Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.067641

ARTICLE

LSAP-IoHT: Lightweight Secure Authentication Protocol for the Internet of Healthcare Things

Marwa Ahmim[1], Nour Ouafi[1], Insaf Ullah[2,*], Ahmed Ahmim[3], Djalel Chefrour[3] and Reham Almukhlifi[4]

[1]Networks and Systems Laboratory, Department of Computer Science, Badji Mokhtar University, Annaba, 23000, Algeria
[2]Institute for Analytics and Data Science, University of Essex, Colchester, CO4 3SQ, UK
[3]Department of Computer Science, University of Souk-Ahras, Souk-Ahras, 41000, Algeria
[4]Cybersecurity Department, College of Computer Science and Engineering, Taibah University, Medina, 42353, Saudi Arabia

*Corresponding Author: Insaf Ullah. Email: Insaf.ullah@essex.ac.uk

Received: 08 May 2025; Accepted: 15 July 2025; Published: 23 October 2025

ABSTRACT: The Internet of Healthcare Things (IoHT) marks a significant breakthrough in modern medicine by enabling a new era of healthcare services. IoHT supports real-time, continuous, and personalized monitoring of patients' health conditions. However, the security of sensitive data exchanged within IoHT remains a major concern, as the widespread connectivity and wireless nature of these systems expose them to various vulnerabilities. Potential threats include unauthorized access, device compromise, data breaches, and data alteration, all of which may compromise the confidentiality and integrity of patient information. In this paper, we provide an in-depth security analysis of LAP-IoHT, an authentication scheme designed to ensure secure communication in Internet of Healthcare Things environments. This analysis reveals several vulnerabilities in the LAP-IoHT protocol, namely its inability to resist various attacks, including user impersonation and privileged insider threats. To address these issues, we introduce LSAP-IoHT, a secure and lightweight authentication protocol for the Internet of Healthcare Things (IoHT). This protocol leverages Elliptic Curve Cryptography (ECC), Physical Unclonable Functions (PUFs), and Three-Factor Authentication (3FA). Its security is validated through both informal analysis and formal verification using the Scyther tool and the Real-Or-Random (ROR) model. The results demonstrate strong resistance against man-in-the-middle (MITM) attacks, replay attacks, identity spoofing, stolen smart device attacks, and insider threats, while maintaining low computational and communication costs.

KEYWORDS: Internet of healthcare things (IoHT); authentication protocol; cryptanalysis; attacks

1 Introduction

The Internet of Health Things (IoHT) represents a major innovation that integrates information technologies with medical care, forming an intelligent ecosystem of connected devices dedicated to health monitoring and management [1,2]. It encompasses a broad spectrum of technologies and application domains, ranging from biometric sensors and wearable devices to remote health monitoring platforms and connected medical equipment. IoHT is actively deployed in scenarios such as chronic disease management, emergency response, elderly care, postoperative recovery, and telemedicine. As illustrated in Fig. 1, these systems interact through wireless communication technologies, including Wi-Fi, Bluetooth, and cellular networks to collect, transmit, and analyze health data in real time [3,4]. By enabling continuous tracking of

vital parameters like heart rate, blood pressure, glucose levels, and respiratory signals, IoHT supports mobile health (mHealth), intelligent drug delivery, and personalized healthcare services.

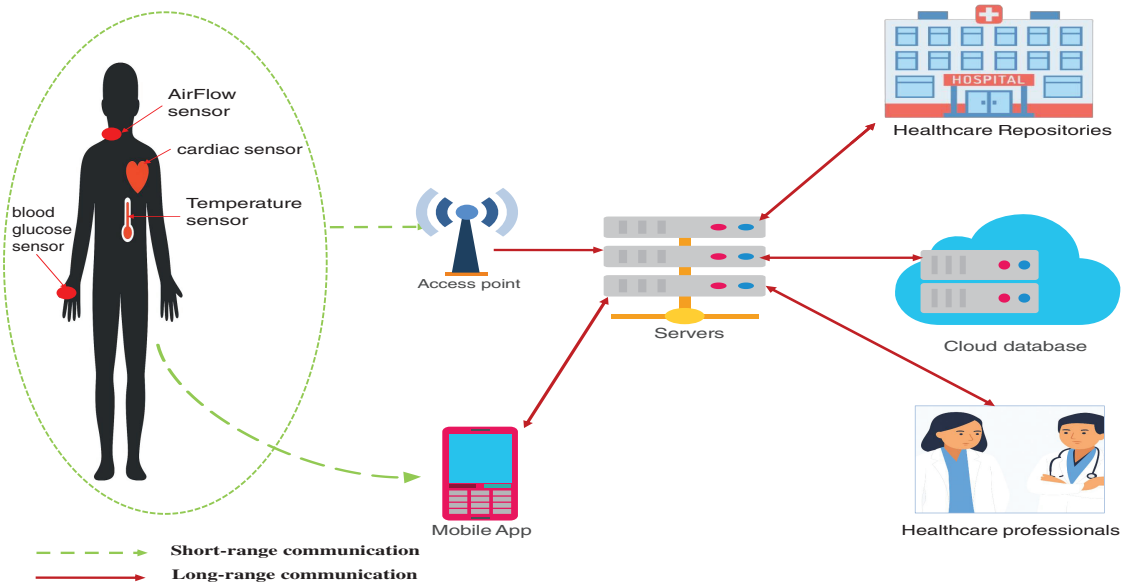


Figure 1: Internet of health things

However, the sensitive nature of health data, combined with the heterogeneity of interconnected devices, introduces substantial challenges regarding security and privacy [5–7]. Adversaries may exploit vulnerabilities through cyberattacks such as data interception, unauthorized access, and device tampering, all of which pose serious risks. In the absence of robust security mechanisms, sensitive medical information may be exposed to privacy breaches, thereby compromising patient confidentiality. Furthermore, compromised device integrity can result in erroneous diagnoses, inappropriate treatments, or even life threatening outcomes for patients [8,9].

1.1 Motivation and Contribution

With the increasing reliance on Internet of Health Things (IoHT) systems, securing sensitive medical data against cyber threats has become a critical concern, particularly given the potential consequences for patient safety. Although Chen et al. [10] introduced LAT-IoHT, an authentication protocol combining three-factor authentication, asymmetric encryption, and biometric data, our cryptanalysis has revealed significant security flaws. Specifically, LAT-IoHT is vulnerable to smart card theft, privileged insider attacks, and user impersonation.

To address these limitations, we propose LSAP-IoHT, a lightweight and secure authentication protocol tailored for IoHT environments, which offers the following advantages:

- Our scheme combines Elliptic Curve Cryptography (ECC), Three-Factor Authentication (3FA), and hardware-level protection based on Physical Unclonable Functions (PUFs), enhancing resistance to duplication and physical attacks.
- Through formal and informal analysis using the Real-Or-Random (ROR) model and the Scyther tool, our scheme is shown to ensure mutual authentication, session key confidentiality, and strong resistance to a wide range of attacks, including stolen smart device attacks, identity spoofing, replay attacks, man-in-the-middle (MITM) attacks, and privileged insider threats.

- Our scheme achieves these security properties while maintaining low computational and communication overhead, making it particularly suitable for deployment in resource constrained IoHT environments.
- Comparative evaluation with related authentication protocols confirms that our scheme provides enhanced security guarantees with low computational complexity and communication cost.

1.2 Paper Outline

This article is structured as follows. Section 2 discusses existing authentication mechanisms developed for IoHT, emphasizing their main features and limitations. Section 3 investigates the authentication scheme introduced by Chen et al. [10], highlighting the cryptographic vulnerabilities identified through a detailed security evaluation. Section 5 presents our proposed authentication scheme, referred to as LSAP-IoHT. Section 6 describes the results of the security verification conducted using the Scyther tool and the Real-Or-Random (ROR) model. Section 7 provides a comparative assessment of LSAP-IoHT and other protocols designed for IoHT environments. Finally, Section 8 concludes the paper by summarizing the main contributions.

2 Related Works

This section reviews key literature on authentication protocols for the Internet of Health Things (IoHT), focusing on foundational contributions, recent advancements, and emerging trends relevant to this study.

Kumar et al. [11] introduced RAPCHI, a robust protocol for cloud-based connected health infrastructures. By employing digital signatures, it secures authentication and key agreement among patients, cloud servers, and doctors. RAPCHI resists man-in-the-middle and replay attacks, as validated through AVISPA simulation.

Aghili et al. [12] developed an access control and ownership transfer protocol for IoHT, addressing vulnerabilities in the ZZTL protocol [13] related to user traceability, denial-of-service (DoS), insider threats, and desynchronization attacks. Their architecture involves users, medical servers, and patients, ensuring authentication, confidentiality, and seamless privilege transfer. Focusing on lightweight designs for sensor networks,

Soni et al. [14] developed an authentication protocol using smart cards, improving upon the weaknesses identified in Challa et al.'s work [15] by integrating a three-factor approach suited to remote patient monitoring. To accommodate heterogeneous IoHT environments and enhance efficiency, Ullah et al. [16] proposed a certificate-less authentication scheme that employs hyperelliptic curve cryptography (HECC) with 80-bit keys. This construction is particularly appropriate for resource-constrained wireless body area networks (WBANs), offering robust protection against common cyber threats while reducing computational overhead.

Rabie et al. [17] proposed a distributed, privacy-preserving protocol for certificate-less medical sensor networks. Using XOR operations, it aggregates signatures from sensor nodes, thereby reducing centralization. The system includes wearable sensors, zonal nodes, and medical servers, and it prioritizes data privacy during transmission. Xie et al. [18] introduced CasCP, an enhanced certificate-less authentication scheme for WBANs, which addresses weaknesses in Ji et al.'s protocol [19]. It ensures conditional privacy preservation and improves security. Thakur et al. [20] proposed an authentication scheme for WMSN-based medical systems that counters identity spoofing and password guessing attacks. Building on the protocol of Servati and Safkhani [21], it retains a five-phase structure while enhancing security properties. Li et al. [22] developed a three-factor authentication protocol for WMSNs, improving upon the scheme of

Amin and Biswas scheme [23] after identifying several vulnerabilities. Based on ECC, the protocol ensures secure transmission of sensitive medical data. Sureshkumar et al. [24] proposed a lightweight ECC-based authentication scheme for WMSNs. Its correctness was formally verified using Burrows-Abadi-Needham (BAN) logic.

