

AEBAAP: An Efficient ECC-Based Authentication Protocol Designed for Wireless Networks

Reza Hooshmand^{1,*} and Sajjad Alizadeh²

^{1*}Department of Electrical Engineering, Shahid Sattari University of Aeronautical Science and Technology, Tehran, Iran.

²Department of Electrical Engineering, Yazd University, Yazd, Iran.

*Corresponding author(s). E-mail(s): rhooshmand50@yahoo.com;

Abstract

The Internet of Things (IoT) comprises a vast array of interconnected devices that rely heavily on wireless communication to enable applications in domains such as healthcare, transportation, and smart environments. However, intrinsic vulnerabilities in wireless networks, such as susceptibility to eavesdropping, impersonation, and message tampering, require the development of robust yet lightweight authentication mechanisms. To address these concerns, we introduce AEBAAP, an efficient authentication protocol based on elliptic curve cryptography (ECC), specifically designed for secure communication in resource-constrained IoT environments. AEBAAP ensures mutual authentication, data integrity, and resilience against both passive and active attacks by leveraging the cryptographic strength and computational efficiency of ECC. We evaluate the protocol's performance through experimental implementation and analyze its security using both informal reasoning and formal verification tools, including Scyther and ProVerif. The results demonstrate that AEBAAP achieves strong security guarantees with minimal computational and communication overhead, making it a promising solution for securing next-generation wireless IoT systems.

Keywords: ECC, Authentication, Wireless Networks, Resource-Constrained Environments, Scyther, ProVerif

1 Introduction

The Internet of Things (IoT) comprises a vast network of spatially distributed sensor nodes characterized by constrained energy and computational capabilities. These

nodes form intelligent, self-organizing ad hoc communication systems capable of operating autonomously in unknown or remote environments. IoT technologies have enabled significant advancements across a range of domains, including battlefield monitoring, smart agriculture, urban infrastructure, wildlife tracking, transportation systems, and most notably, healthcare [Sadhukhan et al. \(2021\)](#). Among these, healthcare has seen particularly transformative improvements through real-time monitoring and data-driven decision-making. Smart healthcare systems leverage integrated sensing, wireless communication, and intelligent data analytics to facilitate continuous, remote monitoring of patient health. At the core of such systems lies medical surveillance infrastructure, which collects and shares patient data in real time among healthcare providers. Interconnected physiological sensors capture vital signs such as blood pressure, heart rate, blood glucose, respiration rate, ECG signals, and oxygen saturation [Chen et al. \(2019\)](#). These measurements are transmitted wirelessly, enabling clinicians to make timely and precise diagnoses. The miniaturization of medical devices has led to the emergence of wireless medical sensor networks (WMSNs), which provide enhanced mobility for patients and allow seamless access to health data by physicians [Jiang et al. \(2016\)](#).

Given the critical and sensitive nature of medical data, ensuring its confidentiality, authenticity, and integrity is paramount. Information transmitted over wireless channels is inherently vulnerable to various attacks, including eavesdropping, data manipulation, spoofing, and replay. It is essential that clinicians receive data that can be unequivocally attributed to the correct patient and authenticated medical devices. Failing to ensure this integrity can result in erroneous diagnoses and potentially harmful medical interventions. Consequently, robust authentication mechanisms are needed to bind sensor-generated data to a verified user and device identity, while simultaneously preventing impersonation or cloning attacks. To satisfy these objectives, authentication protocols for medical IoT environments must meet several critical security requirements. First, they must ensure mutual authentication, confirming the identities of both the patient and the medical entity before any data exchange. Second, anonymity should be preserved to protect users from identification by third parties. Third, untraceability must be achieved so that no entity, including service providers or administrators, can link multiple interactions to a specific user. Fourth, secure session key establishment must be provided to protect each communication instance with a unique cryptographic key. Fifth, protocols must offer perfect forward secrecy, ensuring that past session keys remain confidential even if long-term credentials are later compromised. Lastly, the system must be resilient to known attacks, including impersonation, replay, and man-in-the-middle threats, which are prevalent in open wireless environments.

Moreover, it is crucial that only authenticated and authorized entities can infer the true identity of a patient. Protecting identity confidentiality and preventing the injection of false medical data are essential for maintaining patient privacy and system trustworthiness [Suganthi et al. \(2020\)](#). To address the above security requirements in constrained IoT-based healthcare systems, we propose AEBAP, a robust and lightweight authentication protocol built upon elliptic curve cryptography (ECC). AEBAP facilitates secure mutual authentication among users, wearable devices, and

gateway nodes while supporting dynamic session key generation for each communication session. To enhance user privacy, biometric data is protected within the authentication process, maintaining both anonymity and accountability. The protocol’s security properties are rigorously evaluated through informal analysis and verified using formal methods and tools such as Scyther and ProVerif.

1.1 Threat Model

In designing the AEBAP authentication protocol, we consider a realistic adversarial model based on the capabilities and limitations typically assumed in IoT-based healthcare systems. The communication between entities—including patients’ wearable sensors, gateway devices, and central medical servers—takes place over insecure wireless channels, making the system vulnerable to various active and passive threats. The adversary is assumed to have full control over the communication channel and can intercept, modify, replay, or inject messages arbitrarily. We assume that the adversary has the following capabilities:

- **Eavesdropping:** The adversary can listen to all messages transmitted over public channels, including sensor-to-gateway and gateway-to-server communication links.
- **Replay Attacks:** The adversary may store legitimate protocol messages and re-transmit them at a later time to impersonate a valid entity or disrupt session freshness.
- **Impersonation:** The adversary may attempt to impersonate a legitimate user, sensor, or gateway by forging authentication messages or injecting falsified credentials.
- **Man-in-the-Middle (MitM):** The adversary may intercept and alter communication between two legitimate parties to gain unauthorized access or manipulate exchanged data.
- **Device Cloning and Identity Theft:** The attacker may physically capture a device or clone its identity to masquerade as a legitimate participant in the authentication process.
- **Linkability and Traceability:** The adversary may try to correlate protocol messages over multiple sessions to link them to a particular user or device, thereby violating user privacy.

We also assume that all end devices—including sensors, gateways, and servers—are not physically compromised during the execution of the protocol. The attacker cannot extract long-term secrets stored in a secure hardware module. However, the protocol must be designed to maintain forward secrecy even if long-term credentials are compromised after session completion. Furthermore, the adversary is considered to be computationally bounded and cannot break standard cryptographic primitives such as the elliptic curve discrete logarithm problem (ECDLP) or cryptographic hash functions used in the protocol. The design of AEBAP incorporates measures to resist all of the above attacks while maintaining efficiency suitable for resource-constrained environments.

1.2 System Model

The system model for the proposed AEBAP protocol reflects a typical IoT-based healthcare environment composed of three primary entities: the user (or patient), the wearable medical sensors, and the healthcare service provider. These components interact through a gateway device—such as a smartphone—which acts as a mediator between the sensors and the remote medical server. The communication among these entities takes place over insecure wireless networks. The components of the system are described as follows:

- **User (U):** The user is typically a patient equipped with one or more wearable medical sensors that continuously monitor physiological parameters such as heart rate, blood pressure, or blood glucose levels. The user is responsible for initiating the authentication process when attempting to connect to the healthcare service provider.
- **Wearable Sensors (S):** These sensors are lightweight, resource-constrained devices embedded on or near the patient’s body. They collect health-related data and transmit it to the gateway device. Due to their limited computational capabilities and battery life, the authentication protocol must be lightweight.
- **Gateway Device (G):** The gateway acts as an intermediate trusted device—such as a smartphone or dedicated hub—that aggregates data from sensors and communicates with the healthcare server. It also participates in the authentication process by facilitating secure message forwarding and session key establishment.
- **Healthcare Server (H):** This centralized entity belongs to a trusted healthcare provider. It is responsible for authenticating users and devices, managing session keys, and maintaining patient health records securely. The server is assumed to be computationally powerful and always online.

