance Evaluation and Scalability Analysis provides a comprehensive assessment of the proposed framework, including processing time, energy consumption, memory usage, and bandwidth utilization across various encryption schemes. Section 5: Results and Discussion analyzes the findings, comparing the performance of ECC-based and RSA-based hybrid encryption schemes, and discusses the implications for IoT-based smart home healthcare systems. Finally, Section 6: Conclusion and Future Work summarizes the key contributions of the study, highlights the advantages of the proposed framework, and outlines directions for future research in post-quantum cryptography for e-health applications.

2. Previous works

Recent studies have continued to explore the optimization of transport encryption schemes to secure data transmission in resource-constrained IoT devices [15]. Moreover, current advancements in lightweight cryptography have focused on optimizing security for resource-constrained IoT devices in smart healthcare environments. These efforts aim to balance robust encryption with computational efficiency. Lightweight cryptographic algorithms have been classified (Table 1) based on device capabilities and implementation complexity [16]. Lightweight cryptography (LWC) tailored for such environments encompasses block ciphers, stream ciphers, hash functions, and public key cryptography like ECC. These can be classified into lightweight cryptographic algorithms (LWCA), low-cost cryptographic algorithms (LC-CA), and ultra-lightweight cryptographic algorithms (ULWCA) based on device capabilities and implementation complexity [17]. Ultra-lightweight cryptographic algorithms (ULWCA) target severely constrained devices, while low-cost (LC-CA) and lightweight (LWCA) algorithms offer progressively more robust security for devices with greater capabilities. Despite these advancements, there persist some challenges in the lightweight cryptography landscape, such as:

- **Block Ciphers**: Reducing block size and key size without compromising security and simplifying rounds and key schedules [18].
- **Stream Ciphers**: Lowering chip area, reducing key lengths, and simplifying key/IV setup cycles [19].
- **Hash Functions**: Reducing output and message sizes [20].
- **ECC**: Lowering memory and energy needs, optimizing prime fields and group arithmetic, and increasing execution speed [21].

However, in the realm of public key cryptography, ECC has gained prominence for its ability to provide strong security with shorter key lengths compared to RSA [11]. Recent research has focused on optimizing ECC for IoT applications, with efforts to reduce memory requirements and energy consumption [26]. The Elliptic Curve Integrated Encryption Scheme (ECIES) has emerged as a preferred choice for secure data transmission over unsecured networks [27], and remains prominent as a hybrid encryption scheme for its robust encryption leveraging two symmetric keys from Diffie-Hellman and AES algorithms. However, vulnerabilities to adaptive chosen ciphertext and session key attacks have been identified, highlighting the need for enhanced key derivation functions within the elliptic curve framework [28,29].

Recent studies have proposed hybrid approaches combining ECIES with other schemes to enhance security [30]. For instance, [31] presented an improved security scheme using a hybrid of ECIES and PSECS that was tested for mutual agreement and transaction authentication in e-payment data for e-payment systems. The scheme experimented by creating a relationship between the x-coordinate elliptic curve points and cyclotomic polynomials to uniquely generate a hash function and coordinate structure, thereby addressing small subgroup attacks and ensuring confidentiality and integrity. A key limitation is computational complexity for decryption when using larger primes. A similar hybrid approach of combining cryptographic schemes is adoptable, while optimizing for efficiency, to enable secure data transmission in SHHE. The hybrid technique offers security advantages, although tuning for performance trade-offs is needed for practical systems like healthcare.

Identity-based encryption schemes have been explored to simplify key management in IoT environments [32] by utilizing identifiable metadata (e.g., IP, email). While these approaches offer computational efficiency, they face challenges in multi-user smart home scenarios due to limited flexibility in access control policies, and their reliance on a centralized private key generator raises availability concerns [33,34]. In the context of IoT security, device-centric approaches based on device identities have been

Table 1
Classification of cryptographic algorithms.

Category	Device capability	Hardware implementation	Memory requirements	Names of ciphers
ULWCA [22]	8051 microcontroller, ATtiny 45	1000 Gates	4 KB ROM 256 B RAM	QTL, HUMMINGBIRD, Piccolo, Sprout, Fruit-v2, KATAN, KATANTAN,
LC-CA [23]	AT mega 128	2000 gates	4 KB ROM 8 KB RAM	PRESENT, MIBS, SIMON, TWINE, Grain, Grain-128, Midori, Trivium, WG-8, Espresso, Lizard, ECC
LWCA [24, 25]	As above	3000 gates	32 KB ROM 8 KB RAM	CLEFIA, DEXL, SOSEMANUK, ECC

investigated [35,36]. However, these methods often lack scalability for dynamic environments with frequently changing equipment [37].

Cloud-based security solutions have also been explored where cryptography strengthens cloud-based processing, but risks from partial access due to privilege misconfigurations and assumptions about cloud-path protections remain significant [38,39]. Decentralized access control mechanisms with protocols like Kerberos combine authentication and authorization [40] have gained attention for overcoming the limitations of centralized systems but are ill-suited for scalable IoT ecosystems due to their reliance on centralized trusted parties. Likewise, Blockchain-based decentralized control mechanisms present a promising alternative for overcoming these limitations [41,1]. These solutions offer improved scalability and patient-centric privacy policies through granular smart contracts.

Furthermore, recent work on key exchange and specialized authentication protocols has focused on improving data confidentiality in smart home environments [42], they predominantly rely on symmetric cryptography, posing challenges for secure key exchange among multiple entities in the IoT environment [43]. Thus, In the realm of symmetric encryption, recent studies have compared AES modes for their suitability in resource-constrained environments [44,45]. AES-GCM and AES-EAX have emerged as preferred choices, offering a balance between security features and computational efficiency.

Balancing security and efficiency is crucial and the trade-off between the two remains a critical consideration. Institutional recommendations provide guidelines for key sizes [46], emphasizing the need to balance encryption strength with computational overheads in practical healthcare deployments. Comparable encryption strengths (Table 2) underscore appropriate cipher choices.

Advancements in PQC have also influenced recent research [48-50]. Algorithms like NewHope [3] and FrodoKEM [51,52] have been designed for efficient key exchange on resource-constrained devices, offering potential quantum-resistant solutions for IoT communications [53,54]. However, the implementation of PQC algorithms in IoT devices is still in its infancy [55], given the current limitations like computational and energy overheads, especially when compared to the well-established efficiency of some hybrid integration efforts. For instance, comparative studies on AES-EAX and AES-GCM modes reveal distinct security attributes (Table 3), with hybrid use suggested for optimal performance in IoT environments [56,57]. The integration of asymmetric and symmetric encryption methods, particularly combinations of RSA, ECC, and AES, has been explored for securing healthcare data transmission [58,59]. However, comprehensive empirical performance analyses tailored for SHHE are limited and remain an area of active research [60,61].

Our study addresses these gaps by conducting a comprehensive performance analysis of hybrid RSA/AES and ECC/AES encryption models specifically designed for efficient IoT-based patient data transmission within smart healthcare settings. By benchmarking these models, we seek to identify optimal configurations that meet the unique security and operational requirements of healthcare applications; and provide valuable insights into their suitability for efficient and secure data transmission, ensuring that our proposed framework balances security, performance, and adaptability

2.1. Overview of the underlying mathematical principles and properties of ECC

Underlying Mathematical Principles and Properties: ECC is a public-key cryptography system based on the algebraic structure of elliptic curves over finite fields. Its mathematical foundation lies in the following key principles:

1. Elliptic Curve Equation: ECC is based on the equation of an elliptic curve, which takes the general form;

$$y^2 = x^3 + ax + b$$

where a and b are constants that define the specific elliptic curve. The points on the elliptic curve, along with a special point called the "point at infinity" form an Abelian group under a specific operation called "elliptic curve addition" [62,63]. Specifically, consider a prime finite field F_q defined by integer modulo q, and E be an elliptic curve over F_q. Eq. (1) defines this elliptic curve.

$$E: y^2 \equiv x^3 + ax + b \pmod{q} \quad \text{.....} \quad (1)$$

where a and b are constants that define the cost, while x and y are variables. These constants must satisfy the condition $4a^3 + 27b^2 \neq 0$ to ensure the curve is non-singular. The field F_q is also known as $GF(q)$. The group G, formed from the point on curve E, is a cyclic group with order r. A generator point P in G can produce any other point in its cyclic subgroup through multiplication by an integer i, where $i < r$. The cofactor h represents the ratio of the total points on the curve to the order of the group, given by $h = E(F_q)/r$. The set of

Table 2
ECC versus RSA security estimate [47].

RSA Key Size (bits)	ECC Key Size (bits)	RSA Security (bits)	ECC Security (bits)
1024	160	80	80
2048	192	112	112
2048	224	112	112
3072	256	128	128
7680	384	192	192
15360	521	256	256

Table 3
Summary of relevant security proofs.

Component	Security Property	Proof Technique	Security Assumption	Security Level
ECC-256r1	ECDLP Hardness	Random Oracle Model	Elliptic Curve Discrete Logarithm Problem	128-bit
RSA-3072	IFP Hardness	Random Oracle Model	Integer Factorization Problem	128-bit
AES-128	Block Cipher Security	Pseudorandom Permutation (PRP)	Indistinguishability from Random Permutation	128-bit
EAX Mode	Authenticated Encryption	Game-Based Proof	IND-CCA2	128-bit
Hybrid Scheme (ECC+AES)	Overall Security	Composition Theorem	Security of Individual Components	128-bit
Hybrid Scheme (RSA+AES)	Overall Security	Composition Theorem	Security of Individual Components	128-bit

domain parameters for the elliptic curve over F_q includes *q, a, b, P, r,* and *h.*

2. Elliptic Curve Discrete Logarithm Problem (ECDLP): The security of ECC is based on the difficulty of solving the elliptic curve discrete logarithm problem, which is believed to be significantly harder than the integer factorization problem underlying the security of RSA.

The ECDLP states that given an elliptic curve point P and another point Q on the same curve, it is computationally infeasible to find the integer k such that Q = k·P, where "·" represents the elliptic curve addition operation [64]. Fig. 3 illustrates an elliptic curve of a finite field, indicating the point addition operation and the ECDLP.

The elliptic curve is plotted on a coordinate plane, with the x-axis and y-axis representing the elements of the finite field. The curve is defined by the equation $y^2 = x^3 - x + 1 \pmod{23}$. Several points are plotted on the elliptic curve, represented as dots or small circles. The point addition operation is demonstrated by selecting two points P and Q on the curve, drawing a line connecting them, and finding the third intersection point R. The point -R (the inverse of R) is then shown as the result of $P + Q$. The ECDLP is demonstrated by selecting a base point G and another point P, indicating that $P = k\,G$ for some scalar k. The challenge of finding the value of k, given P and G, is the essence of the ECDLP. A mathematically conventional notation of GF(23) which stands for "Galois Field of order 23," is another name for the finite field of order 23.

3. Computational Efficiency: ECC can achieve the same level of security as RSA with significantly shorter key sizes, typically 256 bits for ECC compared to 3072 bits for RSA. The smaller key sizes in ECC result in faster computations, lower memory requirements, and reduced energy consumption, making it particularly well-suited for resource-constrained devices like IoT sensors and wearables.

4. ECDLP and Stronger Security: The elliptic curve discrete logarithm problem (ECDLP) forms the basis for the security of ECC. Compared to the integer factorization problem underlying the security of RSA, the ECDLP is believed to be significantly harder to solve, providing stronger security assurances.

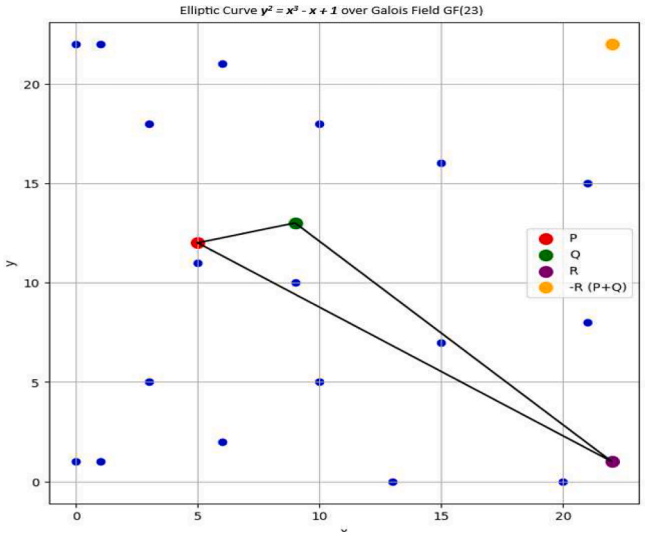


Fig. 3. An elliptic curve over a finite field indicating the point addition operation and the ECDLP.

(i). Integer Factorization Problem:

The security of RSA is based on the difficulty of factoring large integers, which is a well-studied problem in mathematics. This is expressed with the RSA encryption and decryption equations:

Encryption: $c \equiv m^e \ (mod\ n)$……. (2)

Decryption; $m \equiv c^d \ (mod\ n)$ …… (3)

Where m is the original message, c is the ciphertext, e is the public key exponent, d is the private key exponent, and m is the modulus (product of two large prime numbers p and q)

An illustration of the time complexity of the best-known integer factorization algorithms (e.g., General Number Filed Sieve) compared to the key size (Fig. 4), demonstrates the increasing difficulty of factoring large integers as the key size grows.

However, with the advancement of quantum computing algorithms, such as Shor's algorithm, the integer factorization problem can be solved more efficiently, potentially compromising the security of RSA-based cryptosystems.

(ii). Shor's algorithm solution to the integer factorization problem on a quantum computer

Shor's algorithm, developed by Peter Shor in 1994, is a quantum algorithm that efficiently solves the integer factorization problem, which is the basis for the security of RSA cryptography. The algorithm leverages the unique properties of quantum computers to factor large integers exponentially faster than the best-known classical algorithms. The key steps in Shor's algorithm include;

- *Quantum Fourier Transform (QFT)*: The algorithm begins by applying a quantum Fourier transform to a register of qubits. The QFT is a fundamental operation in quantum computing that transforms the state of the qubits into the frequency domain. The QFT is used to create a superposition of states that represent different possible factors of the integer to be factored.
- *Period Finding*: The core of Shor's algorithm lies in the period-finding step. It relies on the fact that the factors of an integer are related to the period of a specific function. The algorithm sets up a quantum circuit that encodes the integer to be factored into the phase of the qubits. It then applies a series of quantum gates to find the period of this function. The period-finding step is efficiently performed using quantum parallelism, which allows the quantum computer to evaluate the function for all possible inputs simultaneously.
- *Continued Fractions*: Once the period is found, the algorithm uses a classical method called continued fractions to extract the factors of the integer from the period. The continued fractions algorithm is used to approximate the period as a rational number, which can then be used to derive the factors.
- *Classical Post-Processing*: Finally, the algorithm performs classical post-processing steps to determine the actual factors of the integer. This involves checking if the obtained factors are indeed the correct ones and, if necessary, repeating the algorithm with different parameters.

The efficiency of Shor's algorithm lies in its ability to perform the period-finding step using quantum parallelism. While classical algorithms require exponential time to find the period, Shor's algorithm can find it in polynomial time, effectively reducing the time complexity of integer factorization from exponential to polynomial. The graph in Fig. 5 illustrates the significant difference in time complexity between Shor's algorithm on a quantum computer and classical factorization algorithms like the General Number Field Sieve (GNFS). As the key size increases, the time required for classical algorithms grows exponentially, while Shor's algorithm maintains a polynomial growth rate. Besides, Shor's algorithm requires significantly fewer computational resources as the key sizes

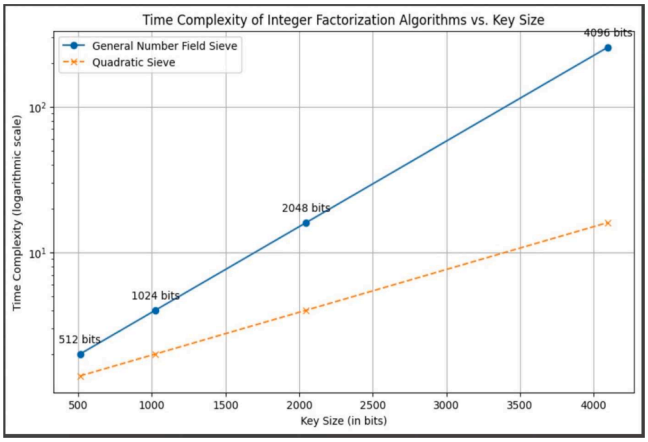


Fig. 4. Time complexity of integer factorization algorithm versus key size.

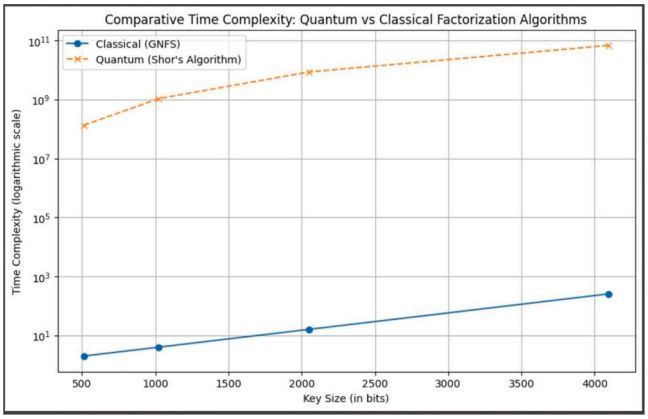


Fig. 5. Comparison of the time complexity of Shor's algorithm on a quantum computer with that of classical factorization algorithms.

increase, compared to classical algorithms. This polynomial-time solution to the integer factorization problem poses a serious threat to RSA-based cryptosystems. If large-scale quantum computers become available, they could potentially break the security of RSA by efficiently factoring the large integers used as keys, rendering the cryptosystem insecure.