In [25], the authors propose a secure communication approach for IoT networks based on key management and authentication. Their method improves data protection, energy efficiency, and overall network performance. In [26], the authors propose an improved lightweight authentication protocol to address security and privacy issues in IoT-based smart healthcare systems. The protocol ensures mutual authentication and secure session key establishment between doctors, gateways, and sensor nodes. Security analysis confirms its efficiency and resistance to various attacks, including identity anonymity and untraceability.

Finally, Chen et al. [10] introduced LAP-IoHT, a three-factor authentication protocol that combines asymmetric encryption and biometric data to ensure confidentiality. It authenticates users and sensors through a gateway, establishes a shared session key, and encrypts biometric features to preserve user anonymity. Its security was analyzed using the Real-or-Random (ROR) model.

Table 1 summarizes the reviewed works by emphasizing their cryptographic features, verification tools, and protocol phases. It also compares them by highlighting their advantages and limitations. In the next section, we detail the workings of LAP-IoHT in particular and reveal its vulnerabilities.

Table 1: An evaluation of IoMT authentication methods based on publication year, cryptographic approaches, identified weaknesses, and distinguishing features

Ref.	Year	Entities in system model	Cryptographic system used	Verification tool	Protocol phases	Advantages/Limitations
[11]	2022	4 “Patient, Doctor, Cloud Server, Body Sensor”	ECC	– ROM – AVISPA	– Initialization – Patient/Doctor Registration – Login and Key Agreement – Password Change	– Resists insider, man-in-the-middle, replay attacks
[12]	2019	3 “User Group, Medical Server, Patient Group”	Symmetric Encryption	ProVerif	– Setup – Registration – Login – Authentication and Key Agreement – Ownership Transfer	– User untraceability – Resists DoS, desynchronization attacks – / – Cannot provide perfect forward security – Cannot resist malicious sensor or server impersonation attacks
[14]	2019	3 “Trusted Authority, User, Sensor Node”	ECC	– AVISPA – BAN	– Initialization – Sensor/User Registration – Login -Authentication – Password Change and Biometric Update – User Revocation and Re-registration – Dynamic Sensor Node Addition	– Mutual Authentication – Resists DoS attacks – / – Cannot provide perfect forward security – Cannot resist sensor node capture attack

(Continued)

Table 1 (continued)

Ref.	Year	Entities in system model	Cryptographic system used	Verification tool	Protocol phases	Advantages/Limitations
[16]	2023	3 "Client, Application Provider (AP), Key Generation Center (KGC)"	– ECC – HECC	ROR	– Setup – Pseudo Identity Generation – Mutual Authentication and Key Management – Password Change	–Improved efficiency in computation and communication
[18]	2019	3 "Client, Application Provider (AP), Key Generation Center (KGC)"	ECC	ROM	– System Initialization – Pseudo Identity Generation – Message Signing – Authentication – Password Change	– Resists various security attacks – / – Limitations in certain attack scenarios
[20]	2023	4 "Gateway, Sensor, User, System Administrator"	ECC	– BAN – ROR – -Scyther	– Initialization – Gateway and Sensor Node Registration – User Registration – Login, Authentication, Key Agreement – Password and Biometric Update	– Better security features compared to earlier schemes
[22]	2019	3 "Gateway, Sensor, User"	ECC	ProVerif	– System Setup – Medical Professional and Patient Registration – Login and Authentication – Password Change – Revocation and Re-registration	– Unable to ensure sensor node security – / – Cannot resist key security, impersonation, and mobile device loss attacks
[24]	2019	3 "Gateway, Sensor, User"	ECC	BAN	– System Setup – Sensor and Gateway Node Registration – User-Login – Gateway-Sensor Authentication – Password Renewal – Sensor Node Inclusion	– Resists traceability, de-synchronization, integrity contra-diction attacks – / – Cannot resist user imperson-ation attacks – Cannot provide perfect forward secrecy
[10]	2022	3 "Users, Gateway, Wearable Sensors"	-Asymmetric and Symmetric Encryption -ECC	ROR	– User Registration – Sensor Registration – Login and Authentication	– Resistant to known security attacks – Improved computational cost and time efficiency – / – Cannot resist DoS attack – Cannot resist message modifica-tion attack

3 LAP-IoHT Cryptanalysis

This section provides an overview of the LAP-IoHT protocol and highlights its security weaknesses, despite its original claim of being secure in the study by Chen et al. [10].

3.1 Review of LAP-IoHT

The LAP-IoHT protocol [10] is a lightweight authentication scheme designed for the Internet of Health Things (IoHT). It employs three-factor authentication, asymmetric encryption, and biometric verification to secure the transmission of sensitive health information. The protocol is structured into two main phases: the registration phase (covering both user and sensor registration) and the authentication phase. In what follows, we denote the user, gateway, and sensor by U_a, GW_b, and SN_c, respectively. The user and sensor identities are represented as ID_a and ID_c, and the user's password is noted PW_a. Biometric data is referred to as *bio*, the random nonces used throughout the protocol are denoted r_a, r_b, r_c, and r_i and G_j represents the secret key of the gateway.

3.1.1 Registration Phase

User Registration Phase

If the user U_a wishes to be a legitimate user, this user must register with the GW_b. Messages are transmitted via a secure channel.

Step 1: The user U_a possesses ID_a, PW_a, and the biometric data Bio and chooses a random number r_i. U_a computes $HID_a = h(ID_a \parallel r_i)$, $Gen(Bio) = (\sigma_a, \tau_a)$, $HPW_a = h(PW_a \parallel \sigma_a)$, and $N = PW_a \oplus h(ID_a \parallel \sigma_a)$. U_a sends to GW_b the message $\{HID_a, HPW_a, N\}$.

Step 2: GW_b verifies if HID_a has already been registered. Then, computes $D_1 = h(HID_a \parallel N)$, $D_2 = (D_1 \parallel G_j) \oplus HPW_a$, $D_3 = D_2 \oplus N$, $D_4 = h(HID_a \parallel G_j) \oplus D_1$. GW_b sends to U_a the message $\{D_1, D_3, D_4\}$. GW_b stores (HID_a, D_1) in its database.

Step 3: U_a computes $\Omega_a = N \oplus r_i$ and $M = h(N \parallel r_i) \oplus HID_a$. U_a stores $\{D_1, D_3, D_4, \Omega_i, M\}$ in its smart card.

Sensor Registration Phase

A sensor must also be registered before it can join the network with GW_b. Messages are transmitted via a secure channel.

Step 1: The sensor SN_c sends its identity ID_c to GW_b.

Step 2: GW_c generates a random number b and computes the pseudo-identity of SN_c, $PID_c = h(ID_c \parallel b)$ and $HSID_c = h(ID_c \parallel G_j)$. Then, it calculates $SG = h(HSID_c \parallel G_j) \oplus PID_c$ and $L = ENC_{pbs}(PID_c)$. GW_b sends to SN_c the message $\{SG, L\}$. GW_b stores (SID_c, PID_c) in its database.

Step 3: SN_c stores $\{SG, L\}$ in its own memory.

3.1.2 Authentication Phase

If U_a requests a connection to a specific portable sensor SN_c, GW_b must verify the legitimacy of the user.

Step 1: U_a inserts its smart card into a computer or card reader and provides its ID_a, PW_a, and Bio. The computer computes $\sigma_a = Rep(Bio, \tau_a)$, $N = PW_a \oplus h(ID_a \parallel \sigma_a)$, and $M' = h(N \parallel r_i) \oplus HID_a$, where $r_i = N \oplus \Omega_a$. Then, it determines if M' is equal to M stored in the smart card. If so, it generates a random number r_a and a timestamp T_1 and computes $HPW_a = h(PW_a \parallel$

σ_a), $K_1 = D_3 \oplus N \oplus HPW_a$, $K_2 = K_1 \oplus r_a$, $X_{ug} = h(T_1 \| r_a \| HID_a \| K_2)$. U_a sends to GW_b the message $\{HID_a, K_2, X_{ug}, T_1\}$.

Step 2: GW_b verifies the freshness of T_1. Then, it computes $K_1 = h(D_1 \| G_j)$, $r_a = K_1 \oplus K_2$, and $X'_{ug} = h(T_1 \| r_a \| HID_a \| K_2)$. If the received X_{ug} is equal to the calculated X'_{ug}, GW_c computes a random number r_c and a timestamp T_2, $HSID_c = h(ID_c \| G_j)$, $K_3 = r_a \oplus h(HSID_c \| G_j)$, $K_4 = D_1 \oplus h(K_3 \| ID_c \| r_a)$, $K_5 = r_c \oplus h(D_1 \| r_a)$, $K_6 = K_3 \oplus PID_c$, $X_{gs} = h(T_2 \| r_a \| r_c \| ID_c \| K_5)$. GW_b sends to SN_c the message $\{K_4, K_5, K_6, X_{gs}, T_2\}$.

Step 3: SN_b verifies the freshness of T_2. Then, SN_c obtains PID_c by decrypting L using the private key, computes $K_3 = K_6 \oplus PID_c$, $r_a = K_3 \oplus SG \oplus PID_c$, $D_1 = K_4 \oplus h(K_3 \| ID_c \| r_a)$, $r_c = K_5 \oplus h(D_1 \| r_a)$, $X'_{gs} = h(T_2 \| r_a \| r_c \| ID_c \| K_5)$. If the received X_{gs} is equal to the calculated X'_{gs}, SN_b calculates a random number r_b and a timestamp T_3, $K_7 = r_b \oplus h(SG \| D_1 \| r_c)$, $K_8 = PID_c \oplus K_7$, $X_{sg} = h(T_3 \| r_c \| r_b \| K_7 \| SG)$, $X_{su} = h(r_a \| r_b \| ID_c \| D_1)$. Finally, SN_c calculates the session key $SK = h(r_a \| r_b \| r_c)$. SN_c sends to GW_b the message $\{K_8, X_{sg}, X_{su}, T_3\}$.

Step 4: GW_b verifies the freshness of T_3. Then, it computes $K_7 = PID_c \oplus K_8$, $SG = h(HSID_c \| G_j) \oplus PID_c$, $r_b = K_7 \oplus h(SG \| D_1 \| r_c)$, $X'_{sg} = h(T_3 \| r_c \| r_b \| K_7 \| SG)$. If the received X_{sg} is equal to the calculated X'_{sg}, GW_b computes a timestamp T_4, $K_9 = D_1 \oplus K_1$, $K_{10} = K_9 \oplus h(HID_a \| G_j) \oplus r_b$, $K_{11} = ID_c \oplus h(K_1 \| r_b)$, $X_{gu} = h(T_4 \| r_a \| r_c \| K_{10})$ for mutual authentication. Finally, GW_b calculates the session key $SK = h(r_a \| r_b \| r_c)$. GW_b sends to U_a the message $\{K_5, K_{10}, K_{11}, X_{gu}, X_{su}, T_4\}$.