All communication between the user, sensors, and the healthcare server is routed through the gateway. The protocol assumes that the server and gateway are provisioned with necessary cryptographic keys during a secure registration phase. Furthermore, all entities maintain synchronized clocks within a permissible time window to prevent replay attacks based on timestamp validation. The system operates in two main phases: (i) an authentication phase in which mutual verification between the user and server is established, and (ii) a session key agreement phase that ensures secure communication for subsequent data exchange. The protocol design takes into account the limitations of wearable sensors while ensuring robust protection against threats outlined in the threat model.

1.3 Paper Organization

The remainder of this paper is structured as follows. Section 2 reviews relevant literature on authentication protocols for IoT-based healthcare systems, highlighting their advantages and limitations. In Section 3, the proposed ECC-based authentication protocol, AEBAP, is introduced with an emphasis on its design rationale and security goals. Section 4 provides a detailed security analysis of the protocol, including both informal reasoning and formal verification using automated tools. Section 5 presents

the performance evaluation of AEABAP and compares its efficiency and security features against existing schemes. Finally, Section 6 concludes the paper by summarizing key findings and discussing directions for future work.

2 Related Work

Ensuring secure authentication remains a fundamental requirement in the architecture of healthcare-focused IoT networks, where resource-limited nodes routinely transmit confidential medical information through unsecured communication media. Although various lightweight cryptographic protocols—particularly those employing elliptic curve cryptography (ECC)—have been introduced to improve security and efficiency, many continue to fall short against advanced adversarial threats, including identity spoofing, key disclosure, message replay, and privacy violations. For instance, Kumar et al. [Kumar et al. \(2019\)](#) introduced a cloud-assisted authentication mechanism tailored to telemedicine environments. However, the scheme was later proven vulnerable to impersonation and data leakage. Similarly, Zheng et al. [Zheng et al. \(2018\)](#) proposed a TMIS-based approach for smart campuses, which was critically evaluated by Safkhani et al. [Safkhani and Vasilakos \(2019\)](#), revealing its weakness against replay and impersonation attacks.

Xiang et al. [Xiang and Zheng \(2020\)](#) designed a situational authentication scheme for smart grid and home IoT systems. Despite its context-awareness, Oh et al. [Oh et al. \(2021\)](#) uncovered its susceptibility to session key exposure and stolen device impersonation. Moreover, the ECC-based anonymity-preserving protocol by Shuai et al. [Shuai et al. \(2019\)](#) was shown to lack resistance to user traceability attacks. Protocols for wearable healthcare devices have also been proposed, such as that of Gupta et al. [Gupta et al. \(2019\)](#). Nevertheless, Hajian et al. [Hajian et al. \(2020\)](#) identified major flaws in the scheme, including vulnerability to insider threats and synchronization breakdowns. Likewise, the lightweight protocol introduced by Xu et al. [Xu et al. \(2019\)](#) was found by Alzahrani et al. [Alzahrani et al. \(2021\)](#) to be prone to session key compromise and replay-based intrusions.

In the domain of ultra-lightweight solutions, Wang et al. [Wang et al. \(2022\)](#) and Chander et al. [Chander and Gopalakrishnan \(2022\)](#) suggested protocols optimized for low-energy devices. However, Servati et al. [Servati et al. \(2022\)](#) highlighted that Wang’s model disclosed confidential secrets and suffered from desynchronization. Likewise, the SecLAP protocol by Aghili et al. [Aghili et al. \(2019a\)](#) was shown by Safkhani et al. [Safkhani et al. \(2020\)](#) to suffer from traceability and secret value leakage. More advanced ECC-driven models have also been explored. Gabsi et al. [Gabsi et al. \(2021\)](#) proposed a protocol for general IoT applications, though Arslan et al. [Arslan and Bingöl \(2022\)](#) later found violations in anonymity and secrecy guarantees. Huang et al. [Huang \(2024\)](#) developed a three-factor ECC-based scheme for WSNs, while Kumar et al. [Kumar et al. \(2024\)](#) suggested a two-factor solution for video surveillance networks. Additionally, Rani et al. [Rani and Tripathi \(2024\)](#) introduced a blockchain-enabled protocol for privacy-aware medical data exchange among hospitals.

Efforts to improve scalability and trust distribution led to hybrid ECC-blockchain models. Chatterjee et al. [Chatterjee et al. \(2023\)](#) optimized ECC for WSNs, while

Wang et al. [Wang et al. \(2024\)](#) incorporated blockchain for decentralized authentication. In a healthcare context, Khan et al. [Khan et al. \(2023b\)](#) developed an ECC-based scheme for protecting electronic health records. Other blockchain-assisted authentication methods were presented by Subramani et al. [Jegadeesan et al. \(2022\)](#) and Jia et al. [Jia et al. \(2022\)](#), though both demonstrated security shortcomings—namely, user traceability and leakage of critical parameters. Wang et al. [Wang and Liu \(2022\)](#) and Pu et al. [Pu et al. \(2022\)](#) designed protocols for WBANs and low-power devices, while RFID-specific schemes were proposed by Shariq et al. [Shariq et al. \(2021\)](#) and Khan et al. [Khan et al. \(2023a\)](#). A comprehensive cryptanalysis by Hosseinzadeh et al. [Hosseinzadeh et al. \(2024\)](#) revealed critical flaws in several of these designs, leading to an improved solution resistant to secret disclosure.

Sureshkumar et al. [Sureshkumar et al. \(2019\)](#) proposed an ECC-based protocol specifically aimed at secure medical IoT communications, focusing on improving performance metrics. Nonetheless, Servati et al. [Servati and Safkhani \(2023\)](#) later revealed that the scheme remains exposed to serious vulnerabilities, such as loss of synchronization, data integrity inconsistencies, and traceability concerns. In their successive research [Servati et al. \(2022, 2025\)](#), Servati et al. conducted detailed cryptographic analysis of ECC-based authentication frameworks. Their most recent contribution [Servati et al. \(2025\)](#) presents a refined authentication scheme for healthcare IoT networks, delivering strong resistance against impersonation and traceability while maintaining computational efficiency. These findings reinforce the ongoing necessity of designing authentication protocols that not only provide formal security proofs but also remain adaptable to practical IoT environments. Despite the growing body of research, existing protocols often emphasize either computational efficiency or cryptographic strength, while neglecting essential features such as unlinkability, forward secrecy, and defense against offline password guessing. This imbalance reflects a gap in the literature and motivates the need for a new authentication design that is both lightweight and holistically secure. In response, this study presents a novel ECC-based authentication protocol that aims to satisfy the critical security requirements of IoT-based healthcare environments. The proposed design is validated through both formal security analysis and performance assessment to demonstrate its practicality and resilience.

3 Proposed Authentication Protocol: AEBAP

The proposed authentication protocol, referred to as AEBAP (Authentication protocol with Elliptic curve and Biometric-enhanced Access Protection), is designed to provide secure, lightweight, and privacy-preserving authentication in IoT-enabled environments. The protocol is structured into multiple well-defined phases, each targeting a critical security objective in the lifecycle of user and device interaction. Specifically, AEBAP consists of the following core phases: (1) **User Registration Phase**, in which each user is securely initialized and issued a smart card containing hashed credentials; (2) **Sensor and Gateway Registration Phase**, where each sensor node and gateway are uniquely identified and assigned cryptographic parameters by the system administrator; (3) **Login and Authentication Phase**, which performs mutual

authentication among the user, gateway, and sensor node while establishing a secure session key; and (4) **Password Change Phase**, allowing legitimate users to update their authentication credentials securely. By integrating elliptic curve cryptography (ECC), one-way hash functions, and user-specific secrets, AEBAP ensures confidentiality, integrity, mutual authentication, and resistance to common attacks such as replay, impersonation, and session key leakage. The following subsections detail the design and operations of each phase.