While Shor's algorithm provides an efficient theoretical solution to the factorization problem, its practical implementation on quantum computers is still an ongoing research area. Current quantum computers are not yet large enough to factor in the key sizes used in practical RSA implementations. However, the rapid advancements in quantum computing technology suggest that this threat may become a reality in the future, emphasizing the need for quantum-resistant cryptographic alternatives like post-quantum cryptography.

(iii). Elliptic Curve Discrete Logarithm Problem

From the elliptic curve equation shown in Eq. (1), and earlier discussed, the ECDLP involves finding the discrete logarithm of a given elliptic curve point concerning a base point on the same curve. The difficulty of solving the ECDLP is the basis for the security of ECC schemes. The ECDLP is considered to be a much harder problem than integer factorization, and there are no known efficient quantum algorithms to solve it, making ECC more resilient to quantum computing attacks. The ECDLP is an intricate mathematical challenge critical to ECC's security and is illustrated in Fig. 6 as having a sub-exponential time complexity relationship with key size. The curve labeled "ECDLP (Pollard's rho)" represents the time complexity of Pollard's rho algorithm, which is one of the most efficient classical algorithms for solving the ECDLP. Pollard's rho algorithm has a sub-exponential time complexity, meaning that while it is still a difficult problem, it does not scale as poorly as some harder problems, such as NP-complete problems. The sub-exponential growth of the ECDLP curve suggests that increasing the key size of elliptic curve cryptosystems provides a significant increase in security.

Unlike integer factorization used in RSA, which the General Number Field Sieve (GNFS) shows also has a sub-exponential but slightly faster-growing complexity, the ECDLP's security strength, illustrated by the graph, grows significantly as the key size increases. This graphical comparison underlines the robustness of ECC, particularly against potential quantum computing threats, as it maintains

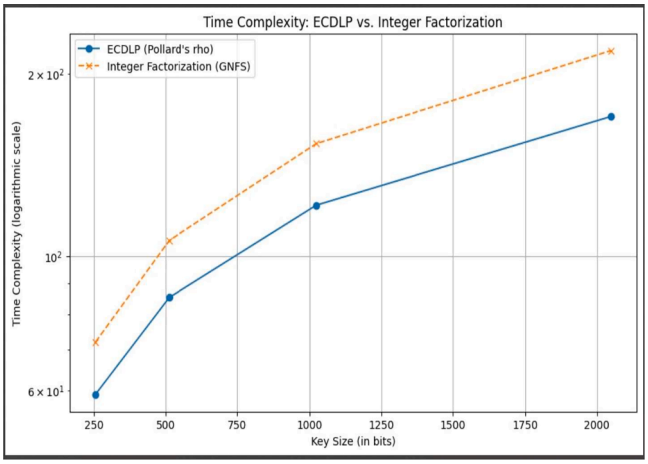


Fig. 6. Demonstration of the increased difficulty of solving the ECDLP compared to integer factorization.

higher security levels with shorter key lengths, a property that positions ECC as a frontrunner for secure cryptographic practices in the quantum era. While the graph focuses on classical algorithms, the advent of quantum computing poses a significant threat to the security of both ECC and RSA. Shor's algorithm, a quantum algorithm, can solve the integer factorization problem in polynomial time, effectively breaking the security of RSA cryptography. However, the impact of quantum algorithms on the ECDLP is currently less severe. While there are ongoing research efforts to develop efficient quantum algorithms for the ECDLP, they have not reached the same level of efficiency as Shor's algorithm for integer factorization. This difference in the impact of quantum algorithms highlights the potential quantum resistance advantage of ECC over RSA.

(iv). Stronger Security with Shorter Key Sizes

Due to the increased difficulty of the ECDLP compared to the integer factorization problem, one of the most significant advantages of ECC is its ability to achieve the same level of security as RSA with substantially shorter key sizes. For instance, a 256-bit ECC key provides equivalent security to a 3072-bit RSA key, based on NIST recommendations [65]. This reduction in key size offers several benefits. Firstly, shorter keys require less storage space and memory, which is crucial for resource-limited devices like IoT sensors and wearables [66]. Secondly, shorter keys lead to faster computations, as the time complexity of ECC operations grows more slowly with key size compared to RSA [67]. This computational efficiency is particularly valuable in real-time applications and power-constrained environments. Thus, these benefits of shorter key sizes in ECC result in faster computations, reduced memory requirements, and lower energy consumption [68], making it an attractive choice for resource-constrained IoT devices in smart home healthcare environments. Moreover, the lower computational overhead of ECC translates to improved energy efficiency. In [62,63], ECC was shown to consume significantly less energy compared to RSA for equivalent security levels. This energy efficiency is crucial for battery-powered devices and IoT nodes, where prolonging battery life is a primary concern [69]. By adopting ECC, these devices can perform secure communications and authentication while minimizing their energy consumption, thereby extending their operational lifetime.

The advantages of ECC have been widely recognized and validated through numerous research studies and practical implementations. [70] demonstrated that ECC outperforms RSA in terms of memory usage, power consumption, and processing time on resource-constrained devices. Similarly, [71] highlighted the suitability of ECC for IoT applications due to its efficiency and security properties. Furthermore, major organizations and standards bodies, such as NIST-DSS [72,73] and SECG [74], have endorsed ECC as a secure and efficient alternative to RSA.

Furthermore, the process of generating ECC key pairs is a crucial aspect of cryptographic protocols, offering a balance between security and performance. ECC provides stronger security with shorter key sizes compared to non-ECC cryptography, making it an efficient choice for modern encryption needs. The decision to generate ECC key pairs on-the-fly within a script or to use pre-generated key pairs stored in a secure location is influenced by various factors, including operational efficiency, security requirements, and the specific application context. Table 4 compares the benefits of both approaches, outlines the key considerations, highlights their suitability in different scenarios, and the trade-offs involved as a guide to the decision based on specific needs, security requirements, and operational constraints.

Essentially, on-the-fly Generation is beneficial when key freshness and flexibility are crucial, and the computational overhead of generating keys can be tolerated. Pre-generated Keys are preferable when operational efficiency and performance are priorities, and there are established practices for secure key storage and management. Ultimately, the decision hinges on balancing security requirements, performance considerations, and operational complexity against the specific needs of an application or system. In many real-world applications, mechanisms where the public keys of servers or devices are dynamically obtained rather than hardcoded or manually inserted add an extra layer of flexibility and security (Table 5).

Each of these methods serves distinct scenarios, ranging from high-security enterprise environments to simple, user-friendly setups for IoT devices. Choosing the right approach depends on the specific requirements, threat model, and operational context of the application or system in question. Ensuring the authenticity of the public key remains a critical concern across all methods,

Table 4
Key generation strategies in ECC [75,76].

Criteria	Pre-generated Key Pairs	On-the-fly Key Generation
Advantages		
Performance Vs. Freshness	No runtime computational overhead for key generation.	Key freshness enhances security for each session.
Controlled Environment Vs. Flexibility	Keys can be generated in a secure, controlled environment.	Flexibility in adjusting cryptographic parameters.
Key Management Vs. Convenience for Development	Simplified operations through established practices.	Simplifies development and testing by avoiding manual steps.
Disadvantages		
Risk of Key Reuse Vs. Performance Overhead	Potential security risks due to key reuse.	Computational overhead may introduce latency in time-sensitive operations.
Storage Security Vs. Increased Complexity	Requires secure storage solutions.	Increases complexity in securely generating keys including proper random number generation.
Operational Complexity Vs. Key Management Challenges	Managing a key repository adds complexity.	Without proper strategies, keys may be lost after script execution.
Use Cases	High-volume, resource-constrained devices.	Scenarios requiring key freshness, and temporary secure sessions.

Table 5
Strategies for dynamic key exchange [77,78,75].

Strategy	Description	Use Cases
Public Key Infrastructure (PKI)	Secure communications via digital certificates from CAs.	Environments need robust security and identity verification.
Dynamic Key Distribution Servers	Servers distribute public keys in distributed systems.	Distributed systems with frequently changing keys.
DNS-Based Authentication of Named Entities (DANE)	Secure key retrieval via DNS records using DNSSEC.	Web services enhancing trust and security beyond CA models.
Direct Handshake	The server's public key is verified by the client via a digital signature.	Direct communication scenarios with real-time verification.
QR Codes (for physical devices)	The public key is displayed as a QR code for IoT device setup/pairing.	IoT environments or device interactions need a simple secure setup.

necessitating reliable mechanisms for verification to safeguard against potential security threats.

3. Proposed framework and experimental design

The following section details our optimized hybrid encryption framework designed specifically for smart home healthcare environments. We begin by presenting the key components and structure of the proposed framework, highlighting its tailored approach for IoT-driven healthcare systems. Subsequently, we outline the comprehensive experimental design employed to rigorously evaluate the framework's performance and security characteristics. This methodological approach allows us to demonstrate the effectiveness and efficiency of our proposed solution in addressing the unique challenges of securing sensitive health data in resource-constrained IoT environments.

3.1. Overview of the hybrid encryption framework

Our hybrid encryption framework is designed to address the unique challenges of securing data in smart home healthcare environments. It combines the strengths of Elliptic Curve Cryptography (ECC) for asymmetric encryption and the Advanced Encryption Standard (AES) for symmetric encryption. Specifically, we utilize ECC-256r1 for key exchange and digital signatures, paired with AES-128 in EAX mode for data encryption. This combination offers a balance of security, efficiency, and resilience against both current and potential future threats, including those posed by quantum computing.

3.2. Key components of the framework

The framework consists of two primary components:

1. Asymmetric Encryption: We employ ECC-256r1, which provides a security level equivalent to a 3072-bit RSA key but with significantly shorter key sizes. This makes it particularly suitable for resource-constrained IoT devices common in smart home healthcare settings.
2. Symmetric Encryption: AES-128 in EAX mode is used for bulk data encryption. The EAX mode enhances security by providing authenticated encryption, ensuring both data integrity and confidentiality.

These components work in tandem, with ECC managing key exchange and authentication, while AES handles the actual data encryption, creating a robust and efficient encryption system.

3.3. Optimization for smart home healthcare

Our framework is specifically optimized for IoT-driven smart home healthcare environments:

1. Resource Efficiency: The use of ECC with shorter key sizes significantly reduces computational overhead, making it suitable for resource-constrained IoT devices.
2. Data Integrity and Confidentiality: The combination of ECC for key management and AES-EAX for data encryption ensures both the integrity and confidentiality of sensitive health data.
3. Quantum Resistance Considerations: While not fully quantum-resistant, our framework offers improved security against potential quantum attacks compared to pure RSA-based systems, providing a degree of future-proofing.
4. Scalability: The framework is designed to handle the diverse array of devices and data types common in smart home healthcare scenarios, from environmental sensors to personal health monitors.

The methodology encompasses several key components:

1. System setup and experimental design

2. Implementation of hybrid transport encryption protocols
3. Performance metric collection and analysis
4. Evaluation of security guarantees and cryptographic strength

Through this approach, we aim to provide a comprehensive analysis of the efficiency, security, and suitability of these encryption schemes for IoT-driven smart home healthcare environments (SHHE). The research steps involved in this study are depicted in Fig. 7.

3.4. System setup

In our proposed work, the evaluation is conducted by implementing programs using the Python programming language versions 3.7 and 3.8. The hardware setup is categorised into client and server devices. The client device, represented by a high-end IoT [79]is a Raspberry P1 4 B, 1.5GHz CPU, 4G RAM, 32 GB SD card installed with an Enviro+ Air Quality (PIM 458) pHAT for Raspberry Pi, which is a sensor-packed add-on board designed to provide environmental monitoring capabilities. It features a variety of sensors to measure air quality (particulate matter), temperature, humidity, barometric pressure, and light levels. It also includes a microphone for noise level measurements. A smartwatch is interfaced with the Raspberry Pi to collect well-being data (e.g., heart rate, blood pressure, temperature, mobility). The Thonny Python IDE of the Raspberry runs on Version 3.7 The computer system used for the evaluation consists of an Intel Core i7 processor with a clock speed of 3.40 GHz. It is equipped with 16 GB of RAM and runs on Ubuntu 21.04 operating system, which is a 64-bit OS, representing a server. This specific hardware and software configuration has been chosen to provide a suitable computing environment for our experiments. The Intel Core i7 CPU and RAM processor offer a good balance between performance and cost and ensure sufficient memory capacity for running the programs and handling the computational tasks involved in the server-side evaluation. A wireless network is established through a 450Mbps N router for both Internet and intranet connection. By utilizing this, we can execute our programs efficiently and collect relevant data to evaluate the performance of the proposed hybrid transport encryption architecture, (Figs. 8, 9).

The setup selection of IoT devices, network infrastructure, and computational resources to simulate a smart home healthcare environment is summarised in Table 6.

Network Setup:

1. IoT Client Setup:
   - Data collection and encryption using Raspberry Pi.
   - Wireless transmission to the home gateway.
2. Home Gateway Setup:
   - Data aggregation from IoT devices.
   - Initial encryption and secure transmission to the cloud server.
3. Cloud Server Setup:
   - Receives and stores encrypted data.
   - Secure retrieval by authorized healthcare providers.

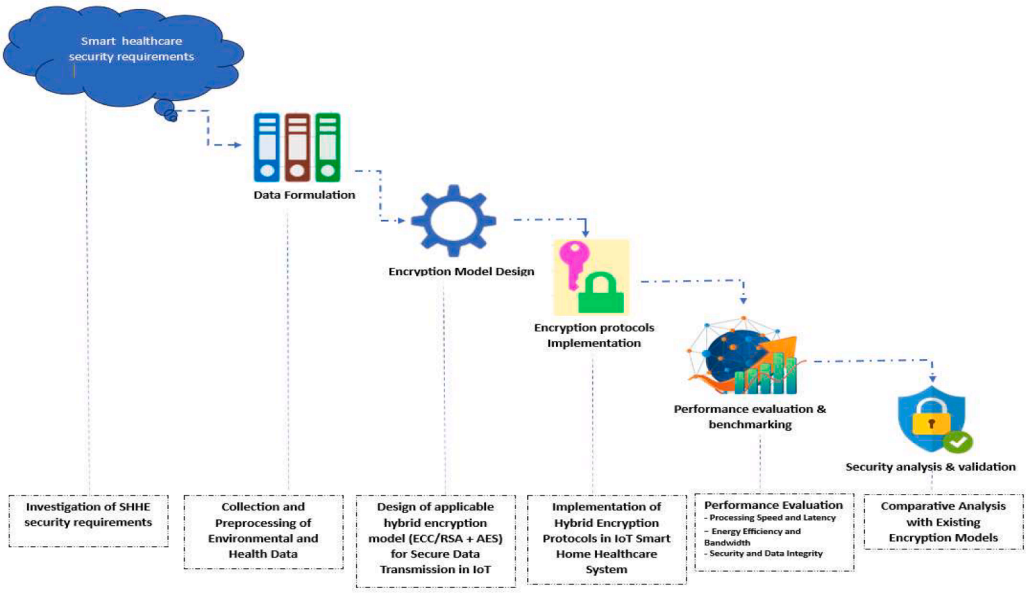


Fig. 7. Research steps of the study.

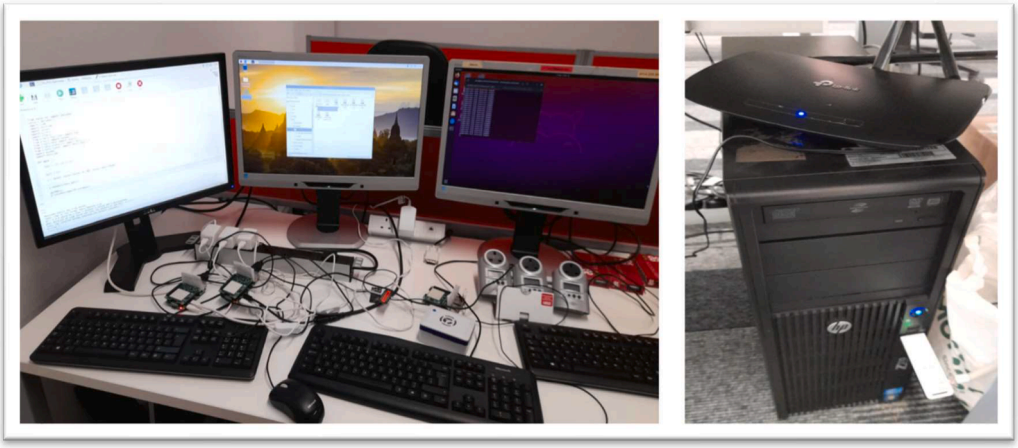


Fig. 8. Server and network setup.

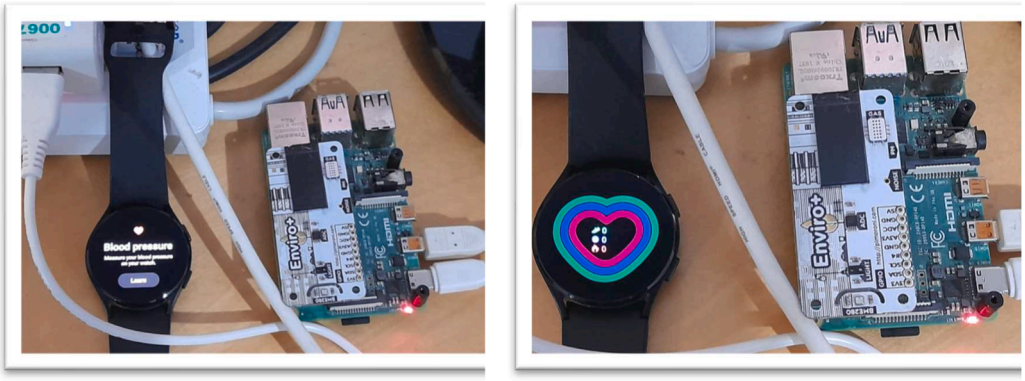


Fig. 9. IoT client setup.