Step 5: U_a verifies the freshness of T_4. Then, it calculates $r_b = K_1 \oplus K_{10} \oplus D_4$, $r_c = K_5 \oplus h(D_1 \| r_a)$, and calculates $X_{gu} = h(T_4 \| r_a \| r_c \| K_{10})$. If the received X_{gu} is equal to the calculated X'_{gu}, U_a calculates the session key $SK = h(r_a \| r_b \| r_c)$.

3.2 Security Weaknesses of LAP-IoHT

3.2.1 Smart Card Theft Attack

The attacker A obtains $\{D_1, D_3, D_4, \Omega_a, M\}$ stored in the stolen smart card.

3.2.2 Privileged Insider Attack

Suppose the attacker A is a privileged insider of the system. The attacker will take HPW_a and N. Knowing that $K_1 = h(D_1 \| G_j)$, the attacker can find K_1 from D_2 of the previous attack. The attacker already has D_3, so they also have D_2 since $D_3 = D_2 \oplus N$. Knowing that $D_2 = h(D_1 \| G_j) \oplus HPW_a$, by applying another $\oplus HPW_a$, the attacker obtains $h(D_1 \| G_j)$, which is K_1. Now, $K_2 = K_1 \oplus r_a$, the attacker could find K_1, so from K_2, we can say that the attacker possesses r_a.

3.2.3 User Identity Impersonation Attack

Fig. 2 illustrates the user impersonation attack where the attacker A seeks to generate a valid Message 1 containing $\{HID_a, K_2, X_{ug}, T_1\}$ in the connection and authentication phase, while knowing that the attacker already possesses $\{r_u, K_1, K_2, D_1, D_3, D_4, \Omega_a, M, HPW_a\}$. Since HID_a was sent via a public channel, they can also obtain it. We have $X_{ug} = h(T_1 \| r_a \| HID_a \| K_2)$ and the attacker has obtained all of these. Since the generated Message 1 by the attacker contains the original identification information of user U_a, the generated message will pass the verification phase. Attacker A has successfully impersonated the identity of the user. In Message 5, $K_5 = r_c \oplus h(D_1 \| r_a)$ the attacker possesses D_1 and r_a, so they can obtain r_c. $r_b = K_1 \oplus K_{10} \oplus D_4 = K_1 \oplus K_9 \oplus h(HID_a \| G_j) \oplus D_4 = K_1 \oplus D_1 \oplus K_1 \oplus h(HID_a \| G_j) \oplus h(HID_a \| G_j) \oplus D_1 \oplus r_b$ thus the attacker can also obtain r_b. As the attacker possesses r_a, r_b, and r_c, they now have the secret key $SK = h(r_a \| r_b \| r_c)$.

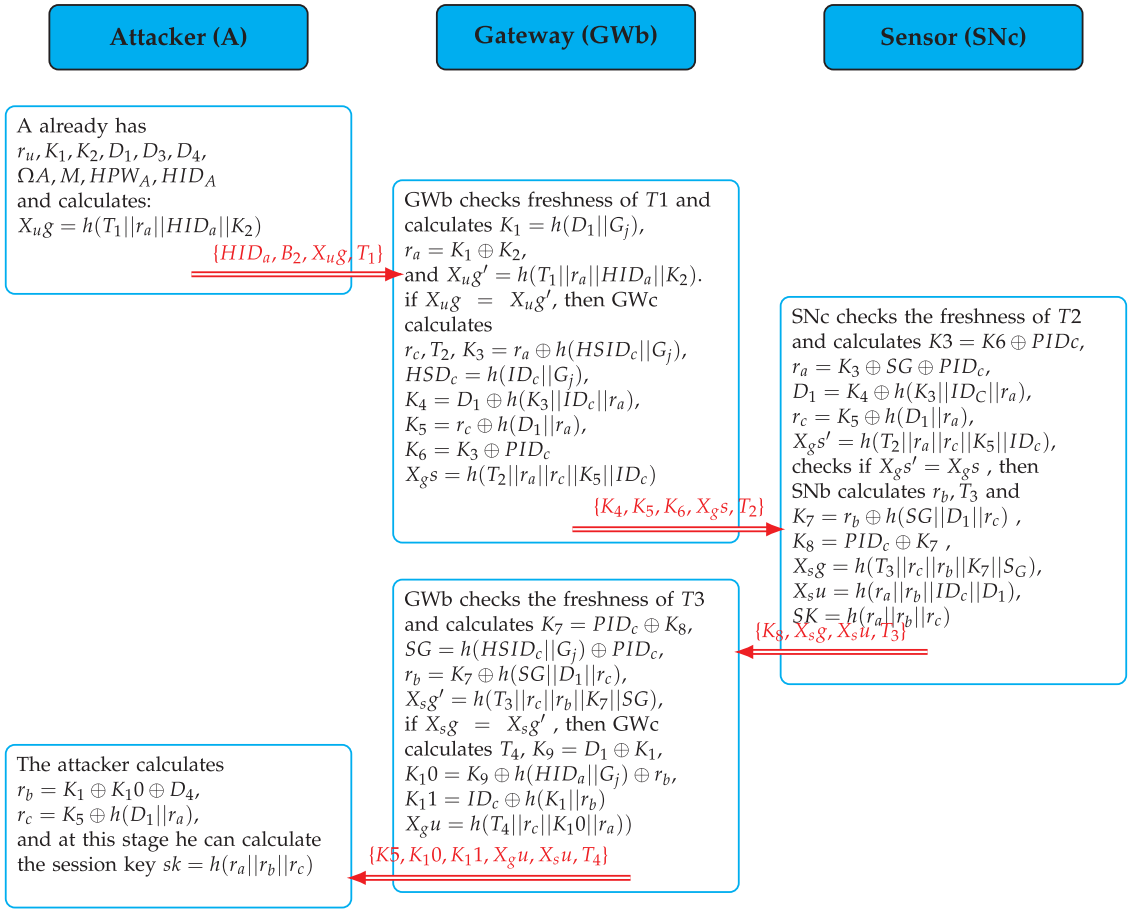


Figure 2: User identity impersonation attack

4 Preliminaries and System Model

This part presents the necessary background to enhance the clarity of the article.

4.1 Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography (ECC) is a method based on elliptic curves defined over a finite field. Consider two large prime numbers p and q; an elliptic curve $E_q(u, v)$ is defined by the equation: $y^2 = x^3 + ux + v \pmod q$ where $u, v \in \mathbb{Z}_q$, and the discriminant $\Delta = 4u^3 + 27v^2 \not\equiv 0 \pmod q$ ensures that the curve is non singular. The set of all points (x, y) satisfying the curve equation, along with the point at infinity O, form an additive group G of order p. Let P be a generator of this group. Scalar multiplication over the group is defined by repeated addition: $x \cdot P = \underbrace{P + P + \cdots + P}_{x \text{ times}}$. The group operation satisfies the property:

$x \cdot P + y \cdot P = (x + y) \cdot P \quad$ for all $x, y \in \mathbb{Z}_q$.

4.2 Physical Unclonable Function (PUF)

Due to physical variations during chip fabrication, a Physical Unclonable Function (PUF) generates a unique, non-replicable output for each input, following a challenge-response approach. Unlike conventional cryptographic systems that store fixed keys, PUFs dynamically produce device-specific secrets, eliminating the need for key storage. If a device is compromised, the internal structure of the PUF is affected, altering

its output and ensuring security without consuming memory. Given a challenge C_i and a corresponding response R_i, the relation is defined as $R_i = \text{PUF}(C_i)$. The LSAP-IoHT protocol leverages this capability by integrating PUF into the gateway, sensor nodes, and user side to protect against physical attacks.

4.3 Network Model

The proposed health system model is illustrated in Fig. 3. In this paradigm, the workflow of the proposed protocol consists of three main entities: sensors, users, and the gateway.

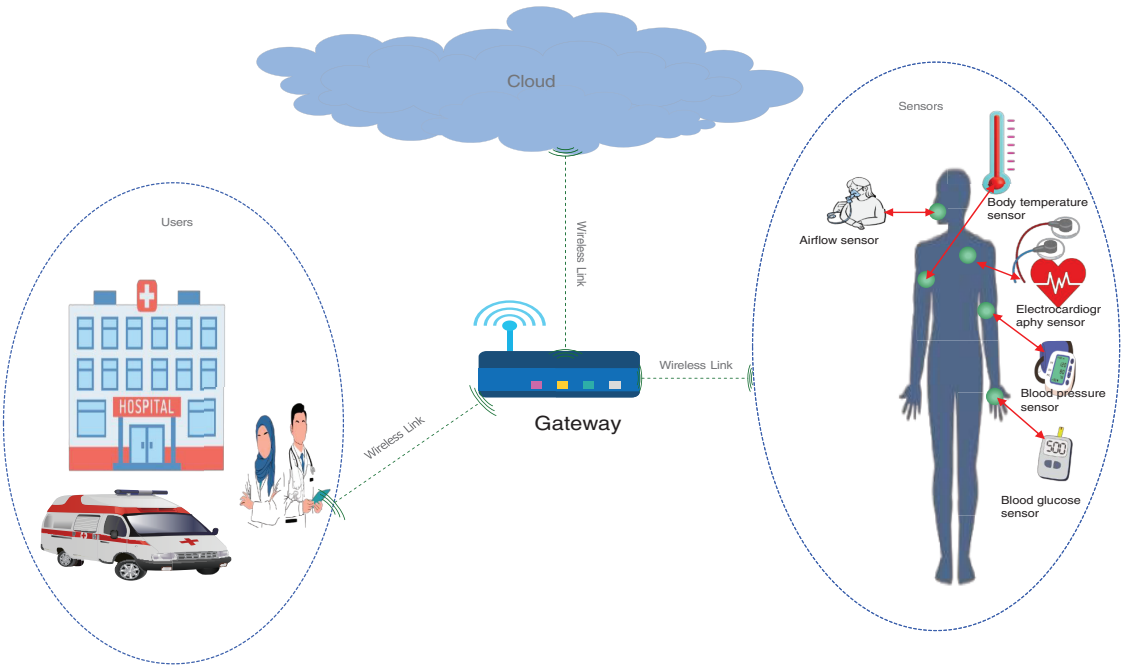


Figure 3: Proposed health system model

4.3.1 Sensors

These sensors are placed or implanted in the human body, and their main role is to collect various health-related data, such as ECG, blood pressure, blood glucose, respiratory rate, and body temperature.