3.1 Initialization Phase

The initialization phase serves as the foundation of the proposed authentication framework, during which the system administrator (A) configures all network entities and cryptographic primitives. This phase ensures that each user, sensor, and gateway node is securely registered with the authentication server before any communication occurs. Each registered user is assigned a unique identity ID_i , a password PW_i , and optionally biometric information B_i . These credentials are processed using a cryptographic hash function $H(\cdot)$ prior to storage, ensuring that original data cannot be reconstructed even in case of unauthorized access. The hash function used in AEBAP is a one-way, collision-resistant function that maps variable-length inputs to fixed-length outputs (e.g., 160 or 256 bits), satisfying essential security properties such as diffusion, unpredictability, and one-wayness.

Formally, a hash function $H : \{0, 1\}^* \rightarrow \{0, 1\}^k$ is considered secure if it is computationally infeasible for any probabilistic polynomial-time adversary $\mathcal{A}$ to find two different inputs $m \neq n$ such that $H(m) = H(n)$. The adversary's advantage in breaking the hash function is defined by:

$$\text{Adv}_{\mathcal{A}}^{\text{Hash}}(t) = \Pr [(m, n) \in_R \mathcal{A} : m \neq n, H(m) = H(n)] \leq \epsilon, \quad (1)$$

where ϵ is a negligible probability and t is the allowable execution time. To further enhance security, salting and iterative hashing techniques are applied, ensuring that even identical credentials yield unique hash outputs, thereby resisting rainbow table attacks. To support low-power and resource-limited devices, the protocol uses lightweight mathematical operations. The bitwise exclusive-OR operation ($\oplus$) is applied to obfuscate or combine sensitive values securely, while the concatenation operator ($\parallel$) is used to assemble multiple fields before hashing. The key symbols and notations used throughout the protocol are listed in Table ??, providing a unified reference for the authentication process.

3.2 User Enrollment Phase

The user enrollment phase is responsible for establishing an initial trust relationship between the user and the authentication server through secure registration and credential provisioning. This phase ensures that the authentication data is protected even in the presence of potential adversaries and includes three main steps:

1. **User-side Initialization:** The user selects a unique identifier ID_i and a password PW_i , then generates a random number u_i . Using these values, the user computes the following:

Table 1 Notations Used in the Proposed Authentication Protocol

Symbol	Description
U_i, GW_j, SN_k	User, gateway, and sensor nodes
ID_i, PW_i	User ID and password
us_i	User's private secret
B_i	Biometric information
SC	Smart card issued to the user
Sk, S_{Key}	Secret key, session key
us_i	Secret of system administrator
$GWID_j, SNID_k$	Gateway and sensor node identifiers
SSN_k	Secret value of sensor node SN_k
$Pkg = sg \cdot P$	Gateway public key (ECC)
sg	Gateway private key
P	Base point on elliptic curve
F_q, F_q^*	Finite field and its multiplicative group
TS_x	Timestamps for freshness checks
$H(.)$	One-way hash function
$\cdot, \oplus, \parallel$	Scalar multiplication, XOR, concatenation
$\stackrel{?}{=}$	Equality check for verification
ΔT	Maximum delay tolerance threshold

$$\begin{aligned}
HID_a &= h(ID_i \parallel u_i) \\
HPW_i &= h(PW_i \parallel u_i) \\
HPID_i &= h(HID_a \parallel HPW_i)
\end{aligned}$$

The values $\langle HPW_i, HPID_i \rangle$ are then transmitted to the authentication server through a secure and confidential communication channel.

2. **Server-side Computation:** Upon receiving the request, the authentication server computes:

$$B_1 = h(HPID_i \parallel HPW_i)$$

The server then returns $\langle B_1, h(.) \rangle$ to the user via a secure or private channel. Here, $h(.)$ denotes the cryptographic hash function employed in the system.

3. **Smart Card Generation:** Upon receiving the message, the user selects a unique secret value us_i and recomputes the hash:

$$B_1^* = h(HPID_i \parallel HPW_i)$$

The user then creates a smart card with the contents:

$$SC = \langle B_1^*, h(.), us_i \rangle$$

The secret value us_i is stored securely in the user's device and is never transmitted over the network. This user-specific secret strengthens the scheme's resistance to offline dictionary and impersonation attacks.

The above registration process is outlined formally in Algorithm 1.

Algorithm 1 User Enrollment Phase

- 1: **Step 1: User Initialization**
 - 2: User selects ID_i , PW_i , and generates a random u_i
 - 3: Computes $HID_a = h(ID_i || u_i)$
 - 4: Computes $HPW_i = h(PW_i || u_i)$
 - 5: Computes $HPID_i = h(HID_a || HPW_i)$
 - 6: Sends $\langle HPW_i, HPID_i \rangle$ to the server securely
 - 7: **Step 2: Server Computation**
 - 8: Computes $B_1 = h(HPID_i || HPW_i)$
 - 9: Sends $\langle B_1, h(.) \rangle$ to the user securely
 - 10: **Step 3: Smart Card Generation**
 - 11: User selects private secret us_i
 - 12: Computes $B_1^* = h(HPID_i || HPW_i)$
 - 13: Stores smart card: $SC = \langle B_1^*, h(.), us_i \rangle$
-

3.3 Sensor and Gateway Registration Phase

This phase is managed by the system administrator (SA) to securely initialize and register both gateway and sensor nodes before they participate in the authentication protocol. The objective is to assign unique identifiers and generate associated secret credentials that enable mutual trust during authentication.

1. Gateway Node Initialization:

The administrator selects a unique identifier $GWID_j$ for the j -th gateway node and computes the corresponding secret using:

$$S_g = h(SA_{Se} || GWID_j)$$

The pair $\langle GWID_j, S_g \rangle$ is securely written into the memory of the gateway node GW_j .

2. Sensor Node Initialization:

Similarly, a unique identifier $SNID_k$ is assigned to the k -th sensor node, and its secret is derived as:

$$SSN_k = h(SA_{Se} || SNID_k)$$

The registration values are then distributed as follows:

- Stored on the sensor node SN_k : $\langle SSN_k, GWID_j, P \rangle$
- Stored on the gateway node GW_j : $\langle SSN_k, SNID_k, P \rangle$

Here, P denotes the base point used in elliptic curve cryptographic operations. This reciprocal storage ensures that both sensor and gateway nodes possess the required credentials to perform secure mutual authentication and communication.

3.4 Login and Authentication Phase

This phase enables a legitimate user to securely log in to the system and initiates a multi-party authentication process among the user, gateway, and sensor node, culminating in a shared session key. The phase consists of five main stages: user login, gateway verification, sensor validation, session key agreement, and confirmation.

Step 1: User Login and Verification

The user initiates the login process by inserting their identity ID_i , password PW_i , and private secret us_i into the device. The smart card (SC) performs the following computations:

$$\begin{aligned} b_i &= h(B_i) \\ HPID_i &= h(ID_i \parallel b_i) \\ HPW_i &= h(PW_i \parallel b_i) \\ A_1 &= h(HPW_i \parallel HPID_i \parallel us_i) \end{aligned}$$

The smart card then compares the computed A_1 with the stored reference value B_1^* . If $A_1 = B_1^*$, the login is successful and the authentication protocol proceeds.

Step 2: Message Generation and Transmission to Gateway

After successful login, the user generates a random nonce $n_1 \in F_q$ and records a current timestamp TS_1 . The following values are computed:

$$\begin{aligned} C_0 &= n_1 \cdot P \\ C_1 &= n_1 \cdot PK_G \\ C_2 &= C_1 \oplus SNID_k \\ C_3 &= h(HID_a) \oplus h(C_1 \parallel TS_1) \\ C_4 &= h(C_0 \parallel C_2 \parallel C_3 \parallel h(HID_a)) \end{aligned}$$

The user then sends $\langle C_0, C_2, C_3, C_4, TS_1 \rangle$ to the gateway node.