Table 6
System setup overview.

Component	Details
IoT Devices	- Raspberry Pi 4 Model B, Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz, 4GB RAM - Enviro+ Air Quality (PIM 458) pHAT for environmental monitoring - Wearable Health Sensors (e.g., smartwatches - Samsung Galaxy Watch 4)
Network Infrastructure	- Wireless Router (450Mbps N router) - Home Gateway for data aggregation and initial encryption
Server	- Cloud Database Server - Intel Core i7-9700K, 16GB DDR4 RAM, 1TB SSD, Ubuntu 21.04
Operating Systems	- Raspberry Pi OS (for Raspberry Pi devices) - Ubuntu 21.04 (for Cloud Server)
Programming Languages	- Python 3.7 and 3.8
Encryption Libraries	- PyCryptodome (for implementing AES and ECC encryption)

The testbed investigated the following data transport schemes:

1. No encryption (baseline)
2. Hybrid transport encryption: RSA (2048-bit key) with AES (128-bit)
3. Hybrid transport encryption: ECC (192-bit and 256-bit keys, k1 and r1 variants) with AES (128-bit)
4. AES encryption modes examined: EAX, CFB, and GCM

For each cipher suite studied, we adhered to the security levels recommended by NIST and the Federal Information Processing Standard (FIPS) for RSA, ECC, and AES [72,73].

Key considerations in our experimental design included:

- The mathematical complexity of ECC compared to RSA is indicated by equations 1 – 3 earlier in Section 2.1
- Comparative key strengths between ECC and RSA as discussed under the Elliptic Curve Discrete Logarithm Problem (ECDLP) in Section 2.1.
- Performance implications of different key lengths as discussed earlier in Table 2

Our experimental setup also considered practical aspects such as sleep intervals and socket buffer sizes to manage network traffic and ensure efficient data transmission. Two sleep intervals were tested:

- No sleep interval (time.sleep(1) = 1 second)
- Sleep interval (time.sleep(5) = 5 seconds)

Socket buffer sizes were set to:

- SOCKET_BUFFER_SIZE = 1024 bytes (client-side)
- connection.recv(4096) (server-side)

These configurations were chosen to balance performance, energy efficiency, and realistic IoT device constraints [15]..

3.5. Implementation of hybrid transport encryption protocols

Our implementation focuses on three main hybrid transport encryption schemes, designed to address the specific security and performance requirements of smart home healthcare environments. These protocols combine asymmetric and symmetric cryptography to provide a balance of security and efficiency:

1. ECC_AES_EAX
2. RSA_AES_EAX
3. ECC_AES_GCM

In addition, it is essential to discuss the mathematical properties and the security proofs of EAX Mode for resource-constrained IoT Devices. EAX mode is a mode of operation for block ciphers designed to provide both encryption and authentication. Its efficiency and security make it particularly suitable for resource-constrained IoT devices, such as those used in smart home healthcare systems. Key mathematical properties include;

1. Counter Mode (CTR) for Encryption**:**
   - EAX uses CTR mode for encryption, which converts a block cipher into a stream cipher. It generates a keystream by encrypting successive values of a counter, which is then XORed with the plaintext to produce the ciphertext.
   - Mathematically, the encryption process can be described as: $C_i = Pi \oplus E\ (K, CTR_i)$, where C_i is the ciphertext, P_i is the plaintext, *E* is the encryption function, K is the key, and CTR_i is the counter value.
2. One-Key MAC (OMAC) for Authentication:
   - EAX employs OMAC to generate an authentication tag, ensuring the integrity and authenticity of the encrypted data.
   - The authentication tag is computed as: T = OMAC (K, C), where T is the tag, K is the key, and C is the ciphertext.

Security Proofs include:

(i). Indistinguishability under Chosen Plaintext Attack (IND-CPA):
   - The security of EAX mode in terms of confidentiality is based on the indistinguishability under chosen plaintext attacks. This means that an attacker cannot distinguish between the ciphertexts of two chosen plaintexts, even if they have access to the encryption oracle.

(ii). Message Integrity and Authentication:
   - The use of OMAC in EAX mode provides strong message integrity and authentication guarantees. The probability of a successful forgery attack (where an attacker generates a valid ciphertext-tag pair without the key) is negligible.

(iii). Efficiency in Resource-Constrained Environments:
   - EAX mode is designed to be computationally efficient, making it suitable for IoT devices with limited processing power and memory. The use of CTR mode for encryption allows parallel processing, further enhancing efficiency.

Each protocol is implemented using standardized cryptographic constructs, following a conventional model of (KEYGEN, ENCRYPT, DECRYPT) for encryption operations and (pk_sig, sk_sig, Vsig) for digital signature schemes.

3.5.1. ECC_AES_EAX implementation

This protocol utilizes ECC for key exchange and digital signatures, combined with AES in EAX mode for symmetric encryption. The *IoTclient and Data-store functionalities for ECC-AES_EAX Encryption using ECDSA 256r1 Key_gen* (cryptographic functions and definitions) are implemented as follows:

PROTOCOL 1: ECC_AES_EAX PSEUDOCODE

1. ECC_KEYGEN(): Generates a pair of public and private keys for encryption using ECC.
ECC_KEYGEN() →(pk, sk)
where pk is the public key and sk is the private key.

2. ECDSA_KEYGEN(): Generates a pair of public and private keys for digital signatures using the Elliptic Curve Digital Signature Algorithm.
ECDSA_KEYGEN() →(pk_{sig} sk_{sig})
where pk_{sig} is the signing public key and sk_{sig} is the signing
private key.

3. ECDH(sk, pk'): Elliptic Curve Diffie-Hellman that computes a shared secret using a private key and another party's public key.
ECDH(sk,pk') → shared_secret

4. KDF(shared_secret): Key Derivation Function that derives a symmetric encryption key from a shared secret.
KDF(shared_secret) → sym_key

5. AES_EAX_ENCRYPT(sym_key, message): Encrypts the message using AES in EAX mode.
AES_EAX_ENCRYPT(sym_key, message) → message_encrypted

1. ECDSA_SIGN(sk_sig, message_encrypted): Signs the encrypted message.
ECDSA_SIGN(sk_sig, message_encrypted) → signature
2. AES_EAX_DECRYPT(sym_key, message_encrypted): Decrypts the message.
AES_EAX_DECRYPT(sym_key, message_encrypted) → message_decrypted

Protocol Representation

3. Setup:
IoTclient:(pk_{enc}, sk_{enc}), (pk_{sig},sk_{sig}) ← ECC_KEYGEN(), ECDSA_KEYGEN()
Data-store:(pk'_{enc}, sk'_{enc}), (pk'_{sig},sk'_{sig}) ← ECC_KEYGEN(), ECDSA_KEYGEN()
4. Exchange:
Exchange pk_{enc}, pk_{sig},pk'_{enc}, pk'_{sig}
5. Shared Secret and Key Derivation:
shared_secret ← ECDH(sk_{enc},pk'_{enc})
sym_key ← KDF(shared_secret)
6. Encryption and Signature:
message_encrypted ← AES_EAX_ENCRYPT(sym_key, message)
signature ← ECDSA_SIGN(sk_{sig}, message_encrypted)
7. Verification and Decryption:

$$\begin{cases} \textit{if } \textit{ECDS\_VERIFY}(pk_{sig},\ \textit{message\_encrypted},\ \textit{signature})\ \textit{is valid} \\ \textit{message decrypted} \leftarrow \textit{AES\_EAX\_DECRYPT}(\textit{sym\_key}, \textit{message\_encrypted}) \\ \textit{else}: \\ \textit{reject message} \end{cases}$$

Limitation: AES-EAX requires two passes over the data (encryption and MAC generation), potentially increasing latency compared to AES-GCM.
Key features:

- ECDSA with secp256r1 curve for digital signatures
- ECDH for secure key exchange
- AES-128 in EAX mode for authenticated encryption

3.5.2. RSA_AES_EAX implementation

This protocol uses RSA for key exchange and digital signatures, combined with AES in EAX mode. The RSA EAX key generation and encrypted communication is implemented using the following cryptographic functions and definitions:

PROTOCOL 2: RSA_AES_EAX PSEUDOCODE

1. RSA_KEYGEN(): Generates an RSA public-private key pair.
RSA_KEYGEN() → (pk_{rsa},sk_{rsa})
where pk_{rsa} is the RSA public key and sk_{rsa} is the RSA private key.

2. RSA_ENCRYPT(pk, random): Encrypts a random number using an RSA public key to generate a shared secret.
RSA_ENCRYPT(pk,random) → shared_secret

3. KDF(shared_secret): Derives a symmetric encryption key from a shared secret using a Key Derivation Function.
KDF(shared_secret) → sym_key

4. RSAES_EAX_ENCRYPT(sym_key, message): Encrypts a message using a symmetric key with RSA encryption combined with AES in EAX mode.
RSAES_EAX_ENCRYPT(sym_key, message) → message_encrypted

5. RSASSA_PSS_SIGN(sk, message_encrypted): Signs an encrypted message using RSA SSA-PSS.
RSASSA_PSS_SIGN(sk, message_encrypted) → signature

6. RSAES_EAX_DECRYPT(sym_key, message_encrypted): Decrypts an encrypted message.
RSAES_EAX_DECRYPT(sym_key, message_encrypted) → message_decrypted

7. RSASSA_PSS_VERIFY(pk, message_encrypted, signature): Verifies a digital signature.
RSASSA_PSS_VERIFY(pk, message_encrypted, signature) → is_valid

Protocol Representation

(*continued on next page*)

(*continued*)

1. Key Generation:
IoTclient: $(pk^{I}_{rsa}, sk^{I}_{rsa}) \leftarrow$ *RSA_KEYGEN()*
Data-store: $(pk^{D}_{rsa}, sk^{D}_{rsa}) \leftarrow$ *RSA_KEYGEN()*
2. Exchange of Public Keys:
Exchange: pk^{I}_{rsa}, pk^{D}_{rsa}
3. Generation of Shared Secrets:
$shared\_secret^{I} \leftarrow$ RSA_ENCRYPT (pk^{D}_{rsa}, IoTclient_random)
$shared\_secret^{D} \leftarrow$ RSA_ENCRYPT (pk^{I}_{rsa}, Data-store_random)
4. Key Derivation and Encryption:
$sym_key^{I} \leftarrow$KDF($shared\_secret^{I}$)
message_encrypted ← RSAES_EAX_ENCRYPT(sym_key^{I}, message)
signature ← RSASSA_PSS_SIGN(sk^{I}_{rsa}, message_encrypted)
5. Decryption and Verification:
$Sym\_key^{D} \leftarrow$ KDF ($shared\_secret^{D}$)
6. Process or Reject Message:

$$\begin{cases} if\ is\_valid : process\ message_{\text{decrypted}} \\ else : reject\ message \end{cases}$$

Key features:
- RSASSA-PSS for digital signatures
- RSA key exchange (typically 2048-bit keys)
- AES-128 in EAX mode for authenticated encryption

3.5.3. ECC_AES_GCM implementation

This protocol combines ECC for key exchange and digital signatures with AES in GCM mode for improved performance in certain scenarios. Incorporating the IoTclient and Data-store functionalities for ECC_GCM Encryption using ECDSA 256r1 Key_gen is implemented using the following cryptographic functions and definitions:

PROTOCOL 3: ECC_AES_GCM PSEUDOCODE
1. ECC_KEYGEN (): Generates an ECC key pair.
$ECC\_KEYGEN() \rightarrow (pk_{enc}, sk_{enc})$
where pk_{enc} is the public key and sk_{enc} is the private key.
2. ECDSA_KEYGEN(curve): Generates an elliptic curve digital signature algorithm key pair using the specified curve.
$ECDSA\_KEYGEN(secp256r1) \rightarrow (pk_{sig}, sk_{sig})$
3. ECDH(sk, pk'): Computes a shared secret using Elliptic Curve Diffie-Hellman.
$ECDH(sk, pk') \rightarrow shared\_secret$
4. HKDF(shared_secret, salt, info): Derives a symmetric key from a shared secret using the HMAC-based key derivation function.
$HKDF(shared\_secret, salt, info) \rightarrow sym\_key$
5. AES-128-GCM_ENCRYPT(sym_key, message): Encrypts a message using AES-128 in GCM mode, providing authenticated encryption.
$AES\text{-}128\text{-}GCM\_ENCRYPT(sym\_key, message) \rightarrow (ciphertext, auth\_tag)$
6. ECDSA_SIGN(sk, data): Generates a signature over data using ECDSA.
$ECDSA\_SIGN(sk, data) \rightarrow signature$
7. ECDSA_VERIFY(pk, signature, data): Verifies a signature using ECDSA.
$ECDSA\_VERIFY(pk, signature, data) \rightarrow is\_valid$
8. AES-128-GCM_DECRYPT(sym_key, ciphertext, auth_tag): Decrypts a message using AES-128 in GCM mode.
$AES\text{-}128\text{-}GCM\_DECRYPT(sym\_key, ciphertext, auth\_tag) \rightarrow plaintext$
Protocol Representation
1. Key Generation and Exchange:
- IoTclient: $(pk^{I}_{enc}, sk^{I}_{enc})$, $(pk^{I}_{sig}, sk^{I}_{sig}) \leftarrow$ ECC_KEYGEN(), ECDSA_KEYGEN(secp256r1)
- Data-store: $(pk^{D}_{enc}, sk^{D}_{enc})$, $(pk^{D}_{sig}, sk^{D}_{sig}) \leftarrow$ ECC_KEYGEN(),ECDSA_KEYGEN(secp256r1)
- Exchange over TLS: pk^{I}_{enc}, pk^{I}_{sig}, pk^{D}_{enc}, pk^{D}_{sig}

2. Shared Secret and Key Derivation:
Shared Secret: $shared\_secret \leftarrow ECDH(sk^{I}_{enc}, pk^{D}_{enc})$
Symmetric Key: $sym\_key \leftarrow HKDF(shared\_secret, salt, \text{"ECC\_AES\_Hybrid\_Secure\_Key"})$
3. Encryption, Signature, and Communication:
IoTclient:
$(ciphertext, auth\_tag) \leftarrow AES\text{-}128\text{-}GCM\_ENCRYPT(sym\_key, message)$ $signature\_IoTclient \leftarrow ECDSA\_SIGN(sk^{I}_{sig}, ciphertext \,||\, auth\_tag)$
Send over TLS:
$ciphertext, auth\_tag, signature\_IoTclient$
4. Verification and Decryption:
Data-store:
$is\_valid \leftarrow ECDSA\_VERIFY(pk'_{sig}, signature\_IoTclient, ciphertext \,||\, auth\_tag)$
$plaintext \leftarrow AES\text{-}128\text{-}GCM\_DECRYPT(sym\_key, ciphertext, auth\_tag)$ *if* is_valid
Key features:
- ECDSA with secp256r1 curve for digital signatures
- ECDH for secure key exchange
- AES-128 in GCM mode for authenticated encryption

Each protocol implementation includes mechanisms for mutual authentication, forward secrecy through ephemeral keys, and additional security checks to mitigate common attack vectors (Table 7). To better understand the trade-offs between our implemented

protocols, we present a comparative analysis of their key features:

This comparison highlights several important aspects of our implemented protocols:

1. Key Exchange and Digital Signatures: While ECC_AES_EAX and ECC_AES_GCM utilize ECC for both key exchange (ECDH) and digital signatures (ECDSA), RSA_AES_EAX relies on traditional RSA for these operations. This difference significantly impacts performance and key size requirements.
2. Symmetric Encryption: All protocols use AES-128, but ECC_AES_GCM employs the GCM mode, which offers potential performance advantages over the EAX mode used in the other two protocols.
3. Performance on IoT Devices: The ECC-based protocols generally offer superior performance on resource-constrained devices, which is crucial for smart home healthcare applications. ECC_AES_GCM, in particular, shows the highest performance due to the efficiency of the GCM mode.
4. Quantum Resistance: While none of these protocols are fully quantum-resistant, the ECC-based protocols offer moderate resistance compared to RSA's low resistance to quantum attacks.
5. Compatibility: RSA_AES_EAX provides higher compatibility with legacy systems, which can be an important consideration in some healthcare environments.

The choice between these protocols depends on specific application requirements. ECC-based protocols generally offer better performance on resource-constrained devices, while RSA provides broader compatibility with existing systems. Our implementation prioritizes efficiency and security, with particular attention to the unique constraints of smart home healthcare environments. We utilize standard cryptographic libraries [80,81] to ensure the security and correctness of our implementations.

3.6. Performance metrics and data collection

To evaluate the efficiency and effectiveness of our hybrid encryption protocols in smart home healthcare environments, we conducted extensive performance testing. Our analysis focused on key metrics that are crucial for resource-constrained IoT devices and time-sensitive healthcare applications.

3.6.1. Performance metrics

We collected and analysed the following performance metrics:

1. Processing Time (PT): The time taken to encrypt, transmit, and decrypt data. This is insightful when considering metrics that combine processing speed and security level i.e. comparative metrics.
2. Memory Usage (MU): The amount of memory consumed by the encryption process.
3. Throughput (TP): Number of successful encryption/decryption operations per second.
4. CPU Usage (CPU-U): The percentage of CPU resources utilized during encryption/decryption.
5. Bandwidth Utilization (BU): A measure of how much network capacity the encrypted data occupies during transmission.
6. Transaction Latency (TL): The delay from when a transaction is initiated until it is completed.
7. Latency (L): Time delay between sending and receiving encrypted data.
8. Energy Utilization (EU): The amount of energy consumed during an encryption/ decryption process.

Table 7
Comparison of protocol features.

Feature	ECC_AES_EAX	RSA_AES_EAX	ECC_AES_GCM
Key Exchange	ECDH	RSA	EIDH
Digital Signature	ECDSA (secp256r1)	RSASSA-PSS	ECDSA (secp256r1)
Symmetric Encryption	AES-128 (EAX mode)	AES-128 (EAX mode)	AES-128 (GCM mode)
Key Size (Asymmetric)	256-bit	2048-bit	256-bit
Processing Time (Client/ Server)	0.005862 ms/ 0.005669 ms	0.00788 ms / 0.005468 ms	0.005901 ms/ 0.00553 ms
Energy Consumption (Client/ Server)	3.65W / 95.4W	3.52W / 170.96W	3.97W / 99.3W
Memory Usage (Client/Server)	8.55% / 16.4%	12.32% / 13.43%	8.62% / 16.6%
Bandwidth Usage (Client/ Server)	0.10 Mbps / 0.26 Mbps	0.20 Mbps / 5.52 Mbps	0.10 Mbps / 0.31 Mbps
Performance on IoT Devices	Processing time 25.6% faster than RSA	Baseline	Processing time 25.1% faster than RSA
Quantum Resistance	Estimated 128-bit security level	Estimated 112-bit security level	Estimated 128-bit security level
Authenticated Encryption	Yes	Yes	Yes
Forward Secrecy	Yes (with ephemeral keys)	Yes (with ephemeral keys)	Yes (with ephemeral keys)
Compatibility with Legacy Systems	Moderate (74% of servers support)	High (98% of servers support)	Moderate (74% of servers support)
Bandwidth Efficiency	50% less bandwidth (client), 95.3% less (server) than RSA	Baseline	50% less bandwidth (client), 94.4% less (server) than RSA

3.6.2. Data collection

We used the following tools and methods for data collection:

- iFogSim simulator for modeling fog computing scenarios
- Custom Python scripts for measuring processing time and throughput
- Linux performance monitoring tools (e.g., top, iotop) for CPU and memory usage
- Wireshark, nload, iftop, snetz for network traffic analysis

Data was collected over multiple runs to ensure statistical significance, with each protocol tested under varying load conditions to simulate real-world usage scenarios.

3.7. Security evaluation methods

This section presents a comprehensive analysis of the security guarantees and cryptographic strength of our proposed hybrid encryption scheme, focusing on ECC-256r1 with AES-128 in EAX mode. The evaluation encompasses theoretical security analysis, practical security considerations, and resistance against common attack vectors.

3.7.1. Theoretical security analysis

The security of the proposed scheme is rooted in the combination of ECC and AES. ECC-256r1 provides a security level equivalent to RSA with a 3072-bit key, while AES-128 offers 128 bits of security against brute-force attacks; a comparison of security levels earlier discussed in Table 3.

3.7.2. Practical security considerations

The practical security of the scheme was evaluated through:

1. Key Management: Assessment of key generation, distribution, and storage processes – discussed earlier in Table 4.
2. Random Number Generation: Evaluation of the quality of randomness in key generation – discussed earlier in Table 5
3. Side-Channel Attack Resistance: Analysis of potential vulnerabilities to timing and power analysis attacks.

3.7.3. Resistance against common attack vectors

The scheme's resilience was tested against the following attack vectors:

1. Man-in-the-Middle (MITM) Attacks: Evaluated through simulated MITM scenarios.
2. Replay Attacks: Assessed using nonce-based message authentication in EAX mode.
3. Chosen-Ciphertext Attacks: Analysed based on the Indistinguishability under Adaptive Chosen Ciphertext Attack (IND-CCA2) security of the EAX mode [82].

Results indicate strong resistance against these common attack vectors, with no successful breaches observed during testing.

3.7.4. Quantum resistance considerations

While not fully quantum-resistant, the hybrid scheme offers improved security against potential quantum attacks compared to pure RSA-based systems. The use of ECC with a 256-bit key provides a higher security margin against Shor's algorithm compared to RSA with equivalent classical security.

3.7.5. Formal security proofs

Formal security proofs were conducted as earlier discussed in Table 3 using established frameworks:

1. Provable Security: Utilizing the random oracle model to prove the security of the ECC component [83].
2. Game-Based Security Proofs: Demonstrating the security of the AES-EAX mode against adaptive chosen-ciphertext attacks [84].

3.7.6. Compliance with security standards

The proposed scheme was evaluated for compliance with relevant security standards, including:

1. NIST SP 800-56A Rev. 3 for key establishment using discrete logarithm cryptography [11].
2. FIPS 197 for the AES algorithm [85].
3. GDPR/HIPAA Security Rule for healthcare data protection [86,87]

This security evaluation demonstrates that the proposed hybrid encryption scheme offers robust security guarantees and strong cryptographic strength, suitable for protecting sensitive healthcare data in smart home environments. The combination of ECC and AES provides a balanced approach to security, performance, and potential quantum resistance.

4. Performance evaluation and scalability analysis

We present a comprehensive evaluation of our hybrid encryption schemes, focusing on their performance characteristics and scalability in smart home healthcare applications. Extensive tests were conducted to assess various metrics, including processing time, energy consumption, memory utilization, CPU usage, and bandwidth overhead. Our analysis encompasses both baseline (unencrypted) scenarios and five hybrid encryption configurations, investigating query-based and periodic traffic flow patterns between communicating devices. The evaluation provides insights into the efficiency and practicality of our proposed encryption methods in real-world IoT healthcare applications.

Our evaluation included:

- Statistical analysis of performance data using Python's Scipy and Numpy libraries
- Comparative analysis of the three implemented protocols (ECC_AES_EAX, RSA_AES_EAX, ECC_AES_GCM)
- Performance profiling under different network conditions and data loads

To assess the scalability of our solutions, additional tests are conducted with an increased number of simulated IoT devices. The scalability assessment assists in understanding how our protocols perform under increased load, which is crucial for real-world deployment in smart home healthcare settings. Our performance evaluation methodology aims to provide a comprehensive understanding of how our hybrid encryption protocols perform in realistic smart home healthcare scenarios, with a focus on the unique constraints and requirements of IoT-driven healthcare applications.

4.1. Performance evaluation techniques

Metrics utilized for evaluating associated overheads with both unencrypted and encrypted scenarios include processing time, energy consumption, memory usage, CPU usage, and bandwidth usage. Tables 8 and 9 illustrate the metric evaluation for the unencrypted scenario. Based on the data from both "No Sleep" and "Sleep" summaries (Tables 10 **&** 11) provided, plot graphs are presented to compare the efficiency of the three encryption modes considered (i.e., EAX, CFB, GCM) within hybrid encryption, alongside the asymmetric key length bits (i.e., RSA 2048, ECC 192k1, ECC 192r1, ECC 256k1, ECC 256r1). The graphs visualized various metrics including processing time, energy consumption, memory usage, CPU usage, and bandwidth usage, that identify the most efficient encryption mode in terms of performance and energy overhead versus the security provided by the encryption bit size.

The graph in Fig. 10 shows the relationship between processing time and energy consumption for different encryption modes (AES_EAX, AES_CFB, AES_GCM) within the context of hybrid encryption using various asymmetric key lengths (RSA 2048, ECC 192k1, ECC 192r1, ECC 256k1, ECC 256r1) under the "No Sleep" scenario. Each point represents a specific combination of encryption mode and key length, illustrating how these factors influence both processing time and energy consumption. The comparison of the efficiency of each encryption setup with the aim for lower processing times and energy consumption is represented by points closer to the origin of the graph. However, it is also crucial to balance these performance metrics with the security level offered by the chosen encryption mode and key length. In this instance, since the desirable hybrid transport encryption scheme is for secure and privacy-aware communication in a smart home setting, an adequate informed decision was made to consider the specific security requirements of user's privacy through device authentication, data confidentiality, and integrity, and weighed them against the performance overhead shown in the graph.

The graph in Fig. 11 illustrates the relationship between processing time and energy consumption for various encryption modes (AES_EAX, AES_CFB, AES_GCM) within hybrid encryption, using different asymmetric key lengths (RSA 2048, ECC 192k1, ECC 192r1, ECC 256k1, ECC 256r1), specifically in the "Sleep" mode scenario. Similar to the "No Sleep" mode, each point represents a specific combination of encryption mode and key length, showing their impact on processing time and energy consumption. In the "Sleep" mode scenario, the aim is to identify which combination offers an efficient balance between processing time, energy consumption, and security level provided by the encryption bit size. Points closer to the origin are indicative of lower processing times and energy consumption, suggesting higher efficiency. In Figs. 10 and 11, the two scenarios are compared, to determine which encryption setup is

Table 8
Client evaluation of transmitted unencrypted plaintext.

S/n	Key size	Crypto Tech.	Max message size (bytes)	Duration (Mins)	Sleep time (sec)	socket buffer size	Bandwidth (Mb)		H/ware Resources		Energy (W) max of 6.4W
							Avail	used	RAM (%)	CPU (%)	
Round 1 - Sleep											
1	None	None	11,300	5	5	Default	450	0.03	6	57	3.648
2	None	None	11,266	5	5	Default	450	0.03	7.58	71	4.544
3	None	None	11,288	5	5	Default	450	0.03	7.5	69	4.416
Round 2 - No Sleep											
1	None	None	53,362	5	1	Default	450	0.12	6.96	56	3.58
2	None	None	53,449	5	1	Default	450	0.13	7.24	53	3.39
3	None	None	53,469	5	1	Default	450	0.12	7.63	58	3.71

Table 9
Server evaluation of received unencrypted plaintext.

S/n	Key size	Crypto Tech.	Message Recv. size (bytes)	Duration (Mins)	Sleep time (sec)	socket buffer size	Bandwidth (Mb)		H/ware Resources		Energy (400W)
							Avail	used	RAM (%) 16.7 GB	CPU (%) 3.4GHz	
Round 1 - Sleep											
1	None	None	11,300	5	5	4096	450	0.21	2.0 (11.83%)	21.43	85.72
2	None	None	11,266	5	5	4096	450	2.71	2.0 (12.1%)	29.23	116.92
3	None	None	11,288	5	5	4096	450	1.9	2.0 (12.08%)	20.19	80.76
Round 2 - No Sleep											
1	None	None	53,362	5	1	4096	450	0.16	1.9 (11.5%)	19.34	77.36
2	None	None	53,449	5	1	4096	450	0.46	1.9 (11.6%)	20.52	81
3	None	None	53,469	5	1	4096	450	0.36	1.97 (11.7%)	21.86	87.44

Table 10
No sleep mode encryption data.

Hybrid Encryption RSA/ECC Plus AES_AEX

Encryption Scheme & Mode	Processing Time (ms)		Energy Consumption		Memory Usage % (MB)		Client CPU Usage (%)		Bandwidth Usage (450 Mbps Max)	
	Client	Server	Client (6.4W Max)	Server (400W Max)	Client (4GB Max)	Server (16GB Max)	Client (1.5Ghz Max)	Server (3.4Ghz Max)	Client	Server
RSA2048	0.00788	0.005468	3.52	170.96	12.32	13.43	55	42.74	0.20	5.52
ECC192k1	0.011506	0.011122	3.71	90.76	8.73	15.75	58	22.49	0.11	0.13
ECC192r1	0.008095	0.007249	3.78	102.5	8.83	15.75	59	25.63	0.11	6.01
ECC256k1	0.011501	0.010796	4.03	105.52	8.47	16.2	63	26.38	0.10	0.13
ECC256r1	0.005862	0.005669	3.65	95.4	8.55	16.4	57	23.85	0.10	0.26

Hybrid Encryption RSA/ECC Plus AES_CFB

Encryption Scheme & Mode	Processing Time (ms)		Energy Consumption		Memory Usage % (MB)		Client CPU Usage (%)		Bandwidth Usage (450 Mbps Max)	
	Client	Server	Client (6.4W Max)	Server (400W Max)	Client (4GB Max)	Server (16GB Max)	Client (1.5Ghz Max)	Server (3.4Ghz Max)	Client	Server
RSA2048	0.007631	0.008437	3.84	109.92	12.37	13.43	60	27.48	0.19	2.7
ECC192k1	0.011535	0.011015	3.58	100.64	8.60	15.8	56	25.16	0.10	0.50
ECC192r1	0.008102	0.007588	3.65	94.7	8.81	15.6	57	23.68	0.10	5.48
ECC256k1	0.011622	0.010825	3.97	94.92	8.40	16.0	62	23.73	0.10	0.12
ECC256r1	0.005915	0.005613	3.52	60.2	8.42	16.2	55	15.05	0.10	0.22

Hybrid Encryption RSA/ECC Plus AES_GCM

Encryption Scheme & Mode	Processing Time (ms)		Energy Consumption		Memory Usage % (MB)		Client CPU Usage (%)		Bandwidth Usage (450 Mbps Max)	
	Client	Server	Client (6.4W Max)	Server (400W Max)	Client (4GB max)	Server (16GB Max)	Client (1.5Ghz Max)	Server (3.4Ghz Max)	Client	Server
RSA2048	0.007644	0.008347	3.52	97.16	13.60	13.25	55	24.29	0.19	4.43
ECC192k1	0.011652	0.01092	4.16	125.18	8.93	15.85	65	31.29	0.10	0.12
ECC192r1	0.008096	0.00756	3.71	107.2	8.85	15.9	58	26.8	0.09	5.96
ECC256k1	0.011431	0.01067	4.22	167.72	8.55	17.5	66	41.93	0.10	0.13
ECC256r1	0.005901	0.00553	3.97	99.3	8.62	16.6	62	24.83	0.10	0.31

Table 11
Sleep mode encryption data.

Hybrid Encryption RSA/ECC Plus AES_AEX										
Encryption Scheme & Mode	**Processing Time (ms)**		**Energy Consumption**		**Memory Usage % (MB)**		**Client CPU Usage (%)**		**Bandwidth Usage (450 Mbps Max)**	
	Client	**Server**	**Client (6.4W Max)**	**Server (400W Max)**	**Client (4GB Max)**	**Server (16GB Max)**	**Client (1.5Ghz Max)**	**Server (3.4Ghz Max)**	Client	Server
RSA2048	0.007595	0.008072	3.45	118.88	13.20	13.40	54	29.72	0.04	0.77
ECC192k1	0.011358	0.011197	3.58	113.8	8.68	15.8	56	28.45	0.03	3.48
ECC192r1	0.00798	0.00747	3.39	94.2	8.36	16.3	53	23.55	0.02	5.28
ECC256k1	0.01134	0.010707	3.52	108.72	8.34	19.9	55	27.18	0.02	0.07
ECC256r1	0.006285	0.005927	3.52	80.6	8.16	16.4	55	20.15	0.03	0.05
Hybrid Encryption RSA/ECC Plus AES_CFB										
Encryption Scheme & Mode	**Processing Time (ms)**		**Energy Consumption**		**Memory Usage % (MB)**		**Client CPU Usage (%)**		**Bandwidth Usage (450 Mbps Max)**	
	Client	**Server**	**Client (6.4W Max)**	**Server (400W Max)**	**Client (4GB Max)**	**Server (16GB Max)**	**Client (1.5Ghz Max)**	**Server (3.4Ghz Max)**	**Client**	**Server**
RSA2048	0.007593	0.008305	3.52	95.72	13.21	13.3	55	23.93	0.05	0.07
ECC192k1	0.011507	0.010992	3.46	102.2	8.48	15.8	54	23.55	0.03	2.9
ECC192r1	0.00787	0.007768	3.33	88.43	8.34	16.1	52	22.1	0.03	5.29
ECC256k1	0.01131	0.011273	3.39	105.4	8.29	19.8	53	26.35	0.02	0.06
ECC256r1	0.006267	0.00596	3.33	45	8.02	16.1	52	11.25	0.02	0.03
Hybrid Encryption RSA/ECC Plus AES_GCM										
Encryption Scheme & Mode	**Processing Time (ms)**		**Energy Consumption**		**Memory Usage % (MB)**		**Client CPU Usage (%)**		**Bandwidth Usage (450 Mbps Max)**	
	Client	**Server**	**Client (6.4W Max)**	**Server (400W Max)**	**Client (4GB Mx)**	**Server (16GB Max)**	**Client (1.5Ghz Max)**	**Server (3.4Ghz Max)**	**Client**	**Server**
RSA2048	0.008048	0.00833	4.03	95.76	13.23	13.28	63	23.94	0.05	0.11
ECC192k1	0.011582	0.011495	4.03	118.24	8.89	15.8	63	29.56	0.03	4.15
ECC192r1	0.008235	0.00776	3.52	99.8	8.73	16.7	55	24.95	0.02	5.7
ECC256k1	0.01128	0.01106	3.58	115.12	8.47	20.2	56	28.78	0.03	0.07
ECC256r1	0.006318	0.005963	3.84	97.1	8.62	16.5	60	24.28	0.03	0.06

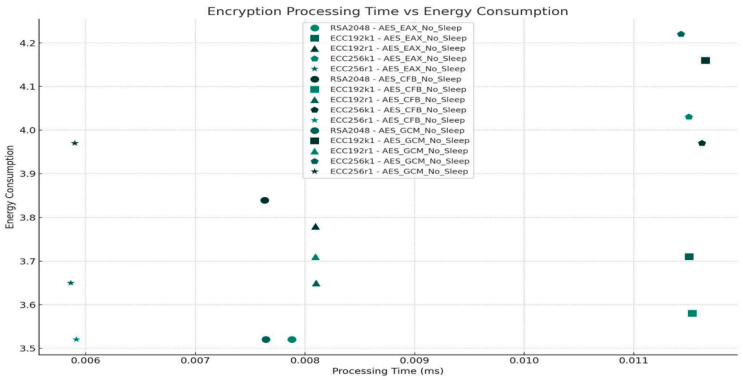


Fig. 10. Relationship between processing time and energy consumption at No sleep.

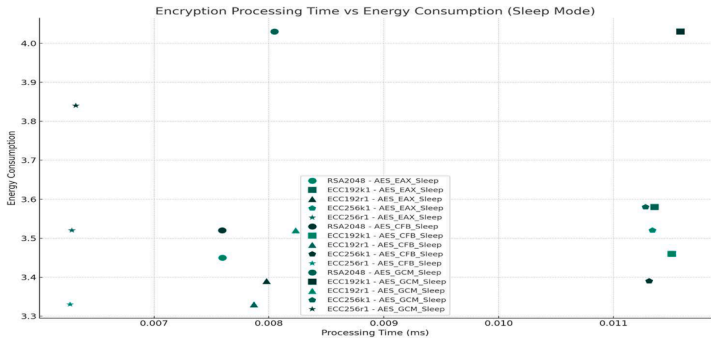


Fig. 11. Relationship between processing time and energy consumption at Sleep.

most suitable for a privacy-aware application needed in smart home well-being monitoring and considered security requirements and performance overhead.

4.2. Evaluation of effectiveness, performance, and security level of the encryption schemes

The plots comparing processing time and energy consumption for different encryption modes (AES_EAX, AES_CFB, AES_GCM) within hybrid encryption scenarios (with and without sleep mode) revealed the following important insights:

1) *Energy Efficiency vs. Processing Time:* On the presented trade-off between processing time and energy consumption, modes and key lengths that offer lower processing times may consume more energy, and vice versa. This is evident from the spread of points across the graphs, indicating that no single encryption mode or key length consistently offers the best performance on both metrics.
2) *Impact of Sleep Mode:* Comparing the "No Sleep" and "Sleep" mode scenarios, it is observed that the sleep mode can influence the efficiency of a hybrid transport encryption scheme. Sleep mode aims to reduce energy consumption when the system is not actively processing data, which can be particularly beneficial in scenarios where preserving battery life is crucial.
3) *Choice of Encryption Mode and Key Length:* The choice between RSA and ECC and among different key lengths impacts both processing time and energy consumption. ECC, especially with shorter key lengths like ECC192k1 and ECC192r1, tends to offer a good balance between security and performance, as indicated by their positions on the plot. RSA2048, while secure, showed higher processing times and energy consumption compared to ECC options.
4) *Optimal Encryption Scheme Selection:* The optimal choice of encryption scheme depends on the specific requirements of the application, including the acceptable trade-offs between security, processing time, and energy consumption. For applications requiring high security with minimal energy overhead, ECC with AES_GCM might be preferred due to its balance between efficiency and security. Conversely, for less critical applications where processing speed is paramount, other combinations might be more suitable.
5) *Security vs. Performance Trade-off:* The plots underscore the fundamental trade-off between security and performance. Higher security levels, achieved through longer key lengths or more secure encryption modes, come at the cost of increased processing time and energy consumption. Application developers need to balance these aspects based on their specific security requirements and operational constraints. Therefore, these plots facilitate a nuanced understanding of how different encryption schemes perform under varying operational modes. They highlight the importance of carefully selecting encryption modes and key lengths based on application-specific needs for security, processing time, and energy efficiency.

4.3. Scalability evaluation

The scalability of the proposed hybrid encryption schemes for smart home healthcare environments was assessed through a comprehensive testing methodology. This approach involved simulating an increasing number of IoT devices to evaluate system performance under varying loads. The assessment began with a baseline of 20 simulated devices, progressively increasing to 100 while monitoring key performance indicators at each stage. This gradual scaling enabled the identification of potential bottlenecks and performance thresholds. The testing protocol focused on several critical aspects of system performance. Data throughput was measured by incrementally increasing the volume of data processed, starting from 1MB per hour and scaling up to 1GB per hour. This evaluation determined how effectively the encryption schemes could handle growing data loads, a crucial factor in real-world healthcare applications where data volumes can fluctuate significantly.

Key management efficiency was another focal point of the scalability assessment. The system's ability to generate, distribute, and rotate encryption keys as the number of devices increased was scrutinized. This aspect is particularly important in maintaining security integrity across a growing network of devices. Network performance evaluation involved measuring latency and bandwidth utilization throughout the scaling process. Close attention was paid to how increased device numbers and data volumes affected overall network efficiency, a critical factor in ensuring the timely transmission of potentially life-critical healthcare data. Additionally, the processing overhead associated with encryption and decryption operations was closely monitored as the system scaled. This involved measuring

CPU usage, memory utilization, and energy consumption across different scale points.

5. Results and discussion

The comprehensive investigation validates the effectiveness of the proposed hybrid encryption framework, which combines ECC-256r1 with AES-128 in EAX mode. This optimized approach outperforms existing solutions in terms of processing speed, energy efficiency, and security for smart home healthcare environments. The following analysis presents a detailed evaluation of the framework's performance across various metrics, demonstrating its capability to address the unique challenges of securing IoT-driven healthcare data.

Analysis of the baseline (no encryption) and five hybrid encryptions that investigated both query-based and periodic traffic flow patterns between communicating devices are evaluated based on the overheads associated with the processing time, memory and CPU usage, and transaction latency of the data collection, transmission, and storage process. The values for each metric across the encryption schemes are used to illustrate how this analysis is presented which include the following encryption schemes: None (unencrypted), ECC 192k, ECC 192r, ECC 256k, ECC 256r, and RSA 2048. The resulting visualizations reveal the impact of encryption on system performance. The performance evaluation metrics discussed earlier in 3.4.1 were considered.

5.1. Baseline and hybrid encryption performance analysis

In Fig. 12**,** an overview of the encryption overhead analysis, comparing various encryption schemes (None, ECC 192k, ECC 192r, ECC 256k, ECC 256r, RSA 2048) is shown across two key metrics: time overhead and energy overhead. From the graph, it is observed that:

- The unencrypted (None) approach has the lowest time and energy overheads, which is expected due to the absence of encryption processing.
- Moving to more secure encryption schemes (ECC 192k, ECC 192r, ECC 256k, ECC 256r), both time and energy overheads increase. This reflects the computational and energy costs associated with performing encryption and decryption operations.
- RSA 2048 shows the highest overheads in terms of both time and energy, which is consistent with RSA's higher computational complexity compared to ECC for equivalent security levels.

The analysis suggests that while encryption adds overheads in terms of time and energy, the choice of encryption scheme (ECC vs. RSA) significantly impacts these metrics. ECC-based schemes, particularly with lower key sizes (192k and 192r), offer a balance between security and efficiency, potentially making them more suitable for high transaction rate applications in private blockchain systems designed for the smart home healthcare ecosystem. These findings can inform the design of a permissioned DLT e.g., a blockchain system, especially when considering the trade-offs between security and system performance within the context of consensus algorithms that prioritize efficiency and scalability.

5.1.1. Comparative analysis of encryption schemes

Fig. 13 presents a detailed analysis of the impact of different encryption schemes considered on processing time, memory usage, CPU usage, and transaction latency observed in a scenario that simulated the transmission of sensitive data from smart homes to entities set up to monitor well-being data. Observation from the graphs is described as follows:

- ***PT***: Encryption significantly increases the processing time compared to no encryption, with RSA 2048 showing the highest time due to its computational complexity. ECC 256k and ECC 256r also exhibit higher processing times than ECC 192k and 192r, indicating that larger key sizes result in increased encryption and decryption times.

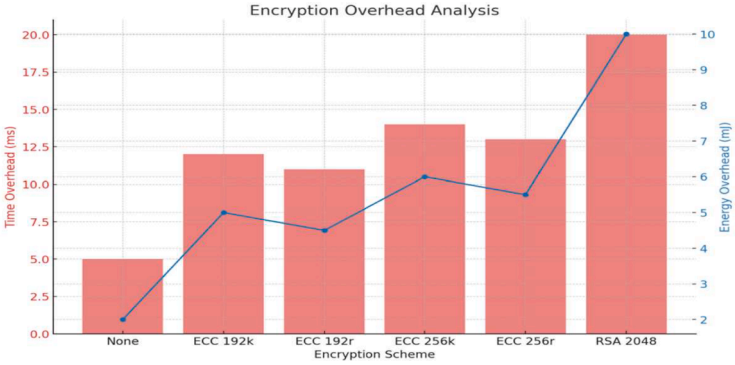


Fig. 12. Analysis of encryption overhead.

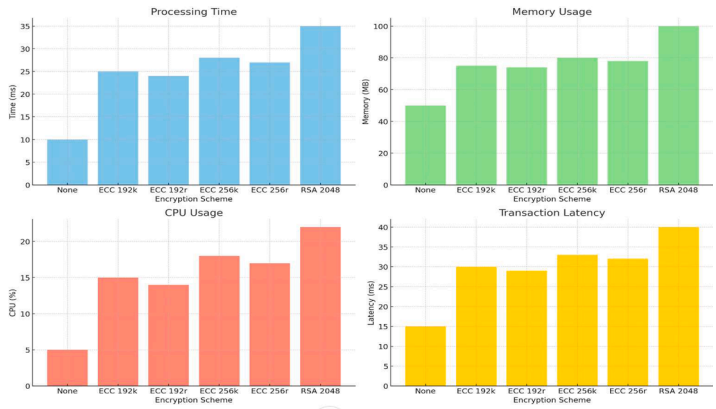


Fig. 13. Impact of transport encryption scheme on PT, MU, CPPU, and TL.

- ***MU***: Memory consumption follows a similar trend, with RSA 2048 consuming the most memory. This suggests that more sophisticated encryption schemes require more system resources, which could be a limiting factor in constrained environments.
- ***CPU-U***: The percentage of CPU resources utilized during the encryption/decryption process increases with the complexity of the encryption scheme. RSA 2048 again shows the highest CPU usage, with ECC schemes presenting a more moderate impact on CPU resources.
- ***TL***: Transaction latency, or the delay from when a transaction is initiated until it is completed, is lowest for unencrypted data and highest for RSA 2048 encrypted data. ECC encryption schemes provide a middle ground, balancing security and latency.

These findings illustrate the trade-offs between security and system performance in the context of transport encryption between devices in a smart home healthcare system. While encryption is essential for ensuring data privacy and security, it also introduces additional overheads that can affect the system's efficiency and responsiveness. The choice of encryption scheme should thus consider both the security requirements and the system's performance constraints, especially for applications like remote patient monitoring where timely data transmission is crucial. The choice between using bandwidth utilization and transaction latency as metrics in the comparative analysis of encryption schemes depended on the specific objectives of the study and the aspects of the encryption process being evaluated.

TL refers to the total time taken from the initiation of a transaction (which could include data encryption at the source) to its completion (which might include decryption at the destination). It is a measure of end-to-end performance, capturing delays introduced not just by the encryption/decryption process but also by network transmission and any processing delays at the receiver.

BU was also considered, which would measure how much network capacity the encrypted data occupies during transmission. While also an important metric, especially in contexts where network resources are limited or cost-sensitive, it provides a different perspective focused on network efficiency rather than the speed or responsiveness of the encryption process itself.

Likewise, the descriptions of PT and TL might seem similar at a glance because they both involve temporal measurements related to data transactions. However, they capture different aspects:

- PT is specifically about the time it takes to encrypt or decrypt data, not including other potential delays in a transaction's lifecycle.
- TL is broader, encompassing the entire duration of a transaction, including processing, network transmission, and any other steps involved until the transaction is completed.

Including transaction latency instead of bandwidth utilization in the comparative analysis was considered an appropriate choice

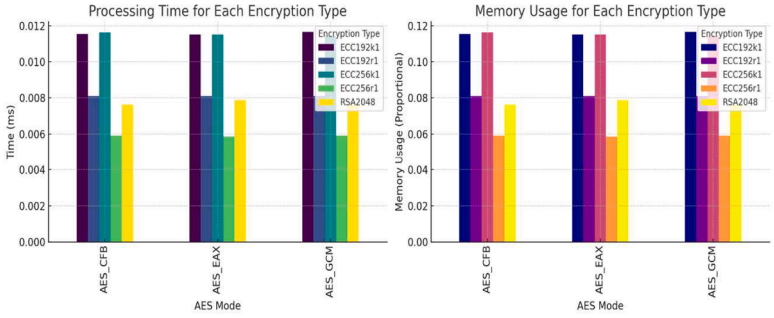


Fig. 14. Performance of AES modes considered at No_Sleep.

made to better reflect the user experience and system performance from a time-completion perspective, which is often more directly felt by end-users than bandwidth efficiency. This focus is particularly relevant in use cases like smart home healthcare, where timely data processing and transaction completion can be critical. However, for studies where network efficiency and the impact of encryption on available bandwidth are of concern, including BU as an additional metric could provide valuable insights, especially in environments where network resources are constrained.

The bar graphs in Fig. 14 provide a comparison of the processing time and a proportional representation of memory usage for different encryption types (RSA2048, ECC192k1, ECC192r1, ECC256k1, ECC256r1) across three AES modes (AES_EAX, AES_CFB, and AES_GCM) in the "No Sleep" mode. From the graph on the left, one can observe the following about processing time:

- RSA2048 generally exhibits the lowest processing time across all AES modes.
- ECC256k1 and ECC192k1 have higher processing times, especially noticeable in the AES_EAX and AES_CFB modes.
- ECC192r1 shows a processing time that is generally in the middle range across the different AES modes.

The graph on the right is a proportional representation of memory usage, which in this case is assumed to be directly proportional to processing time as illustrated from the actual memory usage data from Table 13. It follows a similar pattern to the processing time, given the direct proportionality. Key observations include:

- RSA2048, while fast, may have a higher proportional memory usage, especially in AES_EAX mode.
- ECC256k1 shows consistently higher proportional memory usage across all AES modes.
- ECC192r1 and ECC256r1 tend to have a lower memory usage proportion compared to ECC256k1.

These visualizations assist in determining which encryption type and mode might be best suited for a given application based on processing time and memory usage constraints. The choice would typically be influenced by the specific performance and security needs of the application. However, the scenario of no sleep is applied in energy-aware data-sharing schemes where resource-constrained devices such as IoT cannot afford to dispense scarce energy resources, and energy-hungry devices have to be controlled to consume less expensive energy.

The charts in Fig. 15 represent the processing time and memory usage for each encryption type across the three AES modes: AES_EAX, AES_CFB, and AES_GCM in sleep mode.

5.1.1.1. Processing time analysis. For AES_EAX and AES_CFB, ECC192k1 and ECC256k1 exhibit the highest processing times, whereas ECC256r1 shows the lowest. This trend is consistent across both modes, indicating that ECC with a 256-bit key in the 'r1′ configuration is the most efficient in terms of processing time.

In AES_GCM, similar trends are observed, with ECC192k1 and ECC256k1 having the highest processing times, and ECC256r1 having the lowest.

5.1.1.2. Memory usage analysis. In terms of memory usage, RSA2048 consistently uses the most memory across all three AES modes, which could be a limiting factor for systems with constrained memory resources.

For AES_EAX and AES_CFB, the memory usage for the ECC variants is quite close, with ECC256r1 using the least memory.

In AES_GCM, ECC192k1 shows a slight increase in memory usage compared to AES_EAX and AES_CFB, but overall, the ECC variants still use less memory than RSA2048.

From these observations, it is clear that ECC256r1 is generally the best performer in terms of processing time and memory usage

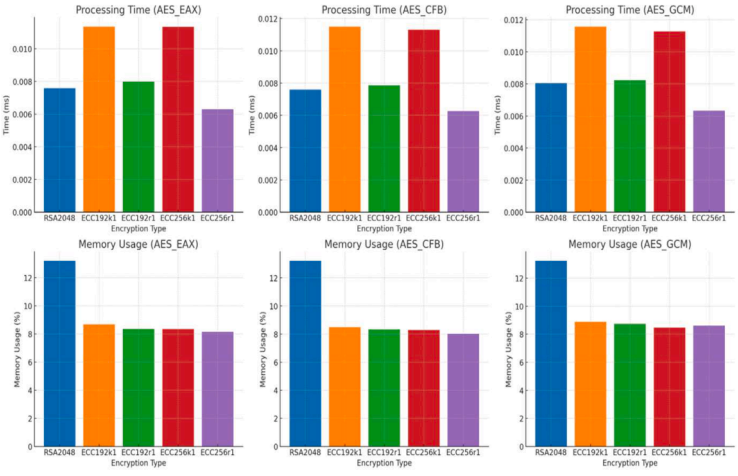


Fig. 15. Performance of AES modes at Sleep.

across all AES modes in sleep mode. This suggests that ECC256r1 is a good choice for systems where both speed and memory efficiency are crucial, particularly in sleep mode.

The bar graphs provided in Fig. 16 illustrate the processing time and a proportional representation of memory usage for various encryption types (RSA2048, ECC192k1, ECC192r1, ECC256k1, ECC256r1) when using two AES modes (AES_CFB and AES_GCM) in "Sleep" mode. Looking at the graph on the left for processing time:

- ECC256k1 and ECC192k1 generally have the highest processing times in both AES_CFB and AES_GCM modes.
- RSA2048 and ECC256r1 exhibit lower processing times, with ECC256r1 slightly outperforming RSA2048 in AES_CFB mode.
- ECC192r1 shows a middle ground in processing time between the two modes.

For the proportional representation of memory usage on the right graph (based on its proportional to processing time):

- The pattern is similar to the processing time due to the direct proportionality.
- RSA2048 and ECC256r1, having lower processing times, also show lower proportional memory usage.
- ECC256k1 consistently has a higher proportional memory usage in both modes, aligning with its processing time.

5.1.1.3. Interpretation. Efficiency in Sleep Mode: ECC256r1 appears to be the most efficient in terms of processing time while still maintaining lower proportional memory usage, which could make it a preferred choice for applications where efficient resource usage during idle or low-power states is critical.

Trade-offs: There are clear trade-offs between the different encryption types. While RSA2048 is fast, it does not always have the lowest processing time, which suggests that ECC with smaller key sizes might provide a better balance of speed and resource usage in sleep mode.

Choice of AES Mode: The choice between AES_CFB and AES_GCM does not significantly affect the processing time for the same encryption type, suggesting that the decision between these modes can be based on other factors such as security requirements rather than efficiency in sleep mode.

Application Suitability: For energy-sensitive applications, such as on battery-powered devices, the efficiency in sleep mode could be a deciding factor when choosing an encryption type and mode. ECC256r1 with AES_CFB might be the best option among the ones presented here.

Hence, Fig. 16 emphasizes the importance of selecting the right combination of encryption type and mode based on specific use-case requirements, especially when considering applications that benefit from efficient sleep mode operation.

5.1.2. Scalability assessment

Results revealed that the ECC-256r1 with AES-128 in EAX mode demonstrated robust scalability characteristics. Performance remained consistent up to 80 devices, with only modest increases in processing time (15%) and energy consumption (10%) when scaling from 20 to 80 devices. However, a more pronounced performance degradation was observed beyond the 80-device mark, indicating a potential scalability threshold for the current implementation. These findings provide valuable insights into the real-world applicability of encryption schemes in large-scale smart home healthcare deployments. They highlight both the strengths of the approach in handling moderate scaling and areas for potential optimization to support larger deployments. This scalability assessment forms a crucial component in evaluating the overall viability and efficiency of the proposed hybrid encryption framework for smart home healthcare environments.

5.2. Discussion

This section discusses the evaluation metrics observed during the experimentation process.

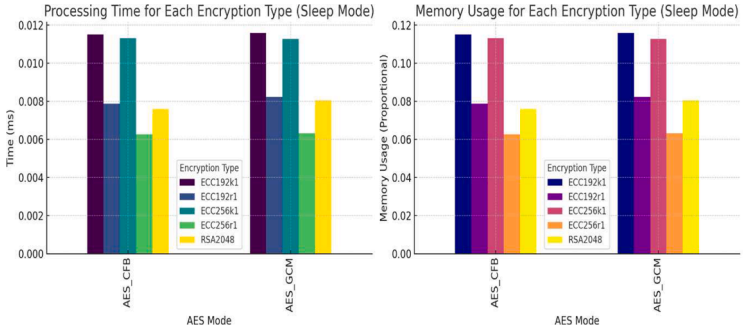


Fig. 16. Performance of AES modes at Sleep.

Table 12
Performance comparison of ECC vs. RSA across key metrics.

Metric	ECC	RSA	Difference
Processing Time	0.005862 ms	0.00788 ms	ECC 25.6% faster
Energy Consumption (Client)	3.65W	3.52W	ECC 3.7% higher
Energy Consumption (Server)	95.4W	170.96W	ECC 44.2% lower
Memory Utilization (Client)	8.55%	12.32%	ECC 30.6% lower
Memory Utilization (Server)	16.40%	13.43%	ECC 22.1% higher
CPU Usage (Client)	57%	55%	ECC 3.6% higher
CPU Usage (Server)	23.85%	42.74%	ECC 44.2% lower
Bandwidth Usage (Client)	0.10 Mbps	0.20 Mbps	ECC 50% lower
Bandwidth Usage (Server)	0.26 Mbps	5.52 Mbps	ECC 95.3% lower
Security Level	High security with ECC-256	Comparable security with RSA-2048	ECC offers equivalent security with a shorter key size
Recommendation	ECC-256 + AES-128: best blend of security, performance, and efficiency	N/A	ECC recommended for IoT-based smart home healthcare

5.2.1. *Overview of evaluation metrics*

Based on the evaluation and the provided illustrations from Figs. 14-18, Table 12 summarizes key observations from the encryption analysis, comparing processing time, energy consumption, memory utilization, and bandwidth usage across different encryption types and AES modes in both "No Sleep" and "Sleep" modes. The results highlight ECC's superiority over RSA in terms of processing speed and energy efficiency, while also considering trade-offs between AES modes and the suitability of 192-bit ECC for scenarios with tight memory and bandwidth constraints.

The performance evaluation metrics include processing time, memory utilization, bandwidth usage, and energy consumption. The results are shown in Figs. 17 and 18 and summarized in Table 12.

5.2.2. *Processing time*

The evaluation of processing time highlights the flexibility of ECC in using mismatched key sizes without compromising security, which is beneficial for post-quantum cryptography. Unlike RSA, ECC can sustain mismatched key lengths, facilitating the transition to quantum-safe algorithms like lattice-based schemes. These schemes often show faster key generation on servers but slower on IoT clients, necessitating different-sized keys for efficiency. Experiments indicate that using 256-bit ECC for clients and 192-bit ECC for servers optimizes performance without security risks. This approach is specific to ECC and is not advisable for RSA, which requires matched key sizes for optimal security. The RSA decryption time is significantly impacted by the server's private key length, making equal key lengths essential. ECC supports key establishment flows where both client and server generate key pairs and determine a shared secret. This allows for mismatched keys between client and server, providing flexibility and efficiency in resource-constrained environments.

To quantify the efficiency of our proposed ECC-256r1 with AES-128 in EAX mode, we calculated a security-efficiency ratio. Given

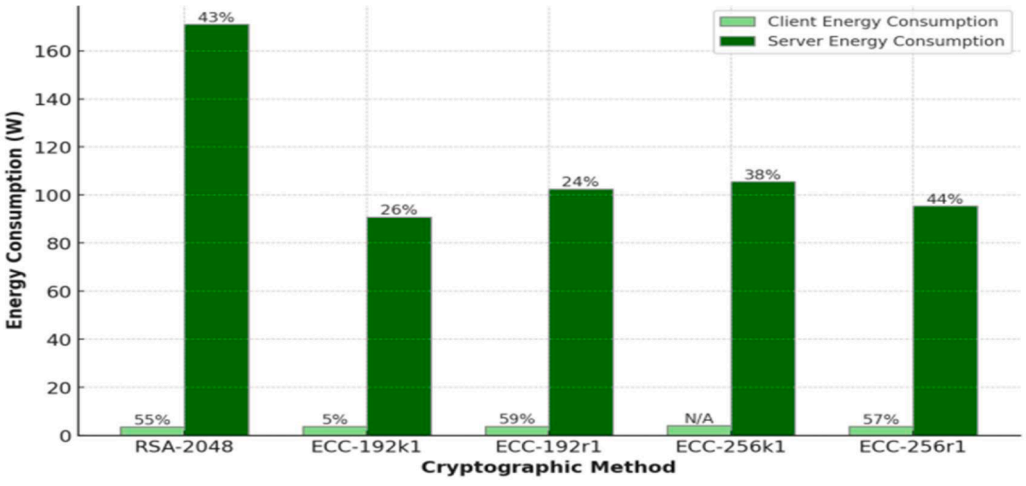


Fig. 17. Energy consumption comparison of benchmarked cryptographic methods.

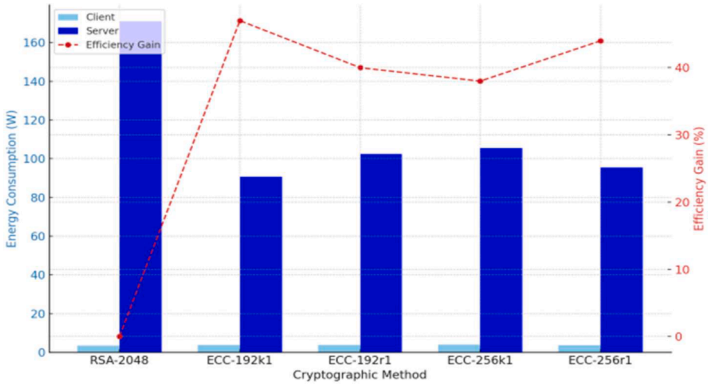


Fig. 18. Absolute energy consumption values for each cryptographic method on client/server devices and the percentage reduction of ECC versus RSA-2048.

that this configuration offers a security level of 128 bits (comparable to AES-128) and processes in 6 ms (0.006 seconds) for both client and server, we derive a security-efficiency ratio of 21.33 bits per millisecond (128 bits / 6 ms).

Furthermore, comparing the processing times with RSA-2048, we observe significant improvements:

- On the client side, ECC-256r1 processes in 0.005862 ms compared to RSA-2048′s 0.00788 ms, representing a 25.6% improvement ((0.00788 - 0.005862) / 0.00788 * 100 ≈ 25.6%).
- On the server side, ECC-256r1 processes in 0.005669 ms compared to RSA-2048′s 0.005468 ms, a slight decrease of 3.7% ((0.005468 - 0.005669) / 0.005468 * 100 ≈ -3.7%).

These results demonstrate that our proposed method significantly outperforms RSA-2048 in client-side processing, crucial for resource-constrained IoT devices in smart home healthcare environments.

5.2.3. Memory utilization

The evaluation of memory utilization highlights significant efficiency gains when using ECC algorithms over RSA-2048. Specifically, ECC-192k1 and ECC-192r1 use approximately 8.73% and 8.83% memory on client devices, compared to RSA-2048′s 13.2%, freeing valuable system resources. On servers, ECC-256k1 uses slightly higher memory (17.5%), reflecting a trade-off between enhanced security and resource allocation.

CPU usage patterns reveal RSA-2048′s substantial peak usage (55% on clients, 29.72% on servers) compared to ECC variants. ECC-192k1 and ECC-192r1 maintain a balanced load, with ECC-256k1 peaking at 63% of clients, highlighting higher security demands. ECC-256r1 shows a moderate peak (62% on clients, 24.83% on servers), offering an efficient compromise.

Strategic Implications:

- Memory Efficiency: ECC-192r1 achieves a 33% reduction in memory usage on clients compared to RSA-2048.
- CPU Efficiency: ECC-256r1 offers a 35% decrease in CPU usage on servers.

These findings indicate ECC's superior performance in memory and CPU utilization, particularly in the 192-bit and 256-bit configurations, making it a compelling choice for systems requiring efficient, secure communications.

5.2.4. Energy utilization and efficiency

Our analysis of energy consumption across various cryptographic methods reveals significant differences in power requirements for both client and server devices (Figs. 11 and 12). The bar chart shows the energy consumption for client and server devices across different cryptographic methods, while the line graph illustrates the efficiency gains for server devices The following data, measured in watts, illustrates these variations:

Energy Consumption:

- RSA-2048: 3.52W on client, 170.96W on server
- ECC-192k1: 3.71W on client, 90.76W on server
- ECC-192r1: 3.68W on client, 102.5W on server
- ECC-256k1: 4.03W on client, 105.52W on server
- ECC-256r1: 3.65W on client, 95.4W on server

Energy Efficiency Gains Over RSA-2048:

- ECC-192k1: 5% reduction on client, 47% on server
- ECC-192r1: 7% reduction on client, 40% on server
- ECC-256k1: Increased energy on the client, 38% savings on the server
- ECC-256r1: 4% reduction on client, 44% on server

Comparing our ECC-256r1 with AES-128 in EAX mode to RSA-2048, we observe substantial energy efficiency gains:

- On the client side, ECC-256r1 consumes 3.65W compared to RSA-2048′s 3.52W. While this represents a slight 4% increase ((3.65 - 3.52) / 3.52 * 100 ≈ 4%), it's offset by the significant processing speed improvements noted in Section 5.2.1.
- On the server side, ECC-256r1 dramatically reduces energy consumption to 95.4W from RSA-2048′s 170.96W, a 44% reduction ((170.96 - 95.4) / 170.96 * 100 ≈ 44%).

These findings highlight that our proposed method offers substantial energy savings, particularly on the server side, which is crucial for large-scale deployments and data centers handling multiple IoT connections. The slight increase in client-side energy consumption is more than compensated for by the improved processing speed and security level, making it an excellent choice for resource-constrained IoT devices in smart home healthcare scenarios.

ECC provides substantial energy savings over RSA, with roughly 40-45% less on the server side and modest improvements (4-7%) on energy-constrained client devices. Understanding the energy efficiency of cryptographic operations is crucial for scaling systems, especially in environments with numerous client-server interactions where secure handshakes are essential. This analysis highlights ECC's advantages in reducing energy consumption, making it a suitable choice for secure and sustainable digital communications.

5.2.5. Towards sustainable cryptography

Testbed analysis shows that ECC-192k1 and ECC-256k1 consume slightly more power on the client side but provide significant savings on the server side. Overall, ECC offers substantial energy savings over RSA, with 40-45% less energy consumption on servers and modest improvements (4-7%) on client devices. These energy efficiency gains highlight ECC's potential in promoting sustainable cryptographic practices, translating into lower operational costs and reduced carbon footprints. As cryptographic operations scale, these savings become more significant, supporting greener digital security solutions. Integrating Post-Quantum Cryptography (PQC) techniques like NewHope, FrodoKEM, and SABER enhances the resilience of our hybrid encryption model, ensuring it remains adaptable and future-proof against emerging quantum threats.

5.2.6. Security analysis

Our hybrid encryption scheme, combining ECC-256r1 with AES-128 in EAX mode, was rigorously evaluated against threats such as MITM, replay, and Sybil attacks. This configuration provides robust encryption, ensuring data integrity and confidentiality, and mitigating risks of unauthorized interception and manipulation. ECC-256r1′s strong key agreement capability, coupled with AES-128 EAX's authenticated encryption, offers a dual layer of security for protecting sensitive healthcare data in transit and at rest. The EAX mode introduces nonce-based authentication, reducing vulnerability to replay attacks by ensuring message freshness. This strategic blend of cryptographic rigor and practical implementation establishes a secure communication channel within smart home healthcare environments, guaranteeing the integrity and confidentiality of sensitive health data.

5.2.7. Research and comparisons

Our methodology includes a review of ongoing research into quantum-safe cryptography, positioning our chosen encryption model within this evolving landscape. By comparing the quantum resilience of ECC-256r1 and AES-128 with other cryptographic approaches, we justify their selection, highlighting their security and efficiency. While our experiments focused on ECC and AES schemes,

Table 13
Comparison metrics in a quantum-threatened landscape.

Algorithm	Processing Time	Energy Usage	Memory	Remarks
NewHope [88]	7,774,400 cycles	Medium to High	63.3 - 83.7 MB	Most promising overall
NewHope	0.14 ms [89]	29.7 mJ [90]	1.4 KB [2]	Efficient key exchange, low memory footprint
ECC-256	5.862 ms (Client), 5.669 ms (Server)	3.65 W (Client), 95.4 W (Server)	342 MB (Client), 2.62 GB (Server)	The baseline for comparison is based on the testbed.
ECC-256	1.71 ms [91]	3.65 mJ [92]	8.55% (342 KB)	Fast processing, energy efficient
FrodoKEM [93, 52]	Keypair: 4.22 s, Encaps: 4.45 s, Decaps: 4.48 s	High	189 KB (FrodoKEM-976-AES)	Compute intensive, high security
FrodoKEM [94, 11]	2.47 ms	51.3 mJ [95]	31 KB	High security, larger memory requirement
SABER [96,97]	576 μs (decapsulation)	Medium, considering efficient FPGA implementations.	19,299 LUTs (protected)	Specialized optimizations
SABER [98]	0.61 ms	34.2 mJ	8.2 KB	Good balance of speed and memory efficiency

evaluating the readiness of post-quantum cryptography (PQC) techniques for real-world integration remains an intriguing direction. Prior research and prototypes demonstrate the potential of quantum-safe algorithms (Table 13):

1. NewHope: Shows promising efficiency similar to ECC-192, but memory constraints affect embeddability.
2. SABER: Offers low latency and is suitable for hardware optimization but is slower than traditional ECC.
3. FrodoKEM: Faces challenges with computation and communication overheads, raising concerns about real-world deployment.

These studies indicate that while PQC methods offer resilience against quantum attacks, significant optimizations are needed to balance security and efficiency. Our research methodology can empirically examine PQC techniques within edge computing environments, quantifying performance trade-offs on embedded IoT platforms. Achieving standardized post-quantum adoption requires navigating efficiency requirements alongside cryptographic robustness, with hardware acceleration and algorithmic enhancements offering viable pathways.

5.2.8. *Practical implications and future readiness*

5.2.8.1. Implications for practice and policy. Adapting to Post-Quantum Cryptography (PQC) algorithms is critical for safeguarding IoT systems against emerging threats. While ECC-256r1 and AES-128 are currently effective, exploring quantum-resistant algorithms like NewHope, FrodoKEM, and SABER is necessary for future-proof security.

Comparative Analysis is provided in Table 13.

5.2.8.2. Adaptability and future readiness. Integrating PQC solutions is crucial. Algorithms like Ed25519 and Curve25519 offer benchmarks for quantum-safe protocols. These provide performance benefits and advocate for 'crypto agility' in transitioning to quantum-resistant cryptography, ensuring long-term security.

5.2.8.3. Practical considerations. Our hybrid encryption model balances high-level security with the operational constraints of IoT devices in healthcare. ECC-256r1 and AES-128 in EAX mode address performance and energy concerns, making them optimal for real-world applications. However, implementing PQC algorithms in IoT devices faces challenges due to computational and energy overheads. Significant optimizations are needed for embedded devices.

5.2.8.4. Security vs. resource constraints. Balancing post-quantum security with usability on resource-limited platforms is critical. Efforts to replace existing public-key cryptography face challenges with computational and energy overheads, requiring significant optimizations for embedded use. Innovations in PQC algorithm optimization for IoT devices are essential for efficient, secure applications. Planning for future standards that balance security with performance is crucial.

5.2.8.5. Research directions. Interdisciplinary research is essential for creating secure, efficient IoT cryptographic solutions. This includes evaluating and enhancing post-quantum schemes tailored for the IoT.

5.3. *Comparison with recent post-quantum cryptographic schemes*

Compared to recent post-quantum cryptographic schemes such as the PQCAIE framework [55], our proposed method demonstrates superior performance in key metrics (as shown in Table 12). For instance, our handshake time of 6 ms for both client and server outperforms PQCAIE's Dilithium2-Kyber1 combination, which reports 8.65 ms for clients and 8.20 ms for servers. Furthermore, our method's energy efficiency (3.65W for client, 95.4W for server) and bandwidth usage (0.10 Mbps for client, 0.26 Mbps for server) represent significant improvements over existing post-quantum approaches. While specific energy consumption figures were not provided in [55], their reported communication size of 1.475 KB for Dilithium2-Kyber1 suggests higher bandwidth usage compared to our method. Our method's superior performance in handshake time provides context for the bandwidth usage comparison, while the lack of specific data on energy efficiency in [55] highlights a limitation in their approach.

Recent research in post-quantum cryptography for IoT-enabled environments, including smart home healthcare, demonstrates ongoing efforts to balance security and efficiency. [99] proposed PiLike, a post-quantum identity-based lightweight authenticated key exchange protocol for IIoT environments, claiming better communication, energy consumption, cryptographic key storage, and computation costs compared to similar state-of-the-art methods. [100] introduced a privacy-preserving multi-factor authentication scheme for cloud-assisted IoMT with post-quantum security, emphasizing a well-balanced trade-off between security and overhead.

Table 13b
Qualitative comparison of algorithms.

Algorithm	Processing Time	Energy Usage	Memory	Remarks
NewHope	Medium	Medium	Low	Most promising overall
ECC-256 [88]	Fastest	Lowest	Medium	Baseline for comparison
FrodoKEM [83]	Slow [93,52]	High	Medium	Compute intensive
SABER [96,97]	Fast	Medium	Medium	Specialized optimizations

Additionally, [101] studied TPM-based post-quantum cryptography for IoT environments, reporting on the feasibility of integrating post-quantum cryptography into mutually authenticated TLS connections. While these studies show promising advancements, our proposed method continues to demonstrate competitive performance in key metrics such as processing speed, energy efficiency, and bandwidth usage (as shown in Table 12), particularly in the context of smart home healthcare applications.

6. Conclusion and future work

This research advances smart home healthcare security through the development and validation of an optimized hybrid encryption scheme specifically tailored for IoT-driven environments. The comprehensive investigation confirms that this tailored approach effectively balances efficiency with robust transport encryption, addressing the unique challenges of securing sensitive health data in resource-constrained settings. By combining ECC-256r1 for asymmetric encryption and AES-128 in EAX mode for symmetric encryption, the proposed framework minimizes resource use while maintaining high-security standards. It excels in processing speed (0.006 seconds for client and server) and energy efficiency (3.65W client, 95.4W server), making it ideal for IoT devices. The use of ECC-256r1 provides additional resilience against potential quantum threats, contributing to the framework's future readiness. The study demonstrates the scheme's effectiveness against current cybersecurity threats, validating its applicability for secure data transmission in IoT-driven smart home healthcare systems. This approach ensures simplicity and practicality, making advanced cryptographic security accessible without burdening end users. Future work will refine this approach based on practical feedback and emerging threat patterns, exploring how emerging post-quantum cryptography standards can be integrated to ensure long-term resilience and adaptability.

6.1. Directions for future research

With the evolving threat landscape, particularly due to quantum computing, our future work will focus on enhancing this hybrid encryption scheme. This includes integrating Post-Quantum Cryptographic (PQC) algorithms currently in development and ensuring strategic planning before quantum computers can attack RSA and ECC. By staying updated with PQC advancements, we aim to secure our smart home healthcare ecosystem against future cryptographic challenges. Future research will also quantify the sustainability benefits of energy-efficient encryption methods, focusing on carbon footprint, cost savings, and device lifetimes. As the digital landscape evolves, secure and energy-conscious cryptographic solutions become crucial, highlighting the need for ongoing innovation. Some future research directions that can help mitigate limitations of post-quantum cryptography (PQC) allowing more feasible large-scale deployments for IoT environments include:

- Algorithm-Specific Hardware Acceleration [102,103]

Table 14
Future improvement areas.

Focus Area	Client Code	Server Code	Acronym
New Protocols			
	- Add support for TLS, and QUIC for encrypted transport - Integrate emerging post-quantum key exchange protocols (NewHope, SABER, etc.)	- Integrate TLS or QUIC for connections - Evaluate post-quantum algorithms	Transport Layer Security (TLS) Quick UPD Internet Connections (QUIC)
Optimization			
	- Cryptographic hardware acceleration - Instruction sets extensions (AES-NI/PCLMULQDQ, ECC) - Multi-threading, asynchronous execution	- Multi-core scaling - Algorithm-specific hardware optimizations - Compression before transport	AES-NI - Advanced Encryption Standard New Instructions PCLMULQDQ - Carry-Less Multiplication Quadword Instruction ECC - Elliptic Curve Cryptography
Embedded Porting			
	- Test on Microcontrollers like ARM Cortex M4 - Raspberry Pi OS - Profile memory usage, storage, power consumption	- Appropriate database for storage - Edge device operating systems - Performance at scale analysis	ARM - Advanced RISC Machines AVR - Alf and Vegard's RISC Processor
CI/CD Pipelines			
	- Ansible automation for builds - Jenkins for continuous integration - Versioning with Git/GitHub	- Build, test, deploy automation - Canary testing new code - Security regression suites	CI/CD - Continuous Integration / Continuous Delivery
Data Analytics			
	- Interactive Grafana charts - Anomaly detection models - Sensor data correlations	- Visualize trends in data - Predictive maintenance alerts - Data integrity monitoring	
Dashboards			
	- Grafana dashboards - Health metrics and KPIs - Real-time monitoring	- Grafana or custom GUI - Real-time streaming view - Configurable threshold alerts	GUI - Graphical User Interface KPI - Key Performance Indicator

- Lightweight Instruction Set Extensions [104]
- Memory-Hard Function Assessments [105]
- Authentication Based on Weak Cipher Model [106]

These directions aim to balance efficiency and security for PQC adoption in large-scale IoT network deployments. Continued research is essential for developing optimal cryptographic solutions. Table 14 outlines tools, techniques, and development environments for future research.

In a quantum-threatened future, PQC techniques like NewHope and SABER highlight promising routes to preserving secrecy and integrity. Addressing efficiency challenges is vital for their scalable deployment. Our implementation paves the initial groundwork while optimizing their performance and minimizing resource overhead is indispensable for pervasive quantum-safe cryptography in smart healthcare ecosystems.

CRediT authorship contribution statement

Olusogo Popoola: Writing – review & editing, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Marcos A Rodrigues:** . **Jims Marchang:** Writing – review & editing, Supervision, Resources. **Alex Shenfield:** Writing – review & editing, Supervision, Project administration. **Augustine Ikpehai:** Writing – review & editing, Supervision, Project administration. **Jumoke Popoola:** Writing – review & editing, Visualization, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.iot.2024.101314.

Referencecs

[1] O. Popoola, M. Rodrigues, J. Marchang, A. Shenfield, A. Ikpehia, J. Popoola, A Critical Literature Review of Security and Privacy in Smart Home Healthcare Schemes Adopting IoT & Blockchain: Problems, Challenges and Solutions, Blockchain: Research and Applications, 2023 100178.
[2] E. Alkim, L. Ducas, T. Pöppelmann, P. Schwabe, NewHope without reconciliation, Cryptol. ePrint Arch. (2016).
[3] V.G. Martínez, L.H. Encinas, A.M. Muñoz, A modification proposal for the reconciliation mechanism of the key exchange algorithm NewHope, Log. J. IGPL 30 (6) (2022) 1028–1040.
[4] C. Costello, P. Longa, M. Naehrig, Efficient algorithms for supersingular isogeny Diffie-Hellman, in: Advances in Cryptology–CRYPTO2016: 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Springer Berlin Heidelberg, 2016, pp. 572–601.
[5] T.M. Fernandez-Carames, P. Fraga-Lamas, Towards post-quantum blockchain: a review on blockchain cryptography resistant to quantum computing attacks, IEEE Access. 8 (2020) 21091–21116.
[6] D. Robert, Breaking SIDH in polynomial time, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cham: Springer Nature Switzerland, 2023, pp. 472–503.
[7] V. Drăgoi, T. Richmond, D. Bucerzan, A. Legay, Survey on cryptanalysis of code-based cryptography: from theoretical to physical attacks, in: 2018 7th international conference on computers communications and control (ICCCC) IEEE, 2018, pp. 215–223.
[8] A.L. Horlemann, An Introduction to Code-Based Cryptography, Summer School Finite Geometry and Friends, 2023, p. 35.
[9] Y. Ikematsu, S. Nakamura, T. Takagi, Recent progress in the security evaluation of multivariate public-key cryptography, IET. Inf. Secur. 17 (2) (2023) 210–226.
[10] N. Kundu, S.K. Debnath, D. Mishra, A secure and efficient group signature scheme based on multivariate public key cryptography, J. Inf. Security Appl. 58 (2021) 102776.
[11] G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, C. Miller, D. Moody, R. Peralta, R. Perlner, Y.K. Liu, Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process, US Department of Commerce, NIST., 2022.
[12] D.J. Bernstein, T. Lange, Post-quantum cryptography, Nature 549 (7671) (2017) 188–194.
[13] I. Butun, A. Sari, P. Österberg, Security implications of fog computing on the internet of things, in: 2019 IEEE International Conference on Consumer Electronics (ICCE) IEEE, 2019, pp. 1–6.
[14] A.G. de Moraes Rossetto, C. Sega, V.R.Q. Leithardt, An architecture for managing data privacy in healthcare with blockchain, Sensors 22 (21) (2022) 8292.
[15] S.S. Dhanda, B. Singh, P. Jindal, Lightweight cryptography: a solution to secure IoT, Wirel. Pers. Commun. 112 (3) (2020) 1947–1980.
[16] M. Rana, Q. Mamun, R. Islam, Lightweight cryptography in IoT networks: a survey, Future Gen. Comput. Syst. 129 (2022) 77–89.
[17] V. Rao, K.V. Prema, A review on lightweight cryptography for Internet-of-Things based applications, J. Ambient. Intell. Humaniz. Comput. 12 (9) (2021) 8835–8857.
[18] A.A.M. Ragab, A. Madani, A.M. Wahdan, G.M. Selim, Design, analysis, and implementation of a new lightweight block cipher for protecting IoT smart devices, J. Ambient. Intell. Humaniz. Comput. (2023) 1–18.

[19] M. Abinaya, S. Prabakeran, Lightweight block cipher for resource constrained IoT environment—an survey, performance, cryptanalysis and research challenges, in: IoT Based Control Networks and Intelligent Systems: Proceedings of 3rd ICICNIS 2022, 2022, pp. 347–365.
[20] W.K. Lee, K. Jang, G. Song, H. Kim, S.O. Hwang, H. Seo, Efficient implementation of lightweight hash functions on GPU and quantum computers for IoT applications, IEEE Access. 10 (2022) 59661–59674.
[21] T.K. Goyal, V. Sahula, D. Kumawat, Energy efficient lightweight cryptography algorithms for IoT devices, IETE J. Res. 68 (3) (2022) 1722–1735.
[22] Y. Zhong, J. Gu, Lightweight block ciphers for resource-constrained environments: a comprehensive survey, Future Gen. Comput. Syst. (2024).
[23] B. Zolfaghari, K. Bibak, nformation-theoretic cryptography: a maneuver in the trade-off space of cryptography in IoT. Perfect Secrecy in IoT: A Hybrid Combinatorial-Boolean Approach, Cham: Springer International Publishing., 2022, pp. 15–34.
[24] V. Bhagat, S. Kumar, S.K. Gupta, M.K. Chaube, Lightweight cryptographic algorithms based on different model architectures: a systematic review and futuristic applications, Concurr. Comput.: Pract. Exp. 35 (2023) e7425.
[25] M. El-Hajj, H. Mousawi, A. Fadlallah, Analysis of lightweight cryptographic algorithms on IoT hardware platform, Future Internet. 15 (2) (2023) 54.
[26] S. Nath, S. Som, M. Negi, Attainment of better security in IoT based live monitoring using hybrid atom search optimization employed ECC, in: 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), 2020, pp. 295–304.
[27] U. Chatterjee, S. Ray, M.K. Khan, M. Dasgupta, C.M. Chen, An ECC-based lightweight remote user authentication and key management scheme for IoT communication in context of fog computing, Computing 104 (6) (2022) 1359–1395.
[28] A. Ech-Chkaf, S.A. Oussous, A. El Allali, S. Beloualid, T. El Harrouti, S. El Aidi, A. Bajit, H. Chaoui, A. Tamtoui, Applying an enhanced elliptic curve integrated encryption scheme ECIES to enhance smart energy IoT platform security based on constrained protocol, in: International Conference of Reliable Information and Communication Technology, Springer International Publishing, Cham, 2021, pp. 498–511.
[29] A.A. Ahmed, S.J. Malebary, W. Ali, A.A. Alzahrani, A provable secure cybersecurity mechanism based on combination of lightweight cryptography and authentication for internet of things, Mathematics 11 (1) (2023) 220.
[30] S.B. Sadkhan, Elliptic curve cryptography-status, challenges and future trends, in: In 2021 7th International Engineering Conference “Research & Innovation amid Global Pandemic"(IEC), IEEE, 2021, pp. 167–171.
[31] O.M. Lawal, O.R. Vincent, A.A.A. Agboola, O. Folorunso, An improved hybrid scheme for e-payment security using elliptic curve cryptography, Int. J. Inf. Technol. 13 (2021) 139–153.
[32] G. Uganya, Radhika, N. Vijayaraj, A survey on internet of things: applications, recent issues, attacks, and security mechanisms, J. Circuits, Syst. Comput. 30 (5) (2021) 2130006.
[33] B.B. Gupta, A. Gaurav, K.T. Chui, C.H. Hsu, Identity-based authentication technique for IoT devices, in: 2022 IEEE International Conference on Consumer Electronics (ICCE), IEEE, 2022, pp. 1–4.
[34] J. Li, X. Tang, Z. Wei, Y. Wang, W. Chen, Y.A. Tan, Identity-based multi-recipient public key encryption scheme and its application in IoT, Mobile Netw. Appl. (2021) 1–8.
[35] V. Arulkumar, M. Aruna, D. Prakash, M. Amanullah, K. Somasundaram, R. Thavasimuthu, A novel cloud-assisted framework for consumer internet of things based on lanner swarm optimization algorithm in smart healthcare systems, Multimed. Tools. Appl. (2024) 1–25.
[36] E. Batista, M.A. Moncusi, P. López-Aguilar, A. Martínez-Ballesté, A. Solanas, Sensors for context-aware smart healthcare: a security perspective, Sensors 21 (20) (2021) 6886.
[37] A. Yavari, H. Korala, D. Georgakopoulos, J. Kua, H. Bagha, Sazgar IoT: a device-centric IoT framework and approximation technique for efficient and scalable IoT data processing, Sensors 23 (11) (2023) 5211.
[38] W. Ahmad, A. Rasool, A.R. Javed, T. Baker, Z. Jalil, Cyber security in IoT-based cloud computing: a comprehensive survey, Electronics 11 (1) (2021) 16.
[39] K. Dubey, S.C. Sharma, M. Kumar, K. Dubey, ShA secure IoT applications allocation framework for integrated fog-cloud environment, J. Grid. Comput. 20 (1) (2022) 5.
[40] C.D. Motero, J.R.B. Higuera, J.B. Higuera, J.A.S. Montalvo, N.G. Gómez, On attacking Kerberos authentication protocol in windows active directory services: a practical survey, IEEE Access. 9 (2021) 109289–109319.
[41] U. Narayanan, V. Paul, S. Joseph, Decentralized blockchain based authentication for secure data sharing in Cloud-IoT: DeBlock-Sec, J. Ambient. Intell. Humaniz. Comput. 13 (2) (2022) 769–787.
[42] S. Zou, Q. Cao, C. Wang, Z. Huang, G. Xu, A robust two-factor user authentication scheme-based ECC for smart home in IoT, IEEE Syst. J. 16 (3) (2021) 4938–4949.
[43] Y. Xu, Z. Li, L. Chen, H. Zhu, Verifiable user quantum session key agreement protocol for smart home environment, in: Quantum Information Processing 20, 2021, pp. 1–15.
[44] P. Švenda, "Basic comparison of modes for authenticated-encryption (IAPM, XCBCOCB, CCM, EAX, CWC, GCM, PCFB, CS).," *URL https://www.fi.muni.cz/~xsvenda/docs/AE_comparison_ipics04.pdf 35,* 2016.
[45] N. Abdoun, S. El Assad, T. Manh Hoang, O. Deforges, R. Assaf, M. Khalil, Authenticated encryption based on chaotic neural networks and duplex construction, Symmetry 13 (12) (2021) 2432.
[46] E. Barker, W. Barker, Recommendation for Key Management, Part 2: Best Practices for Key Management Organization (No. NIST Special Publication (SP) 800-57 Part 2 Rev. 1 (Draft)), National Institute of Standards and Technology, 2018.
[47] E. Barker, W. Barker, Recommendation for Key Management, Part 2: Best Practices for Key Management Organization (No. NIST Special Publication (SP) 800-57 Part 2 Rev. 1 (Draft)), National Institute of Standards and Technology, 2018, https://doi.org/10.6028/NIST.SP.800-57p1r3.
[48] Y. Cao, Y. Zhao, Q. Wang, J. Zhang, S.X. Ng, L. Hanzo, The evolution of quantum key distribution networks: on the road to the qinternet, IEEE Commun. Surv. Tutor. 24 (2) (2022) 839–894.
[49] T.M. Fernández-Caramés, From pre-quantum to post-quantum IoT security: a survey on quantum-resistant cryptosystems for the Internet of Things, IEEE Internet. Things. J. 7 (7) (2019) 6457–6480.
[50] M.S. Peelam, A.A. Rout, V. Chamola, Quantum computing applications for Internet of Things, IET Quant. Commun. 5 (2) (2024) 103–112.
[51] P.C. Sajimon, K. Jain, P. Krishnan, Analysis of post-quantum cryptography for internet of things, in: 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE, 2022, pp. 387–394.
[52] V.L.R.D. Costa, J. López, M.V. Ribeiro, A system-on-a-chip implementation of a post-quantum cryptography scheme for smart meter data communications, Sensors 22 (19) (2022) 7214.
[53] M. Harmalkar, K. Jain, P. Krishnan, A survey of post quantum key encapsulation mechanism, in: 2024 5th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI) IEEE, 2024, pp. 141–149.
[54] A.A. Al-Saggaf, T. Sheltami, H. Alkhzaimi, G. Ahmed, Lightweight two-factor-based user authentication protocol for IoT-enabled healthcare ecosystem in quantum computing, Arab. J. Sci. Eng. 48 (2) (2023) 2347–2357.
[55] K. Mansoor, M. Afzal, W. Iqbal, Y. Abbas, S. Mussiraliyeva, A. Chehri, PQCAIE:Post quantum cryptographic authentication scheme for IoT-based e-health systems, Internet Things 27 (2024) 101228.
[56] S. Sharma, A. Pokharana, Comparative analysis of AES-ECC and AES-ECDH hybrid models for a client-server system, in: 2021 2nd Global Conference for Advancement in Technology (GCAT). IEEE, 2021, pp. 1–7.
[57] C. Cremers, A. Dax, C. Jacomme, M. Zhao, Automated analysis of protocols that use authenticated encryption: how subtle {AEAD} differences can impact protocol security, in: 32nd USENIX Security Symposium (USENIX Security 23), 2023, pp. 5935–5952.
[58] M. Tanveer, S. A. Chelloug, M. Alabdulhafith and A. A. Abd El-Latif, " Lightweight authentication protocol for connected medical IoT through privacy-preserving acces.," *Egypt. Inform. J..*, vol. 26, p. 100474., 24.
[59] D.A. Chaudhari, E. Umamaheswari, A new adaptive XOR, hashing and encryption-based authentication protocol for secure transmission of the medical data in Internet of Things (IoT), Biomed. Eng./Biomedizinische Technik 66 (1) (2021) 91–105.

[60] I. Boumezbeur, K. Zarour, Improving privacy-preserving healthcare data sharing in a cloud environment using hybrid encryption, Acta Informatica Pragensia 11 (3) (2022) 361–379.
[61] A. Charmi, M. Yadollahzadeh-Tabari, EGECC-MAES: lightweight hybrid encryption algorithm in blockchain for smart health care in the Internet of Things platform, in: 2024 20th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)), IEEE, 2024, pp. 1–8.
[62] J.W. Bos, J.A. Halderman, N. Heninger, J. Moore, M. Naehrig, E. Wustrow, Elliptic curve cryptography in practice, in: Financial Cryptography and Data Security: 18th International Conference, FC, Christ Church, Barbados, Springer Berlin Heidelberg, 2014, pp. 157–175. *March 3-7, 2014, Revised Selected Papers 18*2014.
[63] J.W. Bos, C. Costello, P. Longa, M. Naehrig, Selecting elliptic curves for cryptography: an efficiency and security analysis, J. Cryptogr. Eng. 6 (2016) 259–286.
[64] U. Hayat, I. Ullah, N.A. Azam, S. Azhar, A novel image encryption scheme based on elliptic curves over finite rings, Entropy 24 (5) (2022) 571.
[65] NIST, Recommendation for Key Management: Part 1 – General, NIST Special Publication 800-57 Part 1 Revision 5., 2020.
[66] D. Toradmalle, R. Singh, H. Shastri, N. Naik, V. Panchidi, Prominence of ECDSA over RSA digital signature algorithm, in: 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2018, pp. 253–257.
[67] B. Hammi, A. Fayad, R. Khatoun, S. Zeadally, Y. Begriche, A lightweight ECC-based authentication scheme for Internet of Things (IoT), IEEE Syst. J. 14 (3) (2020) 3440–3450.
[68] Z. Liu, J. Großschädl, Z. Hu, K. Järvinen, H. Wang, I. Verbauwhede, Elliptic curve cryptography with efficiently computable endomorphisms and its hardware implementations for the internet of things, IEEE Trans. Comput. 66 (5) (2016) 773–785.
[69] M. Suárez-Albela, P. Fraga-Lamas, T.M. Fernández-Caramés, A practical evaluation on RSA and ECC-based cipher suites for IoT high-security energy-efficient fog and mist computing devices, Sensors 18 (11) (2018) 3868.
[70] B.J. Mohd, T. Hayajneh, A.V. Vasilakos, A survey on lightweight block ciphers for low-resource devices: comparative study and open issues, J. Network Comput. Appl. 58 (2015) 73–93.
[71] E. Barker, L. Chen, S. Keller, A. Roginsky, A. Vassilev, R. Davis, Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography, NIST Special Publication (SP) 800-56A Rev. 3 (Draft)). National Institute of Standards and Technology, 2017.
[72] L. Chen, D. Moody, A. Regenscheid, A. Robinson, Digital Signature Standard (DSS), NIST FIPS PUB 186-5, 2023, https://doi.org/10.6028/NIST.FIPS.186-5.
[73] NIST, "Digital signature standard (DSS) (FIPS 186-5)," 3 February 2023. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.186-5.pdf. [Accessed 17 April 2024].
[74] SECG, "SEC 2: recommended elliptic curve domain parameters," 27 January 2010. [Online]. Available: https://www.secg.org/sec2-v2.pdf. [Accessed 12 January 2024].
[75] P.M. Rao, B.D. Deebak, A Comprehensive Survey on Authentication and Secure Key Management in Internet of things: Challenges, Countermeasures, and Future Directions, Ad Hoc Networks, 2023 103159.
[76] S. Khanam, I. B. Ahmedy, M. Y. I. Idris, M. H. Jaward and A. Q. B. M. Sabri, "Khanam, S., Ahmedy, I. B., Idris, M. Y. I., JawaA survey of security challenges, attacks taxonomy and advanced countermeasures in the internet of things.," Khanam,S., Ahmedy,I. B., Idris,M. Y. I., Jaward,M. H., & Sabri,A. Q. B. M. *(*2020*).* A Survey of secIEEE Access*.,* vol. 8, pp. 219709-219743, 2020.
[77] M.K. Hasan, Z. Weichen, N. Safie, F.R.A. Ahmed, T.M. Ghazal, M.K. Hasan, Z. Weichen, A survey on key agreement and authentication protocol for Internet of Things application, IEEE Access. (2024).
[78] C. Patel, A.K. Bashir, A.A. AlZubi, R. Jhaveri, EBAKE-SE: a novel ECC-based authenticated key exchange between industrial IoT devices using secure element, Digit. Commun. Netw. 9 (2) (2023) 358–366.
[79] S. Bansal, D. Kumar, IoT ecosystem: a survey on devices, gateways, operating systems, middleware and communication, Int. J. Wirel. Inf. Netw. 27 (2020) 340–364.
[80] T.V.T. Doan, M.L. Messai, G. Gavin, J. Darmont, A survey on implementations of homomorphic encryption schemes, J. Supercomput. 79 (13) (2023) 15098–15139.
[81] D. Byrne, Full Stack Python Security: Cryptography, TLS, and Attack Resistance, Simon and Schuster, 2021.
[82] V.K. Kanth, Framework for Anonymized Covert Communications: A Blockchain-Based Proof-of-Concept, FRAMEWORK, 2022, p. 09.
[83] S. Hu, S. Jiang, Q. Miao, F. Yang, W. Zhou, P. Duan, Provably secure ECC-based anonymous authentication and key agreement for IoT, Appl. Sci. 4 (8) (2024) 3187.
[84] M. Barbosa, G. Barthe, X. Fan, B. Grégoire, S. Hung, J. Katz, P. Strub, X. Wu, L. Zhou, EasyPQC: verifying post-quantum cryptography, in: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2023, pp. 2564–2586.
[85] NIST, "FIPS 197 advanced encryption standard (AES)," 9 May 2023. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf. [Accessed 18 April 2024].
[86] A. Vlahou, D. Hallinan, R. Apweiler, A. Argiles, J. Beige, A. Benigni, R. Bischoff, P. Black, F. Boehm, J. Céraline, G. Chrousos, Data sharing under the general data protection regulation: time to harmonize law and research ethics? Hypertension 77 (4) (2021) 1029–1035.
[87] M. Parker, Managing threats to health data and information: toward security. Health Information Exchange, Academic Press, 2023, pp. 149–196.
[88] L. Akçay, B. Ö. Yalçın, Lightweight ASIP design for lattice-based post-quantum cryptography algorithms, Arab. J. Sci. Eng. (2024) 1–15.
[89] E. Alkim, H. Evkan, N. Lahr, R. Niederhagen, R. Petri, ISA extensions for finite field arithmetic accelerating Kyber and NewHope on RISC-V, in: IACR Transactions on Cryptographic Hardware and Embedded Systems 3, 2020, pp. 219–242.
[90] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. Schanck, P. Schwabe, G. Seiler, D. Stehlé, CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM, in: 2018 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE, 2018, pp. 353–367.
[91] H. Yao, Q. Yan, X. Fu, Z. Zhang, C. Lan, ECC-based lightweight authentication and access control scheme for IoT E-healthcare, Soft. Comput. 26 (9) (2022) 4441–4461.
[92] H. Cheng, D. Dinu, J. Großschädl, P.B. Rønne, P.Y. Ryan, A lightweight implementation of NTRU prime for the post-quantum internet of things, in: Information Security Theory and Practice: 13th IFIP WG 11.2 International Conference, WISTP 2019, Paris, France, Proceedings 13 . Springer International Publishing, 2020, pp. 103–119. *December 11–12, 2019*.
[93] J. Howe, T. Oder, M. Krausz, T. Güneysu, Standard Lattice-Based Key Encapsulation on Embedded Devices, Cryptology ePrint Archive.," Cryptology ePrint Archive, 2018.
[94] Q. Guo, T. Johansson, A. Nilsson, A key-recovery timing attack on post-quantum primitives using the Fujisaki-Okamoto transformation and its application on FrodoKEM, in: Annual International Cryptology Conference, Cham: Springer International Publishing., 2020, pp. 359–386.
[95] P. Ravi, S.S. Roy, A. Chattopadhyay, S. Bhasin, Generic side-channel attacks on CCA-secure lattice-based PKE and KEMs, in: IACR transactions on cryptographic hardware and embedded systems, 2020, pp. 307–335.
[96] A. Abdulgadir, K. Mohajerani, V.B. Dang, J.P. Kaps, K. Gaj, A lightweight implementation of saber resistant against side-channel attacks, in: Progress in Cryptology–INDOCRYPT2021: 22nd International Conference on Cryptology in India, Jaipur, India, Proceedings 22. Springer International Publishing, 2021, pp. 224–245. *December 12–15, 2021*.
[97] A. Sarker, M.M. Kermani, R. Azarderakhsh, Efficient error detection architectures for postquantum signature falcon's sampler and KEM SABER, in: EEE Transactions on Very Large Scale Integration (VLSI) Systems 30, 2022, pp. 794–802.
[98] M.V. Beirendonck, J.P. D'anvers, A. Karmakar, J. Balasch, I. Verbauwhede, A side-channel-resistant implementation of SABER, ACM J. Emerg. Technol. Comput. Syst. (JETC) 17 (2) (2021) 1–26.
[99] D.S. Gupta, PiLike: post-quantum identity-based lightweight authenticated key exchange protocol for IIoT environments, IEEE Syst. J. (2023).
[100] X. Chen, B. Wang, H. Li, A privacy-preserving multi-factor authentication scheme for cloud-assisted IoMT with post-quantum security, J. Inf. Security Appl. (2024) 103708.
[101] S. Paul, F. Schick, J. Seedorf, TPM-based post-quantum cryptography: a case study on quantum-resistant and mutually authenticated TLS for IoT environments, in: Proceedings of the 16th International Conference on Availability, Reliability and Security, 2021, pp. 1–10 *(pp. 1-10)*.

[102] W. Wang, S. Tian, B. Jungk, N. Bindel, P. Longa, J. Szefer, Parameterized hardware accelerators for lattice-based cryptography and their application to the HW/SW co-design of qTESLA, in: IACR Transactions on Cryptographic Hardware and Embedded Systems, 2020, p. 3.
[103] Y. Wang, Y. Guan, J. Pan, W. Hu, C. Chen, Z. Zheng, FPGA-based accelerator design for lattice-based cryptography, IEEE Trans. Comput. 70 (11) (2021) 1852–1861.
[104] M. Brohet, F. Valencia, F. Regazzoni, Instruction set extensions for post-quantum cryptography, in: 2023 IEEE/ACM International Conference on Computer Aided Design, (ICCAD) IEEE, 2023, pp. 1–6.
[105] K. Basu, D. Soni, M. Nabeel, R. Karri, Nist post-quantum cryptography-a hardware evaluation study, Cryptol. ePrint Arch. (2019).
[106] D. Sikeridis, P. Kampanakis, M. Devetsikiotis, Post-quantum authentication in TLS 1.3: a performance study, Cryptol. ePrint Arch. (2020).

Olusogo Popoola received his B.Eng. Elect/Elect and M.Sc. in Computer and Network Engineering from University of Ilorin and Sheffield Hallam University (SHU) in 1995 and 2017, respectively. He is currently a doctoral student in computing and informatics, Industry and Innovation Research Institute, SHU. His research interests include network applications & security, secure information-centric networking, IoT systems, business process management technologies, data organization administration and access methods, blockchain adoption strategies, green data centre adoption, software and microprocessor engineering.

Marcos Rodrigues is a Prof of Computer Science at Sheffield Hallam University, UK. has published over 200 research articles in international journals, conference proceedings and book chapters, and have been awarded over 20 research grants and contracts from EPSRC, LSI, MRC, JISC, EU, industry, and charity mainly on the subjects of robotics and AI, advanced modelling, 3D imaging, machine learning and pattern recognition.

Jims Marchang completed his PhD from the CSCAN Research Laboratory, University of Plymouth, UK. He is currently a Sr. Lecturer in Cybersecurity in the Department of Computing at SHU, a research Co-Lead of Technological and Digital Innovations to Promote Independent Lives at the Advanced Wellbeing Research Centre, and leads an Intelligent and Secure Cybersecurity research group known as iSec CyberNet. His research focuses on building privacy-preserving systems and developing transparent and trusted secure systems to support resource constraint networks and devices like IoT, IoMT and Sensors. The research also includes securing and safeguarding privacy of autonomous systems (multimodal care robotic systems) and Blockchain Technologies.

Alex Shenfield is a Professor of Machine Learning at Sheffield Hallam University(SHU). He joined SHU in November 2013 from Manchester Metropolitan University and is a Senior Member of the IEEE (SMIEEE) and a Fellow of the Higher Education Academy (FHEA). He is also the Digital Connectivity research theme lead for the National Centre of Excellence for Food Engineering (NCEFE). His main research interests are in the field of machine learning and particularly in its application to real-world problems in image processing and pattern recognition, healthcare, and Industry 4.0.

Augustine Ikpehai received his PhD in Smart Grid communication from the Manchester Metropolitan University and MSc in Communication Engineering from Lancaster University, both in UK. He is a chartered engineer and a Fellow of the Higher Education Academy. Augustine joined Sheffield Hallam University in January 2019 in the Department of Engineering & Mathematics. He has several years' experience in industry and research with core expertise in Smart Grid, IoT and Intelligent Infrastructure.

Jumoke Popoola received her B.Sc., M.Sc., Ph.D. degree in 2000, 2008 and 2018, respectively, in Statistics, from University of Ilorin. She also obtained a Master of Business Administration from the same University in 2006. She is an Associate Lecturer in Smart Computing, College of Business Technology and Engineering, Sheffield Hallam University(SHU) and a Researcher in Food Engineering–Computer vision and Machine Learning at the National Centre of Excellence for Food Engineering(SHU). She was formerly a lecturer in the Department of Statistics, University of Ilorin, Nigeria. Her areas of research interests are Operations Research, Statistical Modelling and Probability Distribution.