4.3.2 Users

In the proposed system model, users are primarily represented as doctors who have authorized access to patients medical data. These doctors use the system to monitor and analyze patients conditions, prescribe treatments, and make informed therapeutic decisions based on the data provided by the sensors.

4.3.3 Gateway

The gateway acts as a pivot to orchestrate communication between users and sensors, serving as a trusted intermediary that facilitates transparent communication. It constitutes a main unit where information is authenticated and securely linked among the various components of the system.

4.4 Adversary Model

In Internet of Health Things (IoHT) systems, the Dolev–Yao model [27] is applied to represent hostile network conditions. This framework presumes that an attacker, labeled A, can access all exchanges over unsecured links. Moreover, A is capable of discarding transmissions, changing message content, or introducing counterfeit data into the flow of communication.

5 The Proposed LSAP-IoHT Protocol

To address the security vulnerabilities we identified in LAP-IoHT [10], we propose a new reliable and efficient authentication scheme for the IoHT called: Lightweight Secure Authentication Protocol for the IoHT (LSAP-IoHT). The LSAP-IoHT protocol ensures secure authentication between the user, the gateway, and the sensor. Fig. 4 outlines the message exchanges leading to mutual verification and the establishment of a session key.

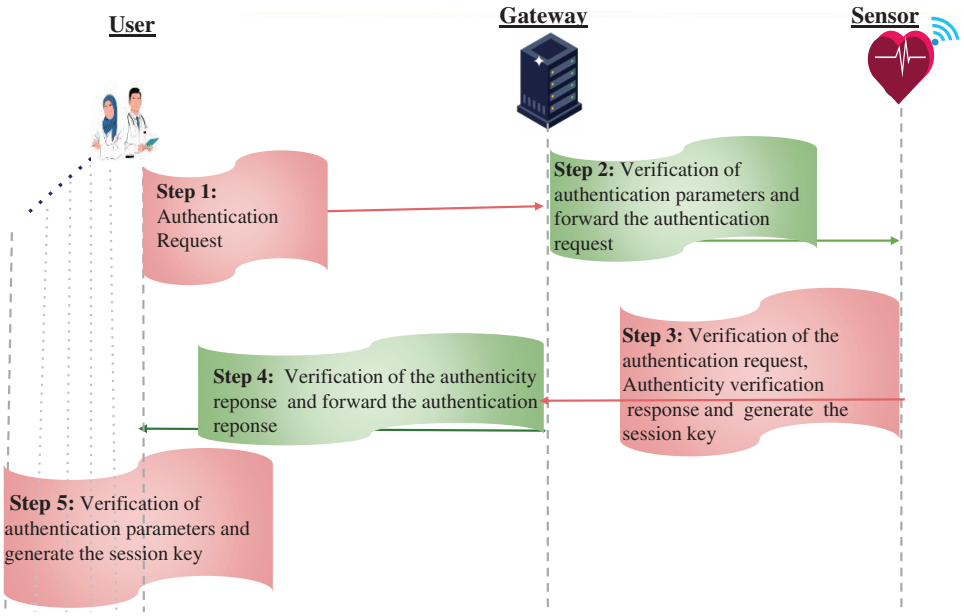


Figure 4: Sequence diagram of the authentication steps in the proposed scheme

5.1 Protocol Phases

The improved system incorporates the following four phases:

- Initiation phase
- User registration phase
- Sensor registration phase
- Authentication phase

Table 2 summarizes the different annotations used for our protocol.

Table 2: Annotations

Notations	Descriptions
U_a, GW_b, SN_c	User, gateway node, sensor node
PW_a	Password of U_a
ID_a	Identity of U_a
ID_c	Identity of SN_c
PW_c	Password of SN_c
SK_{sg}, SK_{ug}, SK_{su}	Session key
T_s	Time stamp, with $s = 1, 2, 3, 4$
N_A, N_B, N_S, NB_1, NB_2, r_u, r_s, r_g	Random numbers
$\oplus$, $\|\|$	XOR, concatenation operation
$H(.)$	Hash function
ENC/DEC	Encryption/Decryption
PUF	Physical unclonable function
ECC	Elliptic-curve cryptography

5.1.1 Initiation Phase

This phase is considered to be applied before the commencement of the protocol in a secure environment. Each entity gateway, sensor, and user is equipped with PUF (Physical Unclonable Function) technology and has a Challenge-Response based on PUF, referred to as CRP. The gateway has the CRPs of both the sensors and the users (CRPs, CRPa), while both the sensors and users possess the CRP of the gateway (CRPb). This protocol uses a lightweight ECC (Elliptic-Curve Cryptography) system for smart devices as shown in Fig. 5.

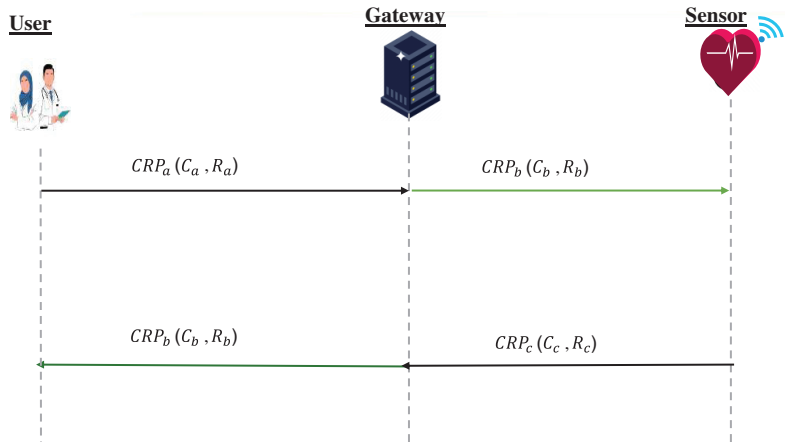


Figure 5: Initialization phase

5.1.2 User Registration Phase

For the user U_a to be recognized as a legitimate user, they must be registered with GW_b. Once this registration is completed, all exchanged messages will be transmitted via a secure channel to ensure the confidentiality and integrity of the communications as shown in Fig. 6.

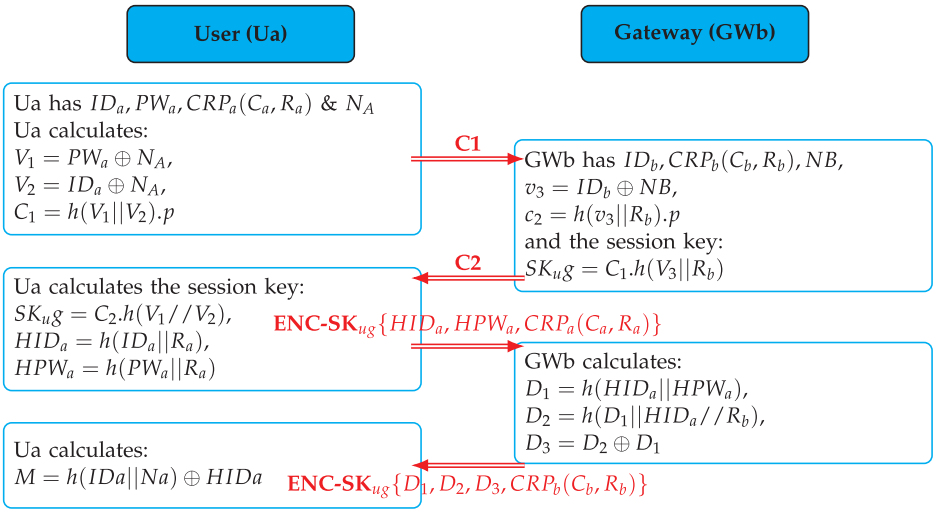


Figure 6: User registration phase

Step 1: The user U_a has ID_a, PW_a, and the PUF generates the unique challenge-response $CRP_a(C_a, R_a)$ and chooses a random number N_A. U_a calculates $V_1 = PW_a \oplus N_A$, $V_2 = ID_a \oplus N_A$, $C_1 = h(V_1 \| V_2)$.P. U_a sends to GW_b: $\{ C_1 \}$

Step 2: GW_b has ID_b and the PUF generates the unique challenge-response $CRP_b(C_b, R_b)$ and chooses a random number N_B, then calculates $V_3 = ID_b \oplus N_B$, $C_2 = h(V_3 \| R_b)$.P. The session key $SK_{ug} = C_1 \cdot h(V_3 \| R_b)$. GW_b sends to U_a: $\{ C_2 \}$

Step 3: U_a can now calculate the session key $SK_{ug} = C_2 \cdot h(V_1 \| V_2)$. U_a calculates $HID_a = h(ID_a \| R_a)$, $HPW_a = h(PW_a \| R_a)$. U_a sends to GW_b: $ENC_{SK_{ug}}\{HID_a, HPW_a, CRP_a(C_a, R_a)\}$

Step 4: GW_b verifies U_a's identity using the PUF, then calculates $D_1 = h(HID_a \| HPW_a)$, $D_2 = (D_1 \| HID_a \| R_b)$, $D_3 = D_2 \oplus D_1$. GW_b sends to U_a: $ENC_{SK_{ug}}\{D_1, D_2, D_3, CRP_b(C_b, R_b)\}$

Step 5: U_a verifies GW_b's identity using the PUF, then U_a calculates $M = h(ID_a \| N_A) \oplus HID_a$

5.1.3 Sensor Registration Phase

A portable sensor must also be registered with GW_b before it can connect to the network. Once registered, messages are transmitted via a secure channel, thereby ensuring the security of communications as shown in Fig. 7.

Step 1: The sensor SN_c has ID_c, PW_c, and the PUF generates the unique challenge-response $CRP_c(C_c, R_c)$ and chooses a random number N_s. SN_c calculates $G_1 = PW_c \oplus N_s$, $G_2 = ID_a \oplus N_s$, $F_1 = h(G_1 \| G_2)$.P. SN_c sends to GW_b: $\{ F_1 \}$

Step 2: GW_b has ID_b and the PUF generates the unique challenge-response $CRP_b(C_b, R_b)$ and chooses a random number N_{B1}, then calculates $G_3 = ID_b \oplus N_B1$, $F_2 = h(G_3 \| R_b)$.P. The session key $SK_{sg} = F_1 \cdot h(G_3 \| R_b)$. GW_b sends to SN_c: $\{ F_2 \}$

Step 3: SN_c can now calculate the session key $SK_{sg} = F_2 \cdot h(G_1 \| G_2)$. SN_c sends to GW_b: $ENC_{SK_{sg}}\{ID_c, CRP_c(C_c, R_c)\}$

Step 4: GW_c generates a random number N_{B2} and calculates the pseudo-identity of SN_c: $PID_c = h(ID_c \| N_{B2})$ and $HSID_c = h(ID_c \| N_{B2} \| R_b)$, $SG = h(HSID_c \| R_b) \oplus PID_c$. GW_b sends to SN_c: $ENC_{SK_{sg}}\{PID_c, HSID_c, SG, CRP_b(C_b, R_b)\}$

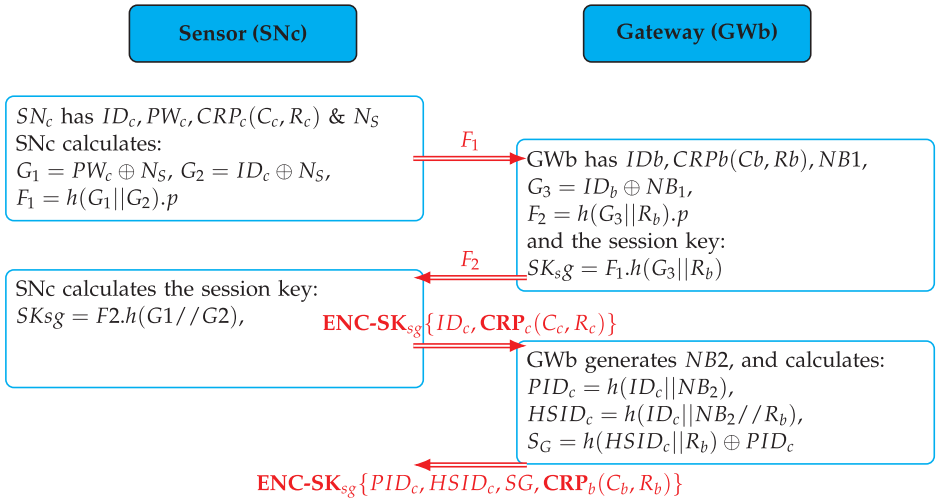


Figure 7: Sensor registration phase

5.1.4 Authentication Phase

Fig. 8 depicts the steps of the authentication phase. When U_a requests a connection to a specific portable sensor SN_c, GW_b must confirm that the user is legitimate.

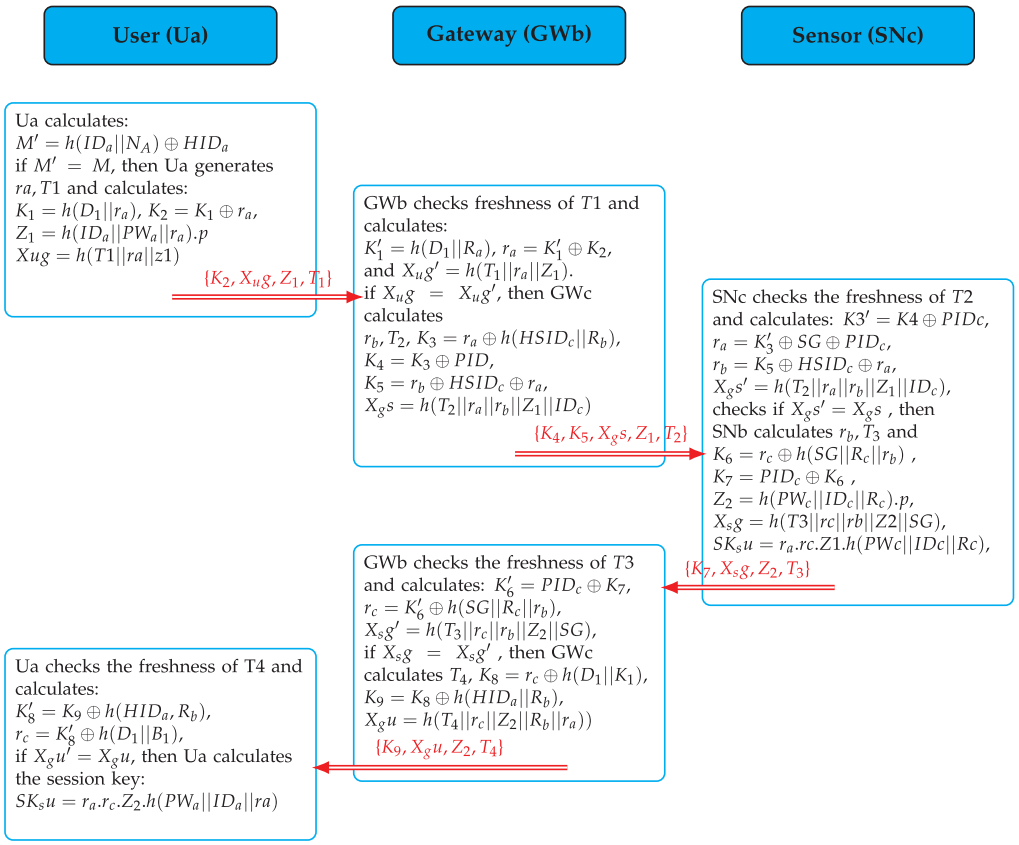


Figure 8: Authentication phase

Step 1: Authentication request

U_a inserts its smart card into a computer or smart card reader, providing its ID_a, PW_a, $CRP_a(C_a, R_a)$. This computer calculates $M' = h(ID_a \| N_A) \oplus HID_a$, then determines if M' is equal to M stored in the smart card. If so, it generates a random number r_a and a timestamp T_1 and calculates $K_1 = h(D_1 \| R_a)$, $K_2 = K_1 \oplus r_a$, $Z_1 = h(ID_a \| PW_a \| R_a)$.P, $X_{ug} = h(T_1 \| r_a \| Z_1)$. U_a sends to GW_b: $\{ K_2, X_{ug}, Z_1, T_1 \}$

Step 2: Verification of authentication parameters and forward the authentication request

GW_b verifies the freshness of T_1. Then, it calculates $K_1' = h(D_1 \| R_a)$, $r_a = K_1' \oplus K_2$, $X_{ug}' = h(T_1 \| r_a \| Z_1)$. If the received X_{ug} equals the calculated X_{ug}', GW_c calculates a random number r_b and a timestamp T_2: $K_3 = r_a \oplus h(HSID_c \| R_b)$, $K_4 = K_3 \oplus PID_c$, $K_5 = r_b \oplus HSID_c \oplus r_a$, $X_{gs} = h(T_2 \| r_a \| r_b \| Z_1 \| ID_c)$. GW_b sends to SN_c: $\{ K_4, K_5, X_{gs}, Z_1, T_2 \}$

Step 3: Verification of the authentication request, authenticity verification response and generate the session key

SN_c verifies the freshness of T_2. Then, SN_c calculates $K_3' = K_4 \oplus PID_c$, $r_a = K_3' \oplus SG \oplus PID_c$, $r_b = K_5 \oplus HSID_c \oplus r_a$, $X_{gs}' = h(T_2 \| r_a \| r_b \| Z_1 \| ID_c)$. If the received X_{gs} equals the calculated X_{gs}', SN_c calculates a random number r_c and a timestamp T_3: $K_6 = r_c \oplus h(SG \| R_c \| r_b)$, $K_7 = PID_c \oplus K_6$, $Z_2 = h(PW_c \| ID_c \| R_c)$.P. $X_{sg} = h(T_3 \| r_c \| r_b \| Z_2 \| SG)$. Finally, SN_c calculates the session key $SK_{su} = r_a \cdot r_c \cdot Z_1 \cdot h(PW_c \| ID_c \| R_c)$. SN_c sends to GW_b: $\{ K_7, X_{sg}, Z_2, T_3 \}$

Step 4: Verification of the authenticity response and forward the authentication response

GW_b verifies the freshness of T_3. Then, it calculates $K_6' = PID_c \oplus K_7$, $r_c = K_6' \oplus h(SG \| R_c \| r_b)$, $X_{sg}' = h(T_3 \| r_c \| r_b \| Z_2 \| SG)$. If the received X_{sg} equals the calculated X_{sg}', GW_b calculates a timestamp T_4: $K_8 = r_c \oplus h(D_1 \| K_1)$, $K_9 = K_8 \oplus h(HID_a \| R_b)$, and calculates $X_{gu} = h(T_4 \| r_c \| Z_2 \| R_b \| r_a)$ for another mutual authentication. GW_b sends to U_a: $\{K_9, X_{gu}, Z_2, T_4 \}$

Step 5: Verification of authentication parameters and generate the session key

U_a verifies the freshness of T_4. Then, it calculates $K_8' = K_9 \oplus h(HID_a \| R_b)$, $r_c = K_8' \oplus h(D_1 \| B_1)$, and calculates $X_{gu} = h(T_4 \| r_c \| Z_2 \| R_b \| r_a)$. If the received X_{gu} equals the calculated X_{gu}', U_a calculates the session key $SK_{su} = r_a \cdot r_c \cdot Z_2 \cdot h(PW_a \| ID_a \| R_a)$

6 LSAP-IoHT Security Analysis

This part thoroughly investigates the protection mechanisms embedded in LSAP-IoHT, highlighting its ability to withstand diverse cyber threats. In addition to an intuitive evaluation underscoring the scheme's strength, we apply structured verification techniques. Leveraging the Real-or-Random model, we perform an in depth theoretical validation of the protocol. Furthermore, we employ Scyther, a widely recognized verification framework, to test the protocol's resilience through automated analysis.

6.1 Informal Analysis

In this analysis, we prove the LSAP-IoHT protocol resistance to known attacks.

6.1.1 Forward Secrecy Property

If persistent credentials are exposed, earlier session information must remain protected. Within our protocol, secret values such as Sk_{ug} and SK_{sg} are independently and randomly selected. As a result, obtaining SK_{SU} does not enable reconstruction of any prior session secrets.

6.1.2 Replay Attack

In our proposed protocol, transmitted messages include timestamps and random numbers, such as T_1, T_2, N_A, and N_B. Timestamps and random numbers are two effective means of preventing replay attacks. By incorporating these mechanisms, our protocol ensures resistance to replay attacks, thus preserving the security and integrity of communications.

6.1.3 Identity Spoofing Attack

For each entity in the system, gateway node, sensor node, and user we used double authentication verification for each transmitted message. Additionally, we adopted a three-factor authentication. Each entity is equipped with an identity, password, and a unique Challenge-Response Pair (CRP) from the PUF. This multi-level security approach ensures that even if an attacker tries to infiltrate the system, they will fail to bypass the double authentication verification.

6.1.4 Insider Attack

The proposed protocol is secure against insider attacks through the implementation of PUF (Physically Unclonable Functions). Each system entity is authenticated using a unique PUF Challenge-Response Pair (CRP), which cannot be duplicated or predicted. This ensures that even privileged insiders, who might have access to other authentication details, cannot replicate the unique PUF responses of other entities. Even if an attacker obtains some data, they will not be able to find the keys since some of the data used to generate the keys was never transmitted.

6.1.5 Physical Attack

The simplicity of IoHT devices makes them highly vulnerable to adversaries. During a physical attack, an attacker may obtain direct access to an IoT device, retrieve its secret keys, and even clone it. The protocol is considered secure against these types of attacks because PUFs and device authentication are implemented on the same chip, ensuring a trusted and secure communication process.

6.1.6 Man-in-the-Middle Attack (MITM)

Our proposed protocol is deemed resistant to MITM attacks because it needs to know specific information to construct valid data and initiate the attack. Discovering all this information is not feasible for an active adversary positioned on the communication line between the three parties involved in the communication.

6.1.7 Stolen Smart Device Attack

The LSAP-IoHT protocol is resilient to stolen device attacks due to its use of physically unclonable functions (PUFs), session dependent secrets, and mutual authentication. Even if an attacker obtains a user's smart card or a registered sensor, critical values such as session keys, pseudo-identities, and challenge-response pairs remain protected through dynamic nonces and device specific computations. Without knowledge of the user's password, random nonces, and the unique PUF-based secrets, an adversary cannot impersonate a legitimate entity or reconstruct valid authentication messages.

6.2 Security Evaluation under the ROR Framework

The Real-or-Random approach represents a commonly adopted technique for formally demonstrating the confidentiality of session keys within cryptographic protocols [28]. The LSAP-IoHT protocol involves

three main participants: U_a, GW_b, and SN_c. We denote the x-th instance of the user as θU_a^x, the y-th instance of the gateway as θGW_b^y, and the z-th instance of the sensor as θSN_c^z. The complete set of protocol participants is represented as: $\mathcal{F} = \{\theta U_a^x,\ \theta GW_b^y,\ \theta SN_c^z\}$.

- **Execute ($\mathcal{F}$) Query:** When this operation is invoked, the adversary is able to eavesdrop on the communication exchanged between the participants U_a, GW_b, and SN_c via an open transmission medium.
- **Send (θ_i, m):** This query allows the adversary to transmit a message m to instance θ_i. The adversary then observes the corresponding output, as generated by the protocol in response.
- **Corrupt (θ_i):** When this request is performed, the adversary acquires confidential elements linked to a specific participant. These may include persistent secrets, ephemeral data, or stored credentials within a smart device.
- **Hash ():** Using this function, the adversary is able to compute the digest corresponding to a given input of predetermined size.
- **Test (θ_i):** Suppose the adversary initiates this request to evaluate the confidentiality of a session key. A virtual coin C is flipped. If $C = 1$, the challenger provides the genuine session key; otherwise, a randomly chosen value is returned to the adversary.

Theorem 1: *In the ROR model [28], we define $Adv_A^{LSAP-IoHT}$ as a function of the attacker's ability to compromise the protocol through query operations; that is, the probability that $\mathcal{A}$ can distinguish the session key from a random value. The advantage is bounded as follows: Within the Real-or-Random framework, the adversary's success metric, denoted as $Adv_A^{LSAP-IoHT}$, quantifies its potential to undermine the protocol by interacting with various oracles. In particular, this value represents the likelihood that the attacker retrieves a valid session secret via the allowed query mechanism.*

$$Adv_A^{LSAP-IoHT} \leq \frac{q_h^2}{|Hash|} + \frac{q_p^2}{|PUF|} + 2 \cdot Adv^{ECDLP} \tag{1}$$

- q_h^2: *total number of queries made to the hash function*,
- q_p^2: *number of times the adversary queries the PUF component*,
- $|Hash|$: *domain size of the hashing function output*,
- $|PUF|$: *range of values producible by the PUF responses*,
- $Adv_A^{ECDLP}()$: *success probability of adversary A in solving the elliptic curve discrete logarithm problem*.

To validate the correctness of the Theorem, the proof proceeds through five game stages, denoted as GM_i for $i = 0, 1, 2, 3, 4$. Let $\text{Succ}_{GM_i}^A$ represent the probability that adversary A succeeds in each respective phase. The sequence is outlined as follows:

Game 0: At this initial game, the adversary attempts to guess a hidden bit b without issuing any oracle requests. Hence, the advantage is calculated as:

$$\text{Adv}_A^{LSAP-IoHT} = \left|2 \cdot \Pr[\text{Succ}_{GM_0}^A] - 1\right| \tag{2}$$

Game 1: The adversary gains access to the `Execute` oracle, which allows it to passively observe full protocol executions between the user, gateway, and sensor. This includes all exchanged messages such as $\{ K_2, X_{ug}, Z_1, T_1 \}$, $\{ K_4, K_5, X_{gs}, Z_1, T_2 \}$, and the final messages that lead to session key derivation. Despite having access to the complete message transcript, the adversary cannot compute the session key $SK_{su} = r_a \cdot r_c \cdot Z_2 \cdot h(PW_c \parallel ID_c \parallel R_c)$ because it depends on the hardness of the elliptic curve discrete logarithm

problem, the random nonces r_a, r_c, and PUF-protected values, which are never transmitted in clear. Hence, the adversary's ability to distinguish the session key does not increase:

$$\Pr[\text{Succ}_1] = \Pr[\text{Succ}_0] \tag{3}$$

Game 2: In this game, the adversary is allowed to interact with the `Hash` oracle, attempting to infer internal values involved in session key derivation. For instance, values such as $Z_1 = h(ID_a \parallel PW_a \parallel r_a) \cdot P$ and $Z_2 = h(PW_c \parallel ID_c \parallel R_c) \cdot P$ depend on secrets and random nonces that are not available to the adversary. Without being able to guess or find collisions on these internal values, the adversary's view in this game remains indistinguishable from that in Game 1. According to the birthday bound, the probability that a collision occurs is bounded as follows:

$$|\Pr[\text{Succ}_2] - \Pr[\text{Succ}_1]| \leq \frac{q_h^2}{2 \cdot |\text{Hash}|} \tag{4}$$

Game 3: This game models a stronger adversary who can invoke the `Corrupt` oracle on a participant (e.g., sensor or user), thereby retrieving stored values such as challenge-response pairs, encrypted identities, or PUF-related secrets. However, the session key relies on PUF-derived data, such as R_c whose outputs are inherently unstable or context dependent due to the physical properties of strong PUFs. Therefore, even with access to stored values, the adversary cannot reproduce valid responses or derive session secrets without replicating the exact environmental conditions under which the PUF was originally evaluated.

The difference in advantage between this game and the previous one is bounded by:

$$|\Pr[\text{Succ}_3] - \Pr[\text{Succ}_2]| \leq \frac{q_p^2}{2 \cdot |\text{PUF}|} \tag{5}$$

Game 4: In the final game, the adversary has access to all protocol transcripts and internal values, including those retrieved through `Corrupt` and `Execute` queries. The only remaining unknowns are secrets derived through elliptic curve multiplication, such as recovering $h(PW_a \parallel ID_a \parallel r_a)$ or $h(PW_c \parallel ID_c \parallel R_c)$ from the values Z_2 and Z_1, respectively. This problem directly reduces to solving the Elliptic Curve Discrete Logarithm Problem (ECDLP). Under standard cryptographic assumptions, solving ECDLP in polynomial time is considered infeasible. Therefore, the difference in advantage between this game and the previous one is bounded by:

$$|\Pr[\text{Succ}_4] - \Pr[\text{Succ}_3]| \leq \text{Adv}_A^{\text{ECDLP}} \tag{6}$$

After simulating the sequence of games Game_0 to Game_4, we apply the triangle inequality to bound the adversary's overall advantage. Each game introduces a new capability for the adversary $\mathcal{A}$, and at each step, we carefully bound the difference in success probability.

Based on the individual game transitions Eqs. (1)–(6), we obtain:

$$|\Pr[\text{Succ}_0] - \Pr[\text{Succ}_4]| \leq \frac{q_h^2}{2|\text{Hash}|} + \frac{q_p^2}{2|\text{PUF}|} + \text{Adv}_A^{\text{ECDLP}} \tag{7}$$

We conclude: $\text{Adv}_A^{LSAP-IoHT} \leq \frac{q_h^2}{2|\text{Hash}|} + \frac{q_p^2}{2|\text{PUF}|} + \text{Adv}_A^{\text{ECDLP}}$

6.3 Formal Analysis

In this section, we conduct an in-depth analysis of the proposed protocol, performing formal verification using the SCYTHER tool. This approach aims to assess the robustness and reliability of the protocol in various scenarios and identify potential vulnerabilities. Using SCYTHER allows us to model the protocol in a simulated environment, where we can explore different parameters and configurations to evaluate its performance and security. This methodical approach provides valuable insights into the protocol's strength, allowing us to take corrective measures if necessary and ensure optimal security for our system. Scyther is a sophisticated verification tool tailored specifically for security protocols, operating under the assumption that all cryptographic functions are flawless [29]. It stands out with its intuitive graphical user interface, making it simple to verify and understand protocols. Whenever an attack is detected concerning a specific claim, Scyther generates corresponding attack graphs, providing a clear visualization of potential vulnerabilities. Furthermore, Scyther can examine all conceivable claims regarding a given protocol. This tool is also valuable in identifying issues arising from protocol design. Additionally, Scyther is capable of generating all possible trace models, allowing for an exhaustive analysis of potential scenarios. The verification can be performed with a limited or unlimited number of sessions, providing optimal flexibility in security assessment [30]. To write protocols in Scyther, we use the SPDL Language(Security Protocol Description Language), which ensures clear syntax and precise protocol representation.

Fig. 9 illustrates an example of SPDL, representing a section of the connection and authentication phase of our proposed protocol. In this excerpt, the three main roles of the system are declared: the "gateway," the "sensor," and the "user." The messages exchanged between these entities are also specified, describing the flow of information during the login and connection phase.

```
protocol protocolIoHT(S, G, U)
{
  role U
  {
    const ra: Nonce;
    var rb , rc: Nonce;
    fresh T1, T2 , T3 , T4 : Timestamp;

    send_1(U, G, K2, Xug, Z1, T1);
    recv_4(G, U, K9 , Xgu , Z2 , T4);
    macro SKsu = Multiplication(ra , rc , Z2 , h(PWa , IDa , Ra));
  }

  role G
  {
    var ra , rc: Nonce;
    const rb: Nonce;
    fresh T1, T2 , T3 , T4 : Timestamp;

    recv_1(U, G, K2, Xug, Z1, T1);
    send_2(G, S, K4,K5, Xgs, Z1 , T2);
    recv_3(S, G, K7 , Xsg , Z2 , T3);
    send_4(G,U, K9 , Xgu , Z2 , T4);
  }

  role S
  {
    var ra , rb: Nonce;
    const rc: Nonce;
    fresh T1, T2 , T3 , T4 : Timestamp;

    recv_2(G,S, K4,K5, Xgs, Z1 , T2);
    send_3(S,G, K7 , Xsg , Z2 , T3);
    macro SKus = Multiplication(ra , rc , Z1 , h(PWc , IDc , Rc));
  }
}
```

Figure 9: Authentication phase in SPDL

After subjecting our protocol to an in-depth formal analysis using the Scyther tool, which is widely used by researchers to evaluate various security systems in related previous works, we can confirm that our system meets the highest security standards. The results of the analysis conclusively demonstrated that our protocol is robust and resilient against various potential attacks, as shown in Fig. 10. By using Scyther, we were able to model and simulate different situations and interactions, thoroughly verifying every aspect of the protocol and identifying any potential weaknesses.

Claim				Status	Comments
protocoleIOHT	U	protocoleIOHT,U1	Secret ra	Ok	No attacks within bounds.
		protocoleIOHT,U2	Secret rc	Ok	No attacks within bounds.
		protocoleIOHT,U3	Nisynch	Ok	No attacks within bounds.
		protocoleIOHT,U4	SKR Multiplication(ra,rc,ECC(h(IDc,PWc,PUF(Cc)),P)...	Ok	No attacks within bounds.
		protocoleIOHT,U5	Niagree	Ok	No attacks within bounds.
		protocoleIOHT,U6	Alive	Ok	No attacks within bounds.
	G	protocoleIOHT,G1	Secret ra	Ok	No attacks within bounds.
		protocoleIOHT,G2	Secret rc	Ok	No attacks within bounds.
		protocoleIOHT,G3	Nisynch	Ok	No attacks within bounds.
		protocoleIOHT,G4	Niagree	Ok	No attacks within bounds.
		protocoleIOHT,G5	Alive	Ok	No attacks within bounds.
	S	protocoleIOHT,S1	Secret ra	Ok	No attacks within bounds.
		protocoleIOHT,S2	Secret rc	Ok	No attacks within bounds.
		protocoleIOHT,S3	Nisynch	Ok	No attacks within bounds.
		protocoleIOHT,S4	SKR Multiplication(ra,rc,ECC(h(IDa,PWa,PUF(Ca)),P)...	Ok	No attacks within bounds.
		protocoleIOHT,S5	Niagree	Ok	No attacks within bounds.
		protocoleIOHT,S6	Alive	Ok	No attacks within bounds.

Figure 10: Result of formal verification

In summary, the formal analysis conducted using the Scyther tool confirms the security and reliability of our protocol, further strengthening our confidence in its ability to protect data and ensure the integrity of communications.

7 Comparative Analysis

In this section, we compare the security features of similar IoHT authentication protocols and we evaluate the computational and Communication costs of each one. Our analysis focuses on how well these protocols resist various attacks and the efficiency of their operations, highlighting the balance between security and performance.

7.1 Security Comparisons

We compare LSAP-IoHT with several related protocols that follow similar architectures. In the comparison table, a "Y" indicates that the protocol is resistant to a specific attack, while an "X" indicates vulnerability. The attack types considered are as follows:

- R1: replay attack
- R2: impersonation attack
- R3: insider attack
- R4: man-in-the-middle attack
- R5: known session-specific temporary information attack
- R6: stolen smart card attack
- R7: offline password guessing attack
- R8: sensor node capture attack
- R9: de-synchronization attack
- R10: session key disclosure attack
- R11: denial of service (DoS) attack
- R12: message modification attack

The results presented in Table 3, show that many existing protocols are susceptible to these types of attacks. In contrast, LSAP-IoHT demonstrates stronger resistance against these threats, providing enhanced security and robustness for communication sessions in IoHT environments. Our cryptanalysis revealed that LAP-IoHT is vulnerable to a wide range of cyberattacks, including man-in-the-middle attacks, replay attacks, and identity spoofing. Despite their claims of robustness, we identified several critical weaknesses that compromise the integrity and confidentiality of the exchanged data. These vulnerabilities indicate that the protocol fails to provide the level of security required for sensitive healthcare applications, leaving them exposed to significant risks.

Table 3: Security Comparison

Protocol	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	Verification tool
[10]	X	X	X	X	X	X	X	X	X	X	X	X	ROR
[12]	Y	Y	Y	Y	Y	Y	Y	Y	X	Y	Y	Y	BAN-AVISPA
[15]	X	Y	Y	X	X	X	X	X	X	Y	X	Y	ROR
[17]	Y	Y	Y	X	Y	Y	Y	X	X	Y	X	X	BAN-ROR-Scyther
[18]	Y	Y	Y	X	Y	Y	Y	X	X	X	Y	X	BAN
LSAP-IoHT	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Scyther

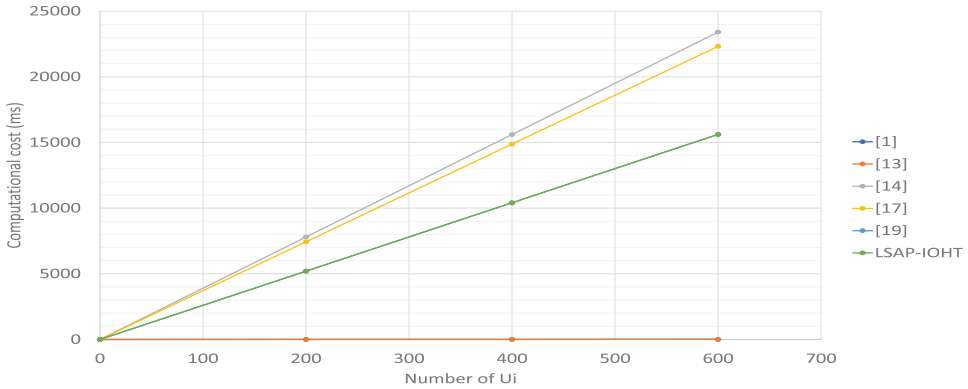
7.2 Computation Cost Analysis

In this section, we evaluate the performance of the LSAP-IoHT protocol by comparing it with other related protocols in terms of computational and communication efficiency. Specifically, in Table 4, we analyze the number of cryptographic operations required, based on the findings in [10]. The time to perform a single hash function operation (HF) is 0.0031 ms for the gateway and 0.0022 ms for the sensor node. For fuzzy extraction operations (FE), it takes 2.2823 ms for the gateway and 1.6197 ms for the sensor node. A single elliptic curve point multiplication (ECCM) requires 16 ms for the gateway and 13 ms for the sensor node, while an elliptic curve point addition (ECCA) takes 0.016 ms for the sensor node and 0.002 ms for the gateway. Additionally, symmetric encryption or decryption operations (AC) require 5.2520 ms for

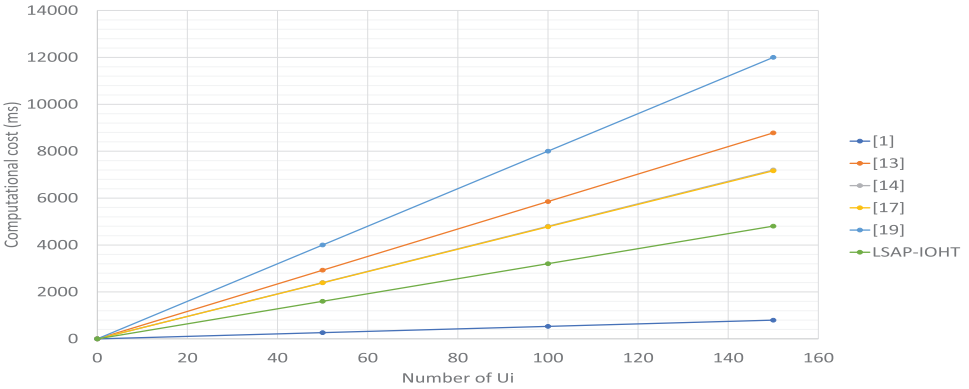
the gateway and 3.7272 ms for the sensor node. We also examine the number of messages exchanged between the user (U), gateway (G), and sensor (S), as well as the total computation time (in milliseconds). According to Table 4, LSAP-IoHT achieves a total computation time of 58.0566 ms, which is lower than [12] (58.5637 ms), [15] (87.0212 ms), [17] (84.9694 ms), and [18] (106.038 ms). This clearly highlights its superior efficiency. Fig. 11 illustrates the computation cost analysis curves.

Table 4: Performance comparison of protocols

Protocol	HF/FE/ECCM/ECCA/AE		Messages (U/G/S)	Computation (ms)		
	Sensor	Server		Sensor	Server	Total
[10]	7/−/−/−/−	20/−/−/−/1	4/4/3	0.0154	5.314	5.3294
[12]	6/−/−/−/−	15/−/3/−/2	4/3/1	0.0132	58.5505	58.5637
[15]	4/−/3/−/−	4/−/3/−/−	2/2/3	39.0088	48.0124	87.0212
[17]	6/−/2/−/3	6/−/2/−/3	6/2/1	37.1948	47.7746	84.9694
[18]	6/−/2/−/−	8/−/5/−/−	6/4/2	26.0132	80.0248	106.038
LSAP-IoHT	6/−/2/−	14/−/2/−	5/6/3	26.0132	32.0434	58.0566



(a) Computation Cost of sensor



(b) Computation Cost of server

Figure 11: The result of computational cost between our scheme and other schemes existing in the literature [1,13,14,17,19]

The combined analysis of Tables 3 and 4 highlights the critical importance of LSAP-IoHT in IoHT systems. Although reference [10] achieves better performance in terms of total computational time, it remains vulnerable to several major security threats. In contrast, LSAP-IoHT successfully resists a wide range of attacks while maintaining a lower computational cost compared to references [12,15,17,18]. It is also the only protocol that withstands all evaluated threats, offering stronger security with acceptable performance, making it well-suited for resource constrained healthcare environments.

7.3 Communication Cost Analysis

The communication cost of the proposed LSAP-IoHT protocol is compared with existing schemes in terms of bit complexity, as shown in Table 5. The total cost for each protocol is calculated based on the number of times each cryptographic component appears in the protocol messages. Specifically, the following assumptions were used: each identifier (ID) is 160 bits, a nonce is 128 bits, a hash function output is 256 bits, a timestamp is 32 bits, and an ECC public key is 160 bits. By applying these values, we obtain the total communication overhead for each protocol. As shown, LSAP-IoHT achieves a significantly lower communication cost (2560 bits) compared to others, while still incorporating strong security primitives.

Table 5: Communication cost comparison (in Bits)

Protocol	ID (160 b)	Nonce (128 b)	Hash (256 b)	Time (32 b)	ECC (160 b)	Total
[1]	–	384	3072	128	–	3584
[13]	–	256	2304	96	–	2656
[14]	160	–	2048	96	160	2464
[17]	–	–	3072	128	640	3840
[19]	–	–	3072	160	640	3872
LSAP-IoHT	–	–	1792	128	640	2560

8 Conclusion

In this paper, we provided an in depth analysis of security in IoHT, highlighting the specific challenges and potential risks associated with this emerging technology. We conducted a cryptanalysis of LAP-IoHT, identifying its vulnerabilities and security limitations. Based on this analysis, we proposed a new authentication scheme tailored to the unique needs of IoHT. By employing advanced cryptographic techniques such as Physical Unclonable Functions (PUF), Elliptic Curve Cryptography (ECC), and three-factor authentication (3FA), our protocol offers robust protection against potential threats. Both our formal and informal analyses confirmed the strong security guarantees of the proposed protocol. We first applied the Real-Or-Random (ROR) model to assess the protocol's resistance to various attack scenarios, demonstrating its theoretical robustness. In addition, we performed a formal verification using the Scyther tool, which allowed us to validate the protocol's security properties within an automated symbolic analysis framework. The combined results highlight the protocol's resilience and reliability, making it well-suited for securing communications in Internet of Healthcare Things (IoHT) environments.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Marwa Ahmim, Nour Ouafi and Ahmed Ahmim; methodology, Insaf Ullah and Reham Almukhlifi; validation, Marwa Ahmim, Djalel

Chefrour, Nour Ouafi and Reham Almukhlifi; formal analysis, Ahmed Ahmim, Insaf Ullah and Marwa Ahmim; writing—original draft preparation, Marwa Ahmim, Ahmed Ahmim, Insaf Ullah, Djalel Chefrour and Nour Ouafi; writing—review and editing, Marwa Ahmim, Ahmed Ahmim and Nour Ouafi; visualization, Djalel Chefrour, Insaf Ullah and Reham Almukhlifi; supervision, Ahmed Ahmim, Insaf Ullah and Djalel Chefrour. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data available within the article.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Coelho KK, Nogueira M, Marim MC, Silva EF, Vieira AB, Nacif JAM. Lorena: low memory symmetric-key generation method for based on group cryptography protocol applied to the internet of healthcare things. IEEE Access. 2022;10:12564–79. doi:10.1109/access.2022.3143210.
2. Khan MA, Ullah S, Ahmad T, Jawad K, Buriro A. Enhancing security and privacy in healthcare systems using a lightweight RFID protocol. Sensors. 2023;23(12):5518. doi:10.3390/s23125518.
3. Wu TY, Wang L, Chen CM. Enhancing the security: a lightweight authentication and key agreement protocol for smart medical services in the ioht. Mathematics. 2023;11(17):3701. doi:10.3390/math11173701.
4. Khan HU, Ali Y, Khan F. A features-based privacy preserving assessment model for authentication of internet of medical things (IoMT) devices in healthcare. Mathematics. 2023;11(5):1197. doi:10.3390/math11051197.
5. Ahmim A, Maazouzi F, Ahmim M, Namane S, Dhaou IB. Distributed denial of service attack detection for the Internet of Things using hybrid deep learning model. IEEE Access. 2023;11:119862–75. doi:10.1109/access.2023.3327620.
6. Ahmim I, Ghoualmi-Zine N, Ahmim A, Ahmim M. Security analysis on "three-factor authentication protocol using physical unclonable function for IoV". Int J Inf Secur. 2022;21(5):1019–26. doi:10.1007/s10207-022-00595-6.
7. Sakraoui S, Ahmim A, Derdour M, Ahmim M, Namane S, Dhaou IB. FBMP-IDS: FL-based blockchain-powered lightweight MPC-secured IDS for 6G networks. IEEE Access. 2024;12:105887–905. doi:10.1109/access.2024.3435920.
8. Ahmim M, Ahmim A, Ferrag MA, Ghoualmi-Zine N, Maglaras L. ESIKE: an efficient and secure internet key exchange protocol. Wirel Pers Commun. 2023;128(2):1309–24. doi:10.1007/s11277-022-10001-y.
9. Ahmim I, Ghoualmi-Zine N, Ahmim M, Ahmim A. Lightweight authentication protocols for internet of vehicles: network model, taxonomy and challenges. In: 2022 4th International Conference on Pattern Analysis and Intelligent Systems (PAIS); 2022 Oct 12–13; Oum El Bouaghi, Algeria. p. 1–6.
10. Chen CM, Chen Z, Kumari S, Lin MC. LAP-IoHT: a lightweight authentication protocol for the internet of health things. Sensors. 2022;22(14):5401. doi:10.3390/s22145401.
11. Kumar V, Mahmoud MS, Alkhayyat A, Srinivas J, Ahmad M, Kumari A. RAPCHI: robust authentication protocol for IoMT-based cloud-healthcare infrastructure. J Supercomput. 2022;78(14):16167–96. doi:10.1007/s11227-022-04513-4.
12. Aghili SF, Mala H, Shojafar M, Peris-Lopez P. LACO: lightweight three-factor authentication, access control and ownership transfer scheme for e-health systems in IoT. Future Gener Comput Syst. 2019;96(1):410–24. doi:10.1016/j.future.2019.02.020.
13. Zhang L, Zhang Y, Tang S, Luo H. Privacy protection for e-health systems by means of dynamic authentication and three-factor key agreement. IEEE Trans Ind Electron. 2018;65(3):2795–805. doi:10.1109/tie.2017.2739683.
14. Soni P, Pal AK, Islam SH. An improved three-factor authentication scheme for patient monitoring using WSN in remote health-care system. Comput Methods Programs Biomed. 2019;182(1):105054. doi:10.1016/j.cmpb.2019.105054.

15. Challa S, Das AK, Odelu V, Kumar N, Kumari S, Khan MK, et al. An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks. Comput Electr Eng. 2018;69(6):534–54. doi:10.1016/j.compeleceng.2017.08.003.
16. Ullah I, Khan MA, Abdullah AM, Noor F, Innab N, Chen CM. Enabling secure communication in wireless body area networks with heterogeneous authentication scheme. Sensors. 2023;23(3):1121. doi:10.3390/s23031121.
17. Rabie OBJ, Selvarajan S, Hasanin T, Mohammed GB, Alshareef AM, Uddin M. A full privacy-preserving distributed batch-based certificate-less aggregate signature authentication scheme for healthcare wearable wireless medical sensor networks (HWMSNs). Int J Inf Secur. 2024;23(1):51–80. doi:10.1007/s10207-023-00798-5.
18. Xie Y, Zhang S, Li X, Li Y, Chai Y. CasCP: efficient and secure certificateless authentication scheme for wireless body area networks with conditional privacy-preserving. Secur Commun Netw. 2019;2019(1):5860286. doi:10.1155/2019/5860286.
19. Ji S, Gui Z, Zhou T, Yan H, Shen J. An efficient and certificateless conditional privacy-preserving authentication scheme for wireless body area networks big data services. IEEE Access. 2018;6:69603–11. doi:10.1109/access.2018.2880898.
20. Thakur G, Prajapat S, Kumar P, Das AK, Shetty S. An efficient lightweight provably secure authentication protocol for patient monitoring using wireless medical sensor networks. IEEE Access. 2023;11:114662–79. doi:10.1109/access.2023.3325130.
21. Servati MR, Safkhani M. ECCbAS: an ECC based authentication scheme for healthcare IoT systems. Pervasive Mob Comput. 2023;90(15):101753. doi:10.1016/j.pmcj.2023.101753.
22. Li X, Peng J, Obaidat MS, Wu F, Khan MK, Chen C. A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems. IEEE Syst J. 2019;14(1):39–50. doi:10.1109/jsyst.2019.2899580.
23. Amin R, Biswas G. A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. Ad Hoc Netw. 2016;36(4):58–80. doi:10.1016/j.adhoc.2015.05.020.
24. Sureshkumar V, Amin R, Vijaykumar V, Sekar SR. Robust secure communication protocol for smart healthcare system with FPGA implementation. Future Gener Comput Syst. 2019;100(4):938–51. doi:10.1016/j.future.2019.05.058.
25. Ataei Nezhad M, Barati H, Barati A. An authentication-based secure data aggregation method in internet of things. J Grid Comput. 2022;20(3):29. doi:10.1007/s10723-022-09619-w.
26. Khajehzadeh L, Barati H, Barati A. A lightweight authentication and authorization method in IoT-based medical care. Multimed Tools Appl. 2025;84(12):11137–76. doi:10.1007/s11042-024-19379-2.
27. Dolev D, Yao A. On the security of public key protocols. IEEE Trans Inform Theory. 1983;29(2):198–208. doi:10.1109/tit.1983.1056650.
28. Ma H, Wang C, Xu G, Cao Q, Xu G, Duan L. Anonymous authentication protocol based on physical unclonable function and elliptic curve cryptography for smart grid. IEEE Syst J. 2023;17(4):6425–36. doi:10.1109/jsyst.2023.3289492.
29. Cremers CJ. The scyther tool: verification, falsification, and analysis of security protocols: tool paper. In: International Conference on Computer Aided Verification. Cham, Switzerland: Springer; 2008. p. 414–8.
30. Cao J, Ma M, Fu Y, Li H, Zhang Y. CPPHA: capability-based privacy-protection handover authentication mechanism for SDN-based 5G HetNets. IEEE Trans Depend Secure Comput. 2019;18(3):1182–95. doi:10.1109/tdsc.2019.2916593.