Step 3: Gateway Verification and Relay

Upon receiving the message, the gateway first verifies the freshness of the timestamp:

$$|TS_1 - TS_1^*| \leq \Delta T$$

If valid, the gateway performs the following:

$$\begin{aligned}
h(HID_a^*) &= C_3 \oplus h(C_1^* \parallel TS_1) \\
SNID_k &= C_1^* \oplus C_2 \\
C_4^* &= h(C_0 \parallel C_2 \parallel C_3 \parallel h(HID_a^*))
\end{aligned}$$

The gateway checks if $C_4 = C_4^*$ to ensure integrity. Then, it computes:

$$\begin{aligned}
C_5 &= h(HID_a^*) \oplus h(SSN_k \parallel GWID_j \parallel TS_2) \\
C_6 &= h(SSN_k \parallel GWID_j \parallel C_0 \parallel C_5 \parallel TS_2 \parallel h(HID_a^*))
\end{aligned}$$

The gateway sends $\langle C_0, C_5, C_6, TS_2 \rangle$ to the associated sensor node.

Step 4: Sensor Validation and Session Key Generation

The sensor verifies timestamp freshness:

$$|TS_2 - TS_2^*| \leq \Delta T$$

Then, it computes:

$$\begin{aligned}
h(HID_a^{**}) &= C_5 \oplus h(SSN_k \parallel GWID_j \parallel TS_2) \\
C_6^* &= h(SSN_k \parallel GWID_j \parallel C_0 \parallel C_5 \parallel TS_2 \parallel h(HID_a^{**}))
\end{aligned}$$

If $C_6^* = C_6$, the sensor proceeds by generating a random nonce $n_2 \in F_q$ and records TS_3 :

$$\begin{aligned}
C_7 &= n_2 \cdot P \\
S_{Key_S} &= n_2 \cdot C_0 \\
C_8 &= h(C_0 \parallel SSN_k \parallel C_7 \parallel GWID_j \parallel TS_3)
\end{aligned}$$

It sends $\langle C_7, C_8, TS_3 \rangle$ to the gateway.

Step 5: Final Validation and Session Confirmation

The gateway performs the following checks:

$$|TS_3 - TS_3^*| \leq \Delta T, \quad C_8^* = h(C_0 \parallel SSN_k \parallel C_7 \parallel GWID_j \parallel TS_3)$$

If valid, it computes:

$$C_9 = h(C_7 \parallel SNID_k \parallel h(HID_a^*))$$

Finally, it sends $\langle C_7, C_9, TS_4 \rangle$ to the user. The user verifies:

$$|TS_4 - TS_4^*| \leq \Delta T, \quad C_9^* = h(C_7 \parallel SNID_k \parallel h(HID_a^*))$$

If $C_9^* = C_9$, the authentication succeeds and the user derives the final session key:

$$S_{Key_U} = S_{Key_S}$$

Thus, mutual authentication and session key agreement are successfully completed among the user, gateway, and sensor. The complete sequence of this process is formally outlined in Algorithm 2, which details all five stages of secure login, verification, and session establishment across the involved entities.

3.5 Password Change Phase

This phase allows a legitimate user to update their identity and password without requiring re-registration through the system administrator. To ensure that only an authenticated user can initiate this change, the smart card first performs a local verification before computing and updating new values.

Step 1: Identity Verification

The user inputs their current identity ID_i , password PW_i , and private randomness u_i into the smart card. Using these, the following computations are performed:

$$\begin{aligned} HID_a &= h(ID_i \parallel u_i) \\ HPW_i &= h(PW_i \parallel u_i) \\ HPID_i &= h(HID_a \parallel HPW_i) \\ B_1 &= h(HPID_i \parallel HPW_i) \end{aligned}$$

The computed value B_1 is then compared with the stored reference B_1^* in the smart card. If $B_1 = B_1^*$, the user's legitimacy is confirmed and the password change process proceeds.

Step 2: Updating Credentials

The user selects a new identity ID_i^{new} and new password PW_i^{new} . With the same private value u_i , the smart card computes the updated parameters:

$$\begin{aligned} HID_a^{new} &= h(ID_i^{new} \parallel u_i) \\ HPW_i^{new} &= h(PW_i^{new} \parallel ID_i^{new}) \\ HPID_i^{new} &= h(HID_a^{new} \parallel HPW_i^{new}) \\ B_1^{new} &= h(HPID_i^{new} \parallel HPW_i^{new}) \end{aligned}$$

The smart card then securely replaces the old stored values with:

$$SC^{new} = \langle B_1^{new}, h(\cdot), u_i \rangle$$

Algorithm 2 Login and Authentication Phase

1: **User Login and Verification**
2: User inputs ID_i , PW_i , and us_i
3: Computes $b_i = h(B_i)$
4: Computes $HPID_i = h(ID_i \parallel b_i)$
5: Computes $HPW_i = h(PW_i \parallel b_i)$
6: Computes $A_1 = h(HPW_i \parallel HPID_i \parallel us_i)$
7: Verifies if $A_1 = B_1^*$, then proceeds
8: **User Message Generation**
9: Generates $n_1 \in F_q$, records TS_1
10: Computes:
11: $C_0 = n_1 \cdot P$
12: $C_1 = n_1 \cdot PK_G$
13: $C_2 = C_1 \oplus SNID_k$
14: $C_3 = h(HID_a) \oplus h(C_1 \parallel TS_1)$
15: $C_4 = h(C_0 \parallel C_2 \parallel C_3 \parallel h(HID_a))$
16: Sends $\langle C_0, C_2, C_3, C_4, TS_1 \rangle$ to Gateway
17: **Gateway Verification**
18: Verifies freshness: $|TS_1 - TS_1^*| \leq \Delta T$
19: Computes:
20: $h(HID_a^*) = C_3 \oplus h(C_1^* \parallel TS_1)$
21: $SNID_k = C_1^* \oplus C_2$
22: $C_4^* = h(C_0 \parallel C_2 \parallel C_3 \parallel h(HID_a^*))$
23: If $C_4^* = C_4$ then:
24: $C_5 = h(HID_a^*) \oplus h(SSN_k \parallel GWID_j \parallel TS_2)$
25: $C_6 = h(SSN_k \parallel GWID_j \parallel C_0 \parallel C_5 \parallel TS_2 \parallel h(HID_a^*))$
26: Sends $\langle C_0, C_5, C_6, TS_2 \rangle$ to Sensor
27: **Sensor Node Verification and Key Generation**
28: Verifies freshness: $|TS_2 - TS_2^*| \leq \Delta T$
29: Computes:
30: $h(HID_a^{**}) = C_5 \oplus h(SSN_k \parallel GWID_j \parallel TS_2)$
31: $C_6^* = h(SSN_k \parallel GWID_j \parallel C_0 \parallel C_5 \parallel TS_2 \parallel h(HID_a^{**}))$
32: If $C_6^* = C_6$ then:
33: Generates $n_2 \in F_q$, records TS_3
34: $C_7 = n_2 \cdot P$
35: $S_{Key_S} = n_2 \cdot C_0$
36: $C_8 = h(C_0 \parallel SSN_k \parallel C_7 \parallel GWID_j \parallel TS_3)$
37: Sends $\langle C_7, C_8, TS_3 \rangle$ to Gateway
38: **Gateway Final Check and Relay**
39: Verifies freshness and computes $C_8^* = h(C_0 \parallel SSN_k \parallel C_7 \parallel GWID_j \parallel TS_3)$
40: If valid, computes $C_9 = h(C_7 \parallel SNID_k \parallel h(HID_a^*))$
41: Sends $\langle C_7, C_9, TS_4 \rangle$ to User
42: **User Final Verification**
43: Verifies freshness and computes $C_9^* = h(C_7 \parallel SNID_k \parallel h(HID_a^*))$
44: If $C_9^* = C_9$ then sets $S_{Key_U} = S_{Key_S}$

This completes the password update without server interaction, preserving privacy and ensuring forward secrecy.

4 Security Evaluation

To thoroughly assess the security guarantees of the proposed authentication protocol, we adopt a dual strategy that combines both informal reasoning and formal automated verification. These complementary approaches are well-established in the evaluation of cryptographic protocols and provide a solid foundation for validating resilience against a wide range of security threats. In the informal analysis, key protocol properties such as mutual authentication, session key confidentiality, and robustness against replay, impersonation, and offline guessing attacks are systematically examined. This conceptual review ensures that the protocol logic aligns with expected security goals in adversarial environments, especially within wireless and resource-constrained IoT settings.

For rigorous formal verification, we employ two prominent tools: **Scyther** and **ProVerif**. These tools enable symbolic modeling of protocol behavior and simulate a wide range of attack scenarios, including those involving active and passive adversaries. The Scyther tool is configured in both standard and compromise modes, allowing for exhaustive validation of protocol robustness in the presence of key leakage or insider threats. Meanwhile, ProVerif offers a powerful Dolev–Yao-based framework for verifying secrecy, authentication properties, and reachability goals. Through these automated analyses, we verify that the proposed protocol successfully preserves core security properties and mitigates critical threats such as man-in-the-middle attacks, identity spoofing, replay attempts, and credential exposure. The consistency of results across both tools further supports the correctness of the protocol design. By integrating both manual inspection and formal tool-based analysis, our evaluation demonstrates that the protocol achieves strong security guarantees without imposing excessive computational or communication burdens, making it suitable for real-world deployment in healthcare-focused IoT systems.

4.1 Informal Security Analysis

This section evaluates the proposed authentication protocol (AEBAP) in terms of its ability to resist common attacks and satisfy critical security requirements. The analysis focuses on well-known adversarial threats that may arise during authentication processes in IoT-based healthcare environments, including both network-based and device-level attacks.

4.1.1 Anonymity and Untraceability

AEBAP incorporates user-specific cryptographic transformations to conceal the actual identity of IoT devices and users. By leveraging secure hash functions and integrating user-side secrets during message generation, it becomes computationally infeasible for an adversary to link transmitted messages to specific identities. The scheme ensures that intercepted data does not expose identifiable information, thus preserving user

privacy and preventing targeted attacks. This mechanism also enhances unlinkability across sessions, ensuring that communication flows cannot be traced to specific users or devices.

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

Although an adversary may attempt to intercept transmitted messages such as $\{C_0, C_2, C_3, C_4, TS_1\}$ during user–gateway communication, the use of timestamp validation, ECC point generation, and cryptographic hashing ensures that any altered messages will be detected upon arrival. Consequently, the protocol effectively prevents adversaries from modifying or injecting messages without being identified, providing robust defense against MITM attacks.

4.1.3 Session Key Disclosure Attack

In AEBAP, the session key S_{Key} is freshly derived in each authentication session using nonces and temporary secrets contributed independently by the user, gateway, and sensor node. As each session uses unique, non-reusable values, compromising one session key does not enable an adversary to derive keys from past or future sessions. This mitigates the risk of key exposure attacks and ensures that long-term session secrecy is preserved.

4.1.4 Replay Attack

The protocol integrates strict timestamp checking across all authentication phases, including the use of TS_1 through TS_4 . Upon message receipt, the timestamp is validated against the local clock, and the message is discarded if the time difference exceeds a predefined threshold ΔT . This mechanism effectively renders replayed messages invalid and protects the protocol from being exploited through replay-based intrusion attempts.

4.1.5 Forward and Backward Secrecy

AEBAP guarantees both forward and backward secrecy by ensuring that session keys are derived from ephemeral secrets that are not reused across sessions. Even if an adversary compromises long-term secrets such as us_i or internal device keys, they cannot infer prior or future session keys due to the randomness and uniqueness of values used in each key derivation process. This protection is critical for systems handling sensitive healthcare data.

4.1.6 Impersonation Attack

The protocol resists impersonation by employing one-way hash functions and ECC-based point operations that mask input values and prevent reverse computation. Additionally, XOR operations are used to obfuscate internal computations, preventing adversaries from extracting meaningful information even from observed values. These design choices prevent attackers from forging legitimate-looking authentication messages.

4.1.7 Denial of Service (DoS) Attack

To mitigate DoS threats, AEBAP employs challenge–response verification with strict timestamp constraints at each protocol layer. Sensor nodes validate received messages before engaging in further computation. Invalid or replayed messages are discarded early, conserving resources and reducing the attack surface. This lightweight validation mechanism ensures that devices remain responsive under adversarial load.

4.1.8 Three-Party Session Key Agreement

The protocol facilitates a secure three-party session key establishment between the user, gateway, and sensor node. Each party contributes its own private input (e.g., n_1 , n_2), and the resulting session key S_{Key} is derived in such a way that no single entity can unilaterally determine or predict the key. This ensures confidentiality, mutual authentication, and synchronized trust among all participants in the session.

4.1.9 Bypassing Attack

AEBAP resists cross-protocol attacks by embedding protocol-specific elements such as device identifiers, hash-based transformations, and ECC-derived points into its message formats. These elements tightly bind each message to the context of AEBAP, preventing attackers from reusing messages in other authentication protocols or tricking legitimate nodes into accepting malformed inputs.

4.1.10 Desynchronization Attack

Due to the use of timestamps and tight coupling between message values, AEBAP inherently resists desynchronization attacks. Any deviation in expected values resulting from delayed or modified messages is immediately detected by the receiving node. The protocol ensures state consistency by validating all critical parameters before proceeding, thereby maintaining synchronization between nodes during and across sessions.

4.2 Formal Assessment

4.2.1 Formal Security Analysis Using Scyther

Performing a rigorous formal analysis is vital to substantiate the security strength of cryptographic protocols. Scyther is one of the most prominent tools used for such purposes, offering symbolic verification based on the well-known Dolev-Yao adversarial model [Dolev and Yao \(1983\)](#). This framework enables automated exploration of protocol executions and systematically detects potential security violations by simulating adversarial interactions. Protocols are modeled in Scyther using the Security Protocol Description Language (SPDL), where roles, message exchanges, and security properties are clearly defined. The tool exhaustively analyzes all feasible execution traces and highlights violations through explicit attack traces, illustrating how an adversary might compromise the system. It verifies essential security guarantees, including:

- **Secrecy:** Protection of confidential data from unauthorized disclosure.

- **Authentication:** Assurance that only legitimate participants are involved in the protocol.
- **Message Integrity:** Prevention of tampering or unauthorized modification of transmitted messages.

Scyther adheres to a symbolic model where cryptographic operations are treated as idealized black boxes, and protocol behavior is captured through structured role-based communication. This abstraction enables precise reasoning about adversarial capabilities without relying on specific algorithmic assumptions. In this work, Scyther was used to formally evaluate the security properties of the proposed authentication scheme. The protocol was modeled with two key roles: the user and the server, capturing the core message exchanges and claims relevant to mutual authentication and confidentiality. The analysis verified the protocol’s resistance against key security threats, including replay, impersonation, and session key leakage.

To perform a comprehensive analysis, two operating modes of Scyther were employed:

- **Standard Mode:** Evaluates the protocol under standard adversarial assumptions.
- **Compromise Mode:** Simulates more powerful adversaries capable of performing long-term key exposure, partial protocol compromise, and key reveal attacks.

The results of the analysis demonstrated that the proposed protocol remains secure even in the presence of advanced threat models. Figures 1 and 2 display the configuration parameters used for the compromise and standard modes, respectively, highlighting the extended attack surface considered in the compromise scenario. Additionally, the protocol’s robustness against advanced threats—such as Post-Compromise Forward Secrecy (PFC) and session key exposure—is illustrated in Figures 3 and 4. The outcomes from the standard security validation are shown in Figure 5, confirming the protocol’s compliance with key security objectives. Overall, the formal analysis confirms that the proposed authentication protocol withstands critical adversarial scenarios. It ensures secure mutual authentication, confidentiality, and resistance to common attack vectors such as replay, key compromise impersonation, and session key disclosure, thereby validating its suitability for deployment in sensitive and resource-constrained environments.

4.2.2 Security Verification Using ProVerif

To comprehensively validate the security guarantees of the proposed AEBAP protocol, we utilized **ProVerif**, a widely recognized formal verification tool that enables symbolic modeling of cryptographic protocols under adversarial conditions. ProVerif provides rigorous validation of properties such as secrecy, integrity, and mutual authentication using an abstract representation of cryptographic operations. The protocol was modeled in such a way that the user enrollment phase occurs over secure channels, while the authentication and session key agreement phases were simulated over public channels to reflect real-world vulnerabilities. The Dolev-Yao adversarial model, which assumes an attacker with full control over the communication medium, was applied to analyze the protocol’s resilience to both passive and active threats.

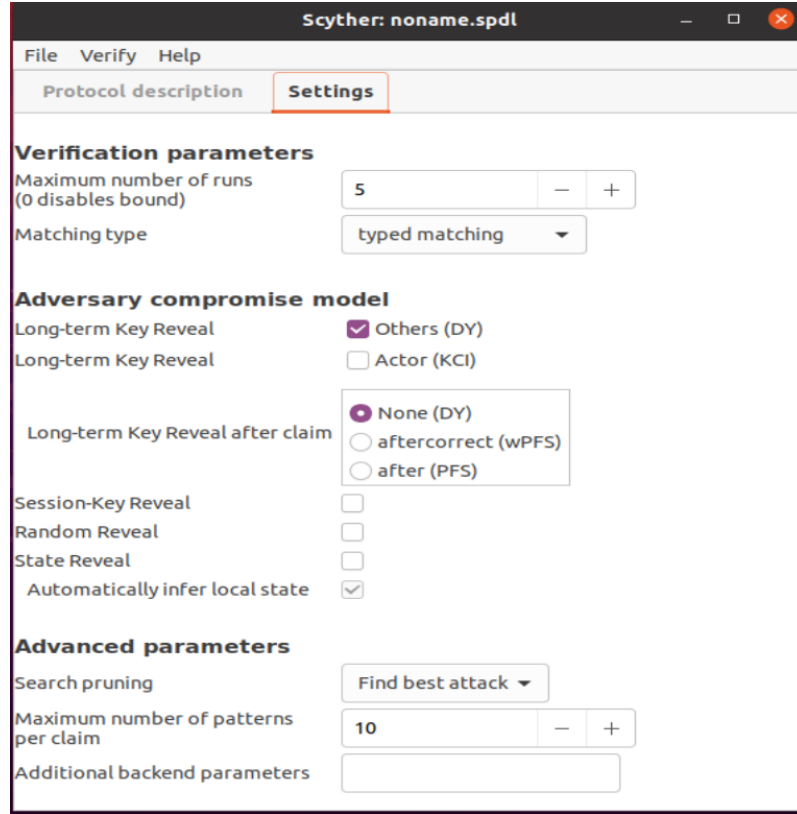


Figure 1 Configuration interface of Scyther tool in Compromise Mode (Version 0.9.2)

Figure 6 presents the verification output, which confirms that the protocol preserves the intended security goals. The following observations summarize the results:

1. **Confidentiality of Weak Secrets:** ProVerif verified that several critical elements remain confidential throughout protocol execution. In particular, the analysis confirmed the secrecy of:
 - User identity ID_i
 - Password PW_i
 - Private user randomness u_i
 - Random nonces n_1 and n_2
 - Sensor node key SSN_k
 - Gateway identifier $GWID_j$

These confirmations—denoted by `Weak secret ... is true` in the output—demonstrate that the protocol successfully protects sensitive data from disclosure under adversarial observation.

2. **Prevention of Unauthorized Access:** ProVerif’s output showed no evidence that the attacker could derive any secret keys, credentials, or protocol-specific

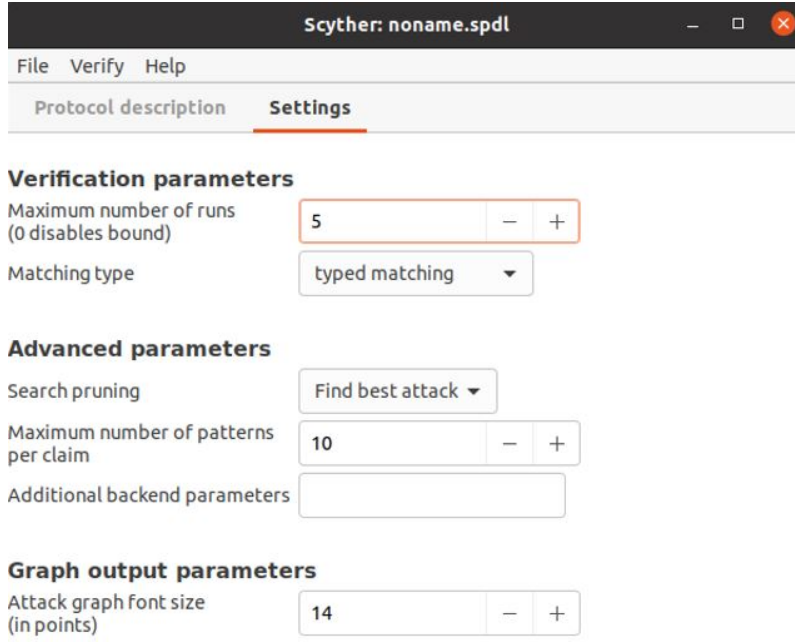


Figure 2 Standard configuration interface of the Scyther tool for protocol verification

parameters. The absence of violated secrecy claims confirms that the scheme enforces access control effectively, even in scenarios involving message interception or manipulation.

3. **Event-Based Authentication Validation:** The tool also evaluated mutual authentication between all entities involved in the protocol. The following injection queries were all satisfied:

- `inj-event(endUserA) ==> inj-event(beginUserA)`
- `inj-event(endGatewayNodeA) ==> inj-event(beginGatewayNodeA)`
- `inj-event(endSensorNodeA) ==> inj-event(beginSensorNodeA)`

These results confirm that each authentication session successfully links initiation and termination events for the user, gateway, and sensor, ensuring that no impersonation or message tampering occurred during session execution. Collectively, the ProVerif analysis demonstrates that the AEBAP protocol upholds strong security properties—including confidentiality, mutual authentication, and resistance to key compromise—under symbolic formal analysis. These results affirm its suitability for deployment in privacy-sensitive and adversarial IoT environments, such as healthcare networks.

Scyther results : verify

Claim				Status	Comments
userandserver	User	userandserver,User	Secret ru	Ok	No attacks within bound
		userandserver,User1	Nisynch	Ok	No attacks within bound
		userandserver,User2	Alive	Ok	No attacks within bound
		userandserver,User3	Weakagree	Ok	No attacks within bound
	Server	userandserver,Server	Secret rs	Ok	No attacks within bound
		userandserver,Server1	Nisynch	Ok	No attacks within bound
		userandserver,Server2	Alive	Ok	No attacks within bound
		userandserver,Server3	Weakagree	Ok	No attacks within bound

Done.

Figure 3 Security evaluation of the proposed protocol using Scyther in Compromise Mode under the "Post-Compromise Forward Secrecy" setting

Scyther results : verify

Claim				Status	Comments
userandserver	User	userandserver,User	Secret ru	Ok	No attacks within bound
		userandserver,User1	Nisynch	Ok	No attacks within bound
		userandserver,User2	Alive	Ok	No attacks within bound
		userandserver,User3	Weakagree	Ok	No attacks within bound
	Server	userandserver,Server	Secret rs	Ok	No attacks within bound
		userandserver,Server1	Nisynch	Ok	No attacks within bound
		userandserver,Server2	Alive	Ok	No attacks within bound
		userandserver,Server3	Weakagree	Ok	No attacks within bound

Done.

Figure 4 Security analysis of the proposed scheme under the "Session Key Reveal" setting in Scyther Compromise Mode

5 Assessment and Comparison

5.1 Security and Performance Evaluation: Comparative Analysis of the Proposed Authentication Protocol

This section provides a detailed comparative analysis between the proposed authentication scheme and several recently published protocols. The evaluation focuses on

Scyther results : verify

Claim				Status	Comments
userandserver	User	userandserver,User	Secret ru	Ok	No attacks within bound
		userandserver,User1	Nisynch	Ok	No attacks within bound
		userandserver,User2	Alive	Ok	No attacks within bound
		userandserver,User3	Weakagree	Ok	No attacks within bound
	Server	userandserver,Server	Secret rs	Ok	No attacks within bound
		userandserver,Server1	Nisynch	Ok	No attacks within bound
		userandserver,Server2	Alive	Ok	No attacks within bound
		userandserver,Server3	Weakagree	Ok	No attacks within bound

Done.

Figure 5 Security validation of the proposed protocol using the standard configuration of Scyther

Verification summary:

- Weak secret IDi is **true**.
- Weak secret PWi is **true**.
- Weak secret ui is **true**.
- Weak secret n1 is **true**.
- Weak secret n2 is **true**.
- Weak secret SSNK is **true**.
- Weak secret GWIDj is **true**.
- Query inj-event(endUserA) ==> inj-event(beginUserA) is **true**.
- Query inj-event(endGatewayNodeA) ==> inj-event(beginGatewayNodeA) is **true**.
- Query inj-event(endSensorNodeA) ==> inj-event(beginSensorNodeA) is **true**.

Figure 6 Automated security verification of the proposed authentication scheme using ProVerif tool

multiple key dimensions, including security resilience, communication cost, computational overhead, and memory requirements. These criteria are essential for assessing the protocol's applicability in environments with limited resources. From a communication standpoint, the comparison considers the number of messages exchanged during the authentication and session key establishment phases. Reducing message exchange is particularly advantageous in IoT systems, where communication bandwidth is often restricted.

Computational performance is assessed by examining the cryptographic operations utilized in each protocol, such as hash functions, XOR computations, and elliptic curve operations. Protocols with lower computational demands are better suited for deployment on devices with limited processing capabilities. The findings of this evaluation highlight the proposed protocol's ability to strike an effective balance between strong security guarantees and lightweight performance. By minimizing overhead while preserving robustness, the protocol demonstrates its potential for practical deployment in constrained IoT-based environments.

5.2 Evaluating Security Robustness: The Proposed Authentication Protocol vs. Contemporary Authentication Methods

This section provides a comparative assessment of the security features offered by the proposed authentication protocol in relation to several recent approaches, as outlined in Aghili et al. (2019b); Li et al. (2019); Fariss et al. (2022); Fotouhi et al. (2020); Wu et al. (2023). The analysis focuses on the protocol’s effectiveness in addressing key security challenges such as replay attacks, unauthorized privilege escalation, session key disclosure, user traceability, the absence of forward secrecy, and desynchronization vulnerabilities. In contrast, the referenced protocols reveal varying levels of susceptibility to one or more of these threats, which undermines their applicability in real-world, adversarial environments. As demonstrated in Table 2, the proposed scheme consistently offers stronger protection across all considered dimensions, highlighting its robustness and its suitability for deployment in lightweight, security-sensitive IoT infrastructures.

Table 2 Security Comparison of the Proposed Protocol with Recent Authentication Schemes

Protocols	A_1	A_2	A_3	A_4	A_5	A_6	A_7
Aghili et al. (2019b)	✓	✗	✓	✓	✗	✓	✓
Li et al. (2019)	✓	✗	✓	✗	✓	✓	✓
Fotouhi et al. (2020)	✗	✓	✗	✗	✗	✓	✓
Fariss et al. (2022)	✓	✗	✓	✓	✓	✗	✗
Wu et al. (2023)	✗	✓	✗	✗	✗	✓	✓
Huang (2024)	✗	✓	✗	✗	✗	✓	✓
AEBAF	✓	✓	✓	✓	✓	✓	✓

A_1 : Resistance to replay attacks A_2 : Protection against insider threats
 A_3 : Prevention of session key leakage A_4 : Support for user untraceability
 A_5 : Defense against impersonation attacks A_6 : Provision of forward secrecy
 A_7 : Robustness against desynchronization attacks
 ✓: Security feature supported ✗: Security feature not supported

Table 3 Notation and Metrics for Evaluating Computational and Communication Overhead

Symbol	Description	Execution Time	Transmission Size
T_h	Time required to compute a single cryptographic hash operation	3 ms	256 bits
$T_{en/d}$	Time consumed for executing symmetric encryption or decryption	3.7 ms	128 bits
T_{tc}	Bit length of the timestamp embedded in protocol messages	–	32 bits

5.3 Comparison of Communication and Computational Overhead

Performance Evaluation. According to benchmark results referenced in [Servati et al. \(2025\)](#), the average execution time for a cryptographic hash function (T_h) is approximately 3 ms, while symmetric encryption or decryption operations ($T_{en/d}$) require around 3.7 ms. As shown in Table 3, the proposed authentication scheme involves a total of 16 hash computations and 6 elliptic curve point multiplications. Assuming a point multiplication takes roughly 21 ms [Servati et al. \(2025\)](#), the cumulative computational cost of the proposed protocol amounts to approximately 150 ms. A detailed comparison of this computational overhead with that of recently published schemes is summarized in Table 4. The results clearly demonstrate that the proposed protocol offers a favorable balance between security strength and processing efficiency, making it suitable for resource-constrained environments.

In terms of communication efficiency, the total number of bits exchanged during the authentication process is evaluated. The overall communication overhead consists of 128 bits for identity transmission, 32 bits for timestamps, 128 bits for encryption or decryption data, 256 bits for random nonces, and 256 bits for hash outputs. This results in a total communication cost of approximately 3200 bits for the proposed protocol. A comparative breakdown is presented in Table 5, showing that the proposed scheme maintains competitive communication efficiency while delivering strong security assurances.

Table 4 Comparative Analysis of Computational Overhead Between the Proposed Authentication Scheme and Contemporary Protocols (in ms)

Protocols	Overall Computational Cost for entities in ms
Aghili et al. (2019b)	$(27 + 3m)T_h + 1T_{bh} = 176$ ms where $m = 10$ is the number of sensors
Li et al. (2019)	$22T_h + 4T_{mu} + T_{E/D} = 157.4$ ms
Fotouhi et al. (2020)	$32T_h = 96$ ms
Huang (2024)	$48T_h + 12T_{mu} = 396$ ms
AEBA	$16T_h + 4T_{mu} = 150$ ms

Table 5 Communication Cost Comparison Between the Proposed Authentication Protocol and Recently Published Schemes

Protocols	Year	Total Communication Overhead Between entities(in bits)
Aghili et al. (2019b)	2019	3456
Li et al. (2019)	2019	3328
Fotouhi et al. (2020)	2020	3200
Huang (2024)	2024	5644
AEBA	–	3200

6 Conclusion

This study presents AEBAP, a secure and full fledged authentication protocol tailored for IoT-based environments, with a specific focus on healthcare applications. The protocol facilitates mutual authentication and secure key establishment among three core entities: the user, the gateway, and the sensor node. By integrating elliptic curve cryptography with efficient cryptographic hash functions, AEBAP ensures the confidentiality, integrity, and authenticity of data exchanged across potentially insecure wireless channels. The proposed scheme has been rigorously evaluated through formal and automated security verification tools, including Scyther and ProVerif, to validate its resistance against various attack vectors such as impersonation, replay, session key disclosure, and desynchronization. Additionally, the protocol demonstrates low computational and communication overhead, making it a practical candidate for deployment in resource-constrained IoT infrastructures.

Despite its strong security guarantees and operational efficiency, future challenges remain regarding integration with heterogeneous IoT platforms and large-scale deployment scenarios. Moreover, extending the scheme to support dynamic node registration, trust management, and adaptive authentication remains an open area of research. Future directions will explore the integration of AEBAP with emerging security paradigms, including blockchain-based decentralized identity management and AI-driven anomaly detection. Further empirical testing in diverse real-world IoT environments—such as smart healthcare systems, industrial control networks, and smart cities—will also be pursued to assess usability, scalability, and long-term resilience.

References

- Aghili SF, Mala H, Kaliyar P, Conti M. SecLAP: Secure and lightweight RFID authentication protocol for Medical IoT. *Future Generation Computer Systems*. 2019;101:621–634.
- 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 generation computer systems*. 2019;96:410–424.
- Alzahrani BA, Irshad A, Albeshri A, Alsubhi K. A provably secure and lightweight patient-healthcare authentication protocol in wireless body area networks. *Wireless Personal Communications*. 2021;117(1):47–69.
- Arslan A, Bingöl MA. Security and Privacy Analysis of Recently Proposed ECC-Based RFID Authentication Schemes. *Cryptology ePrint Archive*. 2022;.
- Chander B, Gopalakrishnan K. A secured and lightweight RFID-tag based authentication protocol with privacy-preserving in Telecare medicine information system. *Computer Communications*. 2022;191:425–437.
- Chatterjee U, Ray S, Adhikari S, Khan MK, Dasgupta M. An improved authentication and key management scheme in context of IoT-based wireless sensor network using

- ECC. *Computer Communications*. 2023;209:47–62.
- Chen Y, Zhang N, Zhang Y, Chen X, Wu W, Shen X. Energy efficient dynamic offloading in mobile edge computing for internet of things. *IEEE Transactions on Cloud Computing*. 2019;9(3):1050–1060.
- Dolev D, Yao A. On the security of public key protocols. *IEEE Transactions on information theory*. 1983;29(2):198–208.
- Fariss M, El Gafif H, Toumanari A. A lightweight ECC-based three-factor mutual authentication and key agreement protocol for WSNs in IoT. *International Journal of Advanced Computer Science and Applications*. 2022;13(6).
- Fotouhi M, Bayat M, Das AK, Far HAN, Pournaghi SM, Doostari MA. A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT. *Computer Networks*. 2020;177:107333.
- Gabsi S, Kortli Y, Beroulle V, Kieffer Y, Alasiry A, Hamdi B. Novel ECC-based RFID mutual authentication protocol for emerging IoT applications. *IEEE Access*. 2021;9:130895–130913.
- Gupta A, Tripathi M, Shaikh TJ, Sharma A. A lightweight anonymous user authentication and key establishment scheme for wearable devices. *Computer Networks*. 2019;149:29–42.
- Hajian R, ZakeriKia S, Erfani SH, Mirabi M. SHAPARAK: Scalable health-care authentication protocol with attack-resilience and anonymous key-agreement. *Computer Networks*. 2020;183:107567.
- Hosseinzadeh M, Servati MR, Rahmani AM, Safkhani M, Lansky J, Janoscova R, et al. An Enhanced Authentication Protocol Suitable for Constrained RFID Systems. *IEEE Access*. 2024;.
- Huang W. ECC-based three-factor authentication and key agreement scheme for wireless sensor networks. *Scientific Reports*. 2024;14(1):1787. <https://doi.org/10.1038/s41598-024-52134-z>.
- Jegadeesan S, Azees M, Rajasekaran AS, Al-Turjman F. Lightweight Privacy and Confidentiality Preserving Anonymous Authentication Scheme for WBANs. *IEEE Trans Ind Informatics*. 2022;18(5):3484–3491. <https://doi.org/10.1109/TII.2021.3097759>, <https://doi.org/10.1109/TII.2021.3097759>.
- Jia X, Luo M, Wang H, Shen J, He D. A blockchain-assisted privacy-aware authentication scheme for Internet of Medical Things. *IEEE Internet of Things Journal*. 2022;9(21):21838–21850.

- Jiang Q, Ma J, Wei F, Tian Y, Shen J, Yang Y. An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks. *Journal of Network and Computer Applications*. 2016;76:37–48.
- 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.
- Khan N, Zhang J, Mallah GA, Chaudhry SA. A Secure and Efficient Information Authentication Scheme for E-Healthcare System. *Computers, Materials & Continua*. 2023;76(3).
- Kumar P, Pal AK, Islam SH. 2F-MASK-VSS: Two-factor mutual authentication and session key agreement scheme for video surveillance system. *Journal of Systems Architecture*. 2024;p. 103196.
- Kumar V, Ahmad M, Kumari A. A secure elliptic curve cryptography based mutual authentication protocol for cloud-assisted TMIS. *Telematics and Informatics*. 2019;38:100–117.
- 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 Systems Journal*. 2019;14(1):39–50.
- Oh J, Yu S, Lee J, Son S, Kim M, Park Y. A secure and lightweight authentication protocol for IoT-based smart homes. *Sensors*. 2021;21(4):1488.
- Pu C, Zerkle H, Wall A, Lim S, Choo KKR, Ahmed I. A lightweight and anonymous authentication and key agreement protocol for wireless body area networks. *IEEE Internet of Things Journal*. 2022;9(21):21136–21146.
- Rani D, Tripathi S. Design of blockchain-based authentication and key agreement protocol for health data sharing in cooperative hospital network. *The Journal of Supercomputing*. 2024;80(2):2681–2717.
- Sadhukhan D, Ray S, Biswas G, Khan MK, Dasgupta M. A lightweight remote user authentication scheme for IoT communication using elliptic curve cryptography. *The Journal of Supercomputing*. 2021;77(2):1114–1151.
- Safkhani M, Rostampour S, Bendavid Y, Bagheri N. IoT in medical & pharmaceutical: Designing lightweight RFID security protocols for ensuring supply chain integrity. *Computer Networks*. 2020;181:107558.
- Safkhani M, Vasilakos A. A new secure authentication protocol for telecare medicine information system and smart campus. *IEEE Access*. 2019;7:23514–23526.
- Servati MR, Safkhani M. ECCbAS: An ECC based authentication scheme for healthcare IoT systems. *Pervasive and Mobile Computing*. 2023;p. 101753. <https://doi.org/10.1016/j.pmc.2023.101753>

[//www.sciencedirect.com/science/article/pii/S1574119223000111](https://www.sciencedirect.com/science/article/pii/S1574119223000111), <https://doi.org/https://doi.org/10.1016/j.pmcj.2023.101753>.

- Servati MR, Safkhani M, Ali S, Malik MH, Ahmed OH, Hosseinzadeh M, et al. Cryptanalysis of Two Recent Ultra-Lightweight Authentication Protocols. *Mathematics*. 2022;10(23):4611.
- Servati MR, Safkhani M, Rahmani AM, Hosseinzadeh M. ERASMIS: An ECC-based robust authentication protocol suitable for medical IoT systems. *Computer Networks*. 2025;258:110938. <https://www.sciencedirect.com/science/article/pii/S1389128624007709>, <https://doi.org/https://doi.org/10.1016/j.comnet.2024.110938>.
- Shariq M, Singh K, Maurya PK, Ahmadian A, Ariffin MRK. URASP: An ultra-lightweight RFID authentication scheme using permutation operation. *Peer-to-Peer Networking and Applications*. 2021;14:3737–3757.
- Shuai M, Yu N, Wang H, Xiