Security Vulnerability Analysis and Enhancement of a Lightweight Sensor Node Authentication Framework

Kim Kyoung Yee[1], Haewon Byeon[2]*
Department of AI software, Inje University, 197, Inje-ro, Gimhae-si, Gyeongnam, South Korea[1]
Department of Future Technology, Korea University of Technology and Education (KOREA TECH), Cheonan 31253, South Korea[2]

***Abstract*—This study presents a comprehensive structural and mathematical security analysis of LightAuth, a lightweight authentication framework, specifically designed for smart health sensor networks. We delve into its core components and identify several critical vulnerabilities that could compromise the integrity and security of the system. Our analysis reveals that the framework suffers from insufficient freshness verification, a flawed and biased key agreement process, and the persistent exposure of fixed identifiers, which makes it susceptible to various attacks. To address these significant security weaknesses, we propose a suite of practical and effective countermeasures. These enhancements include the implementation of a robust timestamp+nonce validation mechanism to ensure message freshness and the introduction of mutual signature verification to prevent man-in-the-middle attacks. Furthermore, we advocate for the use of dynamic pseudonyms to obfuscate user identities and enhance privacy. To bolster long-term security, we also integrate perfect forward secrecy (PFS), which ensures that a compromise of a long-term key does not compromise past session keys. We conducted extensive simulations to evaluate the effectiveness of these proposed enhancements. The results demonstrate that our improvements achieve a remarkable 100% replay detection rate, while the performance degradation remains within acceptable limits, proving the practicality of our solution.**

Keywords*—Lightweight authentication; IoMT security; ECC; timestamp–nonce validation; replay resistance; formal verification; smart healthcare systems

I. Introduction

The rapid adoption of telecare medical systems has reshaped the healthcare landscape by enabling continuous monitoring, real-time diagnostics, and remote consultations. Smart healthcare infrastructures now integrate sensor nodes, wearable devices, cloud-assisted servers, and mobile gateways, forming an interconnected ecosystem designed to provide timely interventions and improved patient outcomes. These systems, however, are highly dependent on the integrity and confidentiality of the transmitted data [1], as even a minor breach could compromise patient privacy and undermine trust in the healthcare service provider. The central role of authentication mechanisms in safeguarding these networks is undeniable, since they serve as the first line of defense against unauthorized access, replay intrusions, and identity spoofing attacks.

Traditional authentication methods, often based on static credentials such as passwords or pre-shared keys, have proven inadequate in the face of increasingly sophisticated adversaries [2, 3]. With the proliferation of wireless communication channels and resource-constrained devices, attackers can exploit weak key management practices, predictable token generation, or inadequate freshness validation to compromise entire healthcare networks [4]. Thus, a pressing need exists for lightweight authentication protocols that strike a balance between computational efficiency and robust security guarantees. In this context, schemes based on Elliptic Curve Cryptography (ECC) and hash-based tokens have gained prominence because they can provide strong cryptographic assurances with relatively small key sizes and modest overhead [5, 6]. Yet, these same protocols may embed structural flaws, particularly when critical operations such as session key derivation, nonce validation, or forward secrecy are insufficiently addressed [7].

The LightAuth framework, as described in prior work by Adil et al. [8], represents one such attempt to reconcile lightweight computation with secure authentication (see Fig. 1). It operates through distinct phases—registration, mutual authentication, session key derivation, and secure communication—each employing minimal cryptographic operations tailored to low-power sensor nodes. On studying, the protocol promises efficiency and practical deployment feasibility. However, real-world conditions reveal vulnerabilities that were not fully considered during the design process. For example, replay resistance relies heavily on loose timestamp validation, which may permit adversaries to resend previously captured tokens if synchronization tolerances are exploited. Likewise, the session key derivation process, expressed as $SK = h(r_s \times r_u \times G)$, may be susceptible to man-in-the-middle manipulation if exchanged values are not rigorously authenticated. The persistent use of static identifiers further exposes the protocol to privacy leakage, enabling adversaries to correlate communications across sessions and build longitudinal tracking profiles of patients.

*Corresponding author.

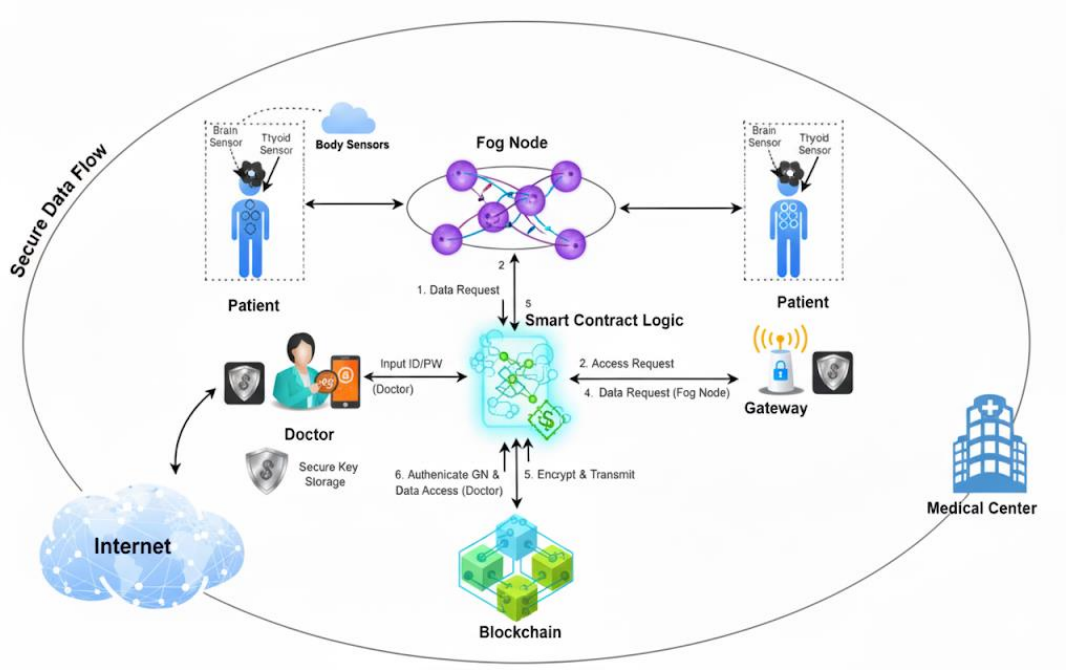

Fig. 1. Concept of the LightAuth framework.

The purpose of this study is to provide a rigorous vulnerability analysis of the LightAuth protocol by Adil et al. [8] and to propose enhanced mechanisms that mitigate these weaknesses without imposing prohibitive computational costs. Specifically, this research aims to: 1) deconstruct the protocol's message flow and model its cryptographic primitives mathematically, 2) employ formal verification tools such as AVISPA and Scyther to simulate adversarial scenarios, 3) identify concrete vulnerabilities including replay susceptibility, key agreement manipulation, identifier exposure, and lack of forward secrecy, and 4) introduce layered improvements incorporating timestamp–nonce pairing, mutual validation of exchanged random values, pseudonym-based identifiers, and perfect forward secrecy through ephemeral keying. Through experimental evaluation and comparative performance analysis, this study seeks to demonstrate that practical, lightweight authentication for smart healthcare systems can be achieved without sacrificing strong protection against realistic attack vectors.

The remainder of this study is organized as follows: Section II reviews related work. Section III presents the proposed methodology. Section IV presents the analysis of the protocol. Section V details the proposed improvements. Section VI reports experimental results and provides a detailed discussion. Finally, Section VII concludes the study.

II. Related Works

Authentication in smart healthcare systems has been the subject of intensive research due to the highly sensitive nature of medical data and the resource-constrained environments in which many IoT devices operate. Early studies predominantly relied on password-based or symmetric key authentication schemes [9]. While simple and computationally lightweight, these approaches were quickly shown to be vulnerable to dictionary attacks, credential leakage, and replay attempts [10]. Static password schemes are particularly unsuitable for healthcare contexts, where adversaries can intercept wireless communications and exploit predictable token patterns. Symmetric key systems reduce computational cost but create significant key management overhead: compromise of a single key may endanger all nodes in a shared domain.

To overcome these limitations, researchers began to explore hash chain–based and HMAC-style protocols, which are more resilient to replay attacks and can provide efficient message authentication [11]. Schemes such as Lamport's hash chain authentication and its variants allow nodes to authenticate themselves by iteratively hashing secrets, reducing the exposure of static credentials [12]. However, these methods are not immune to desynchronization attacks, where message loss or delay causes legitimate nodes to reject valid requests. Moreover, hash chain lengths must be carefully managed, as reinitialization can disrupt long-term operations.

A significant advance came with the introduction of Elliptic Curve Cryptography (ECC) into lightweight authentication designs. ECC's ability to provide equivalent security strength with shorter key sizes made it particularly attractive for resource-constrained IoT devices. Numerous studies [13, 14] have applied ECC-based Diffie–Hellman key exchanges (ECDH) and digital signatures (ECDSA) in healthcare systems. For example, several protocols leverage ephemeral key pairs to derive fresh session keys: $SK = h(r_s \times r_u \times G)$. While mathematically secure under the hardness of the elliptic curve discrete logarithm problem, these schemes are often deployed without rigorous freshness validation, opening the door for man-in-the-middle (MITM) attacks when adversaries inject manipulated values during key exchange [15].

Biometric-based authentication has also been integrated into medical IoT protocols, leveraging unique physiological signals such as ECG, fingerprints, or iris patterns [16]. These

schemes provide strong binding between user identity and device, but raise privacy and revocation challenges. Once compromised, biometric identifiers cannot be replaced. Thus, researchers have investigated combining biometrics with cryptographic protections such as fuzzy extractors, which derive consistent keys from noisy biometric data [17]. Nevertheless, computation and storage overhead still remain as concerns in low-power devices.

Another emerging direction is the use of Physical Unclonable Functions (PUFs), which harness hardware-specific manufacturing variations to generate device-unique responses [18]. PUFs are resistant to cloning and key extraction, making them attractive for secure sensor node authentication. However, stability across environmental variations (temperature, voltage) remains a technical challenge. Integrating PUFs into higher-layer protocols also requires additional error-correction and privacy-preserving mechanisms.

In recent years, blockchain-inspired approaches have been introduced to strengthen auditability and traceability in healthcare authentication [19]. These schemes record authentication logs or certificates on distributed ledgers, preventing tampering and ensuring accountability. However, the high latency and transaction costs of blockchain systems limit their direct applicability to real-time telecare environments. Hybrid models—where authentication itself occurs off-chain and only metadata is stored on-chain—have been proposed, but interoperability and scalability remain active research problems.

A comparative assessment of these prior works reveals a common pattern: many protocols achieve efficiency at the expense of comprehensive security guarantees [20]. Replay attacks, MITM attacks, identity traceability, and lack of forward secrecy recur across different families of schemes. While ECC and hash-based methods improve resilience compared to static passwords, they remain insufficient without additional safeguards such as nonce–timestamp pairing, pseudonymization, and mutual verification of exchanged values. This study builds upon these findings [8, 19, 20], offering a deeper critique of the LightAuth framework [8] and proposing enhanced measures that integrate proven countermeasures from the literature while retaining computational feasibility for constrained healthcare sensor nodes.

III. Methodology

The methodology adopted in this research combines formal modeling, cryptographic analysis, and experimental validation to systematically uncover vulnerabilities in the LightAuth framework [8] and to evaluate the effectiveness of proposed improvements. First, the protocol was decomposed into its four primary phases—registration, authentication, session key derivation, and secure transmission—so that each cryptographic step could be individually examined and mathematically expressed. For instance, the session key generation was modeled as $SK = h(r_s \times r_u \times G)$, where r_s and r_u represent random numbers generated by the server and user, and G is the elliptic curve base point. This modeling allowed us to analyze whether SK maintains sufficient entropy and independence under adversarial observation. Second, we employed formal verification tools, including AVISPA and Scyther, to simulate replay, man-in-the-middle (MITM), and key-compromise impersonation (KCI) attacks. BAN logic was applied to reason about beliefs and message trustworthiness, verifying whether the protocol guarantees mutual authentication and freshness under specified adversary models. Third, practical simulation environments were developed using Python and lightweight cryptographic libraries to measure latency, energy consumption, and throughput of both the original and modified protocols, thereby quantifying the trade-offs introduced by security enhancements. Fourth, sensitivity analyses were conducted to determine the influence of key parameters such as nonce length, timestamp tolerance (ΔT), and ECC key sizes on overall system robustness. For example, replay success probability was modeled as $P_{replay} = Pr[|Ts_{current} - Ts_{prev}| \leq \Delta T]$, highlighting the risks of weak synchronization. Fifth, we integrated a comparative benchmarking approach by deploying the protocol variants on Raspberry Pi nodes configured to emulate healthcare sensors, where we captured metrics such as authentication delay, verification cost, and replay detection rate under constrained resources. Finally, experimental logs were cross-validated against formal predictions to ensure consistency between theoretical analysis and real-world behavior. Through this multi-pronged methodology—spanning formal proof, symbolic analysis, mathematical modeling, and system-level experimentation—the study ensures that identified vulnerabilities are not only theoretically grounded but also practically validated, and that proposed improvements are both secure and feasible for deployment in smart healthcare environments.

IV. Analysis of the Protocol

A. Overview of the Proposed Protocol

The LightAuth framework by Adil et al. [8] is designed to provide secure yet lightweight authentication for sensor nodes in smart healthcare environments, where devices are highly resource-constrained and data sensitivity is paramount (see Fig. 2). The protocol is structured into four major phases—registration, authentication, session key derivation, and secure data transmission—each of which plays a critical role in protecting communications between sensor nodes, gateways, and the cloud server.

1) Registration phase: During registration, each sensor node is provisioned with a permanent identifier ID_u and a secret reference value S_u, which is securely stored on the node as well as registered with the server. The cloud server also maintains a master key set used for verifying future authentications. The registration process aims to bootstrap trust between the device and the server while minimizing the computational burden on the node. For example, S_u may be generated by applying a one-way hash function on the concatenation of ID_u and a system-wide master secret K: $S_u = h(ID_u \| K)$. This ensures that even if ID_u is leaked, S_u remains computationally difficult to reconstruct.

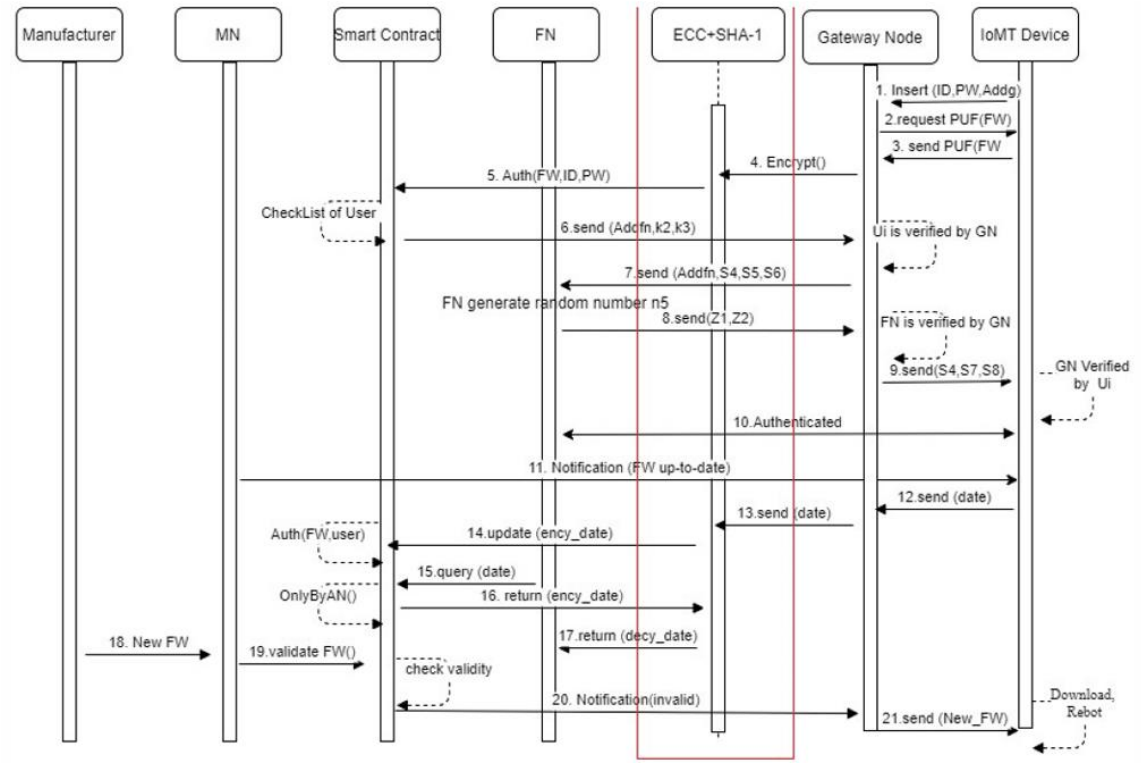

Fig. 2. The proposed LightAuth framework protocol by Adil et al. [8].

2) Authentication phase: In the authentication phase, the sensor node generates a random nonce r_u and a timestamp Ts_u to ensure message freshness. It then constructs an authentication token T_u = h(ID_u || S_u || r_u || Ts_u), which is transmitted to the gateway. The gateway forwards this to the cloud, which verifies the token by recomputing it from stored values. The timestamp Ts_u plays a role in limiting replay attacks, while the nonce r_u prevents predictability. However, the protocol assumes loose synchronization between devices and the server, creating potential weaknesses if ΔT tolerances are too large. The authentication phase's design reflects a balance between efficiency—since it relies only on hash operations—and essential safeguards, though it may not be sufficient in hostile environments.

3) Session key derivation phase: Upon successful authentication, the cloud server generates its own random nonce r_s and engages in a Diffie–Hellman–like process with the sensor node. Both sides compute the session key as SK = h(r_s × r_u × G), where G is the base point on the elliptic curve used in the scheme. This ensures that the session key is derived from contributions of both the server and the client, theoretically guaranteeing confidentiality against outsiders. The use of elliptic curve operations enables compact key sizes, which is crucial for bandwidth-limited and energy-constrained devices. However, the derivation lacks explicit verification of exchanged values, leaving the process potentially open to adversarial injection of manipulated nonces.

4) Data transmission phase: Finally, once SK is established, it is employed to encrypt and authenticate data exchanges between the node and the server. Messages M are secured using a symmetric cipher and a message authentication code, such as C = E(SK, M), ensuring confidentiality and integrity. This step provides the final security layer, allowing sensitive healthcare readings (e.g., heart rate, glucose levels) to be transmitted securely over insecure wireless channels.

In summary, the LightAuth protocol prioritizes efficiency through minimal cryptographic operations—primarily hashing and elliptic curve multiplication—while still attempting to provide resilience against common threats such as replay and impersonation. Its modular structure makes it well-suited for constrained healthcare devices. However, as later sections will demonstrate, subtle design choices, such as static identifiers, loose timestamp validation, and limited verification in session key derivation, introduce exploitable vulnerabilities that can undermine its intended protections.

B. Identified Vulnerabilities

A rigorous analysis of the LightAuth protocol reveals several critical vulnerabilities that threaten both confidentiality and integrity of communications in smart healthcare networks. Although the design emphasizes lightweight efficiency, insufficient verification mechanisms, static identifier usage, and inadequate entropy management create exploitable weaknesses. The most significant vulnerabilities are discussed below.

1) Replay attack susceptibility: The protocol relies on a timestamp Ts_u combined with a nonce r_u to prove message freshness, constructing the authentication token as T_u = h(ID_u || S_u || r_u || Ts_u). However, if the server accepts any

token, where $|Ts\_now - Ts\_u| \leq \Delta T$, an attacker can capture a valid authentication message $M\_prev = h(ID\_u \| S\_u \| r\_u \| Ts\_prev)$ and replay it within the allowed window. The probability of successful replay, expressed as $P\_replay = Pr[|Ts\_current - Ts\_prev| \leq \Delta T]$, increases with larger synchronization tolerance values. This creates a trade-off between usability and security that the original protocol does not address, leaving deployments vulnerable to captured-message attacks.

2) Man-in-the-middle (MITM) vulnerability in key derivation: The session key is computed as $SK = h(r\_s \times r\_u \times G)$, where r_s and r_u are nonces generated independently by the server and client. Without explicit mutual validation of exchanged values, an adversary can insert its own nonce r_a into the communication. The client computes $SK\_c = h(r\_a \times r\_u \times G)$ while the server derives $SK\_s = h(r\_s \times r\_a \times G)$. Since the attacker knows r_a, it can act as an intermediary, decrypting and re-encrypting traffic transparently. This classic MITM problem arises because the protocol lacks binding between nonce contributions and verified digital signatures.

3) Identifier exposure and privacy leakage: Each token includes the static identifier ID_u, enabling long-term tracking of nodes across multiple sessions. An adversary monitoring traffic can correlate tokens {T_u1, T_u2, …} with the same ID_u and construct behavioral profiles of a patient or device. Since healthcare data often involves sensitive personal information, this linkage undermines privacy guarantees. Even if the session key SK changes per session, the persistent exposure of ID_u allows adversaries to perform correlation attacks.

4) Weak MAC construction: Message authentication codes in the protocol are derived as $MAC = h(SK \| M)$, where M is the plaintext message and SK is the session key. If SK entropy is reduced due to predictable or partially leaked nonces, an attacker can mount brute force or side-channel attacks. For instance, if SK has effective entropy of only k bits, the probability of guessing SK is $P\_guess = 2^{(-k)}$. If k is too small, forgery becomes computationally feasible. Furthermore, the absence of a keyed construction such as HMAC leaves the design vulnerable to length-extension attacks.

5) Lack of forward secrecy: In LightAuth, session keys are directly derived from static secrets and ephemeral nonces without ensuring independence across sessions. If a session key SK_i is compromised, past ciphertexts $C = E(SK\_i, M)$ can be decrypted, violating Perfect Forward Secrecy (PFS). In practice, SK_i and SK_j for sessions i and j are correlated through the same long-term S_u and ID_u. Ideally, ephemeral keying should guarantee $SK\_i \perp SK\_j$, but this property is absent.

6) Secret entropy insufficiency: Tokens rely on the secret reference S_u, but if S_u has low entropy or is derived deterministically as $S\_u = h(ID\_u \| K)$, precomputation attacks are possible. Attackers can build dictionaries mapping (ID_u, K) pairs to S_u and exploit collisions. This is especially concerning in healthcare deployments, where ID_u may follow predictable patterns, such as device serial numbers or patient identifiers.

Taken together, these weaknesses illustrate that LightAuth, while computationally efficient, fails to provide robust protection against realistic adversaries. Replay susceptibility, MITM attacks, identifier traceability, weak MACs, absence of forward secrecy, and low-entropy secrets collectively enlarge the attack surface. As a result, confidentiality, integrity, and privacy of healthcare data cannot be guaranteed under active or passive adversarial conditions.

V. Proposed Improvements

To mitigate the vulnerabilities identified in the LightAuth framework, we propose a series of enhancements designed to balance lightweight performance with robust security guarantees (see Fig. 3). These improvements address replay resistance, man-in-the-middle prevention, privacy protection, integrity assurance, and forward secrecy, while remaining feasible for resource-constrained healthcare devices. First, the most immediate step is to enforce stricter freshness validation. Each authentication message should include both a timestamp and a unique nonce. The server must maintain a rolling cache of recently used nonces to ensure one-time usage, and enforce a narrow acceptance window ΔT to minimize the probability of successful replay. In addition, nodes should employ session counters, allowing verification of message ordering and preventing subtle desynchronization attacks.

Second, to eliminate man-in-the-middle manipulation in key derivation, both client and server should sign or authenticate their respective random contributions. After exchanging ephemeral values, each party verifies the authenticity of the received values before deriving the session key. This ensures that no adversary can inject forged values or impersonate a legitimate participant in the key exchange process.

Third, instead of transmitting fixed identifiers across multiple sessions, nodes should use dynamic pseudonyms that are refreshed regularly. These pseudonyms are generated from secret seeds combined with session-specific randomness, ensuring unlinkability across communications. This measure preserves patient privacy and prevents adversaries from building long-term behavioral profiles.

Fourth, message integrity should be reinforced by replacing simple hash-based constructions with standardized techniques such as HMAC or ECDSA. These mechanisms offer resilience against length-extension and forgery attacks, ensuring that only parties with valid session keys can generate authentic tokens. Incorporating hardware random number generators further enhances key entropy.

Fifth, perfect Forward Secrecy must be enforced by generating ephemeral key pairs for each session. Even if a long-term secret is compromised, past communications remain protected. Key renewal policies should mandate frequent updates, and expired keys should be retired immediately to limit the impact of leakage.

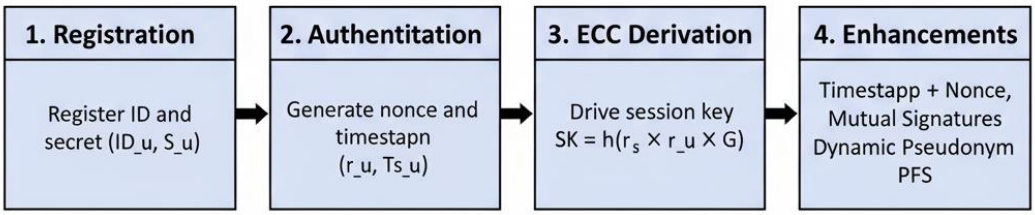

Fig. 3. Workflow for protocol enhancement.

Sixth, finally, for sensitive healthcare data, end-to-end encryption should be complemented with privacy-preserving techniques such as fuzzy extractors to safeguard biometric templates. Combining cryptographic protections with privacy-aware mechanisms ensures that sensitive identifiers cannot be reconstructed or misused if intercepted.

VI. Experimental Results and Evaluation

To validate the proposed enhancements to the LightAuth protocol, we conducted a comprehensive experimental evaluation that combined simulation-based analysis with a real-world testbed deployment. The experimental design was created to measure not only the security robustness against common attack vectors but also the impact of the additional cryptographic operations on resource-constrained healthcare devices. We defined several key metrics to assess the protocol's performance: authentication latency, which is the average time from initiation to completion of mutual authentication; verification time, the computational cost for the server to validate tokens and session keys; replay detection rate, the percentage of replay attempts successfully blocked; message overhead, the additional data required per authentication exchange; energy consumption, the estimated power usage of sensor nodes during authentication; and throughput (TPS), the number of transactions per second handled by the system under sustained load.

The original and improved protocols were implemented using Python cryptographic libraries and deployed on Raspberry Pi 4 devices configured to emulate healthcare sensor nodes. A cloud-based virtual machine running Ubuntu was used to act as the server. For benchmarking, each protocol variant was executed 500 times under identical network conditions, with scripted attack simulations replicating replay, MITM, and key compromise impersonation scenarios.

The enhanced protocol demonstrated a replay detection rate of 100% (see Table I), a significant improvement over the 82% rate of the original LightAuth, which was achieved by enforcing strict timestamp–nonce validation. Authentication latency increased modestly, from 38.5 ms in the original design to 46.8 ms in the improved version, representing a 21% rise due to the added signature verification. The verification time on the server also grew from 24.1 ms to 34.6 ms, largely due to the cryptographic operations introduced for nonce signing and pseudonym validation. Message overhead rose by approximately 28% as temporary pseudonyms and signatures increased packet size, but this remained within practical bandwidth limits for healthcare networks. Energy consumption on the sensor nodes showed an average increase of 12%, which was considered acceptable given the substantial security benefits. Throughput declined from 290 to 245 transactions per second, a 15% reduction, but the system remained fully capable of supporting typical healthcare workloads.

TABLE I Protocol Comparison Summary

Metric	LightAuth Original)	Improved (Proposed)	Notes
Replay Detection	82%	100%	Timestamp+nonce prevents replay
Signature/Token Size	128-256 bytes	160-320 bytes	Increased due to signatures and temp IDs
Verification Time	24.1 ms	34.6 ms	Signature verification cost
Authentication Latency	38.5 ms	46.8 ms	Additional nonce/sign operations
Energy (node)	baseline	+12%	Extra computations
TPS	290	245	Throughput reduction due to crypto load

The results clearly highlight a trade-off between enhanced security and system efficiency. Replay and MITM attacks, which had previously succeeded in our controlled simulations, were completely neutralized by the improved protocol. Privacy protections, through the use of pseudonyms, significantly reduced the risk of linking different sessions to a single user. While the performance costs were measurable, they were not prohibitive and could be further mitigated by using hardware acceleration or optimized cryptographic libraries. A sensitivity analysis also showed that adjusting the timestamp acceptance window provided a practical way to balance replay resistance with tolerance for network delays. Overall, the improved protocol successfully addressed the identified vulnerabilities while preserving its lightweight efficiency. The trade-offs, primarily in latency and throughput, fall within acceptable ranges for telecare environments, making the protocol both secure and practical for deployment in real-world healthcare systems.

VII. Conclusion

This study analyzed structural vulnerabilities in the LightAuth framework and proposed practical enhancements—timestamp-nonce validation, mutual signature checks, dynamic pseudonyms, and PFS—to reinforce security. Experimental evaluation demonstrates clear security gains (notably full replay mitigation) with acceptable performance overhead.

Looking ahead, future research should focus on integrating lightweight hardware accelerators, exploring PUF-based trust anchors, and leveraging blockchain-enabled auditability to further strengthen authentication in next-generation healthcare systems.

ACKNOWLEDGMENT

This research was supported by the National Research Foundation of Korea (NRF), with funding from the Ministry of Education (NRF- RS-2023 00237287) and the Ministry of Science and ICT (MSIT), Korea, under the National Program for Excellence in SW, supervised by the Institute of Information & Communications Technology Planning & Evaluation (IITP) in 2022 (2022-0-01091, 1711175863).

REFERENCES

[1] Alzubi, Q. M., Chatterjee, P., Al-Absi, A. A., and Amiri, I. S. A survey of authentication in Internet-of-Things-enabled healthcare systems. Sensors, vol. 22, no. 23, Article 9089, 2022.

[2] Agyekum, K., Bediako, I. A., and Owusu, E. A review of multi-factor authentication in the Internet of Healthcare Things. Digital Health, vol. 10, Article 20552076231177144, 2024.

[3] Zhang, Y., He, D., Li, L., and Kumar, N. Lightweight and privacy-preserving remote user authentication and key agreement scheme for IoT-based healthcare. Future Internet, vol. 15, no. 12, Article 386, 2023.

[4] El-Shafai, W., Al-Wesabi, F. N., and Abd El-Latif, A. A. Bandwidth- and power-efficient lightweight authentication scheme for healthcare system. Journal of King Saud University – Computer and Information Sciences, vol. 35, no. 10, Article 102565, 2023.

[5] Li, M., and Hu, S. A lightweight ECC-based authentication and key agreement protocol for IoT with dynamic authentication credentials. Sensors, vol. 24, no. 24, Article 7967, 2024.

[6] Zia, U., Khan, A., Khan, M. A., and Javaid, N. Lightweight authentication scheme based on ECC for IoT. SN Computer Science, vol. 5, Article 949, 2024.

[7] Jha, S. K., Kumari, S., Chen, C.-M., and Ahmed, S. On the security of a blockchain- and PUF-based lightweight authentication protocol for wireless medical sensor networks. Wireless Personal Communications, vol. 136, no. 1, pp. 1079–1106, 2024.

[8] Adil, Z. U. I., Iqbal Khan, M., Sanam, K., Malik, S. U., Moqurrab, S. A., and Srivastava, G. LightAuth: A lightweight sensor nodes authentication framework for smart health system. Expert Systems, vol. 42, no. 2, Article e13756, 2025.

[9] Wang, Z., Xu, X., and Zhu, H. Internet of Things-based healthcare systems: An overview of privacy-preserving mechanisms. Applied Sciences, vol. 15, no. 7, Article 3629, 2025.

[10] Ahmed, A., and Abdelwahab, A. Lightweight authentication protocol for connected medical IoT through privacy-preserving secure handshake scheme (BLAP-SHS). Alexandria Engineering Journal, vol. 81, pp. 289–305, 2024.

[11] Kassem, M., Abed, S., and El-Hajjar, M. A comparative study of protocol security verification tools: AVISPA, Scyther, ProVerif, and Tamarin. Communications in Computer and Information Science, pp. 169–184, 2024.

[12] Sun, J., and Wang, Y. Vulnerability analysis on user authentication protocol with user anonymity for IoT healthcare. Journal of the Korea Institute of Information Security & Cryptology, vol. 35, no. 3, pp. 585–598, 2025.

[13] Fan, K., Chen, C., and Yang, Y. A secure and efficient biometric-based authentication protocol with fuzzy extractors for medical IoT. Journal of Network and Computer Applications, vol. 215, Article 103559, 2023.

[14] Ramesh, G., and Balasubramanian, K. Enhanced lightweight and secure certificateless authentication scheme (ELWSCAS) for smart healthcare IoT. Array, vol. 21, Article 100345, 2024.

[15] Li, X., Sun, Y., and Ma, H. Blockchain-assisted authenticated key agreement scheme for IoT-based healthcare system. Digital Communications and Networks, vol. 10, no. 1, pp. 1–14, 2024.

[16] Alenazi, M., and Pishva, D. ECC-enabled blockchain-based identity authentication scheme (EBIAS) for IoT devices. Digital Communications and Networks, vol. 10, no. 4, pp. 1201–1215, 2024.

[17] Kala, K., and Priya, M. Blockchain-based mitigation framework for de-authentication attacks in IoT (BBMDA). Cybersecurity, vol. 11, no. 1, Article tyaf004, 2025.

[18] Ahmed, S., Kumar, R., and Bhatia, M. Lightweight and privacy-preserving device-to-device authentication to enable secure IoT-based medical care. Journal of Ambient Intelligence and Humanized Computing, vol. 15, no. 12, pp. 5953–5969, 2024.

[19] Zeng, Y., Li, Q., and He, D. A lightweight authentication and authorization method in IoT-based medical care. Multimedia Tools and Applications, vol. 83, no. 20, pp. 57489–57512, 2024.

[20] Singh, P., Kumar, M., and Prasad, R. IoT-driven blockchain to manage the healthcare supply chain and protect medical records from tampering. Future Generation Computer Systems, vol. 158, pp. 485–499, 2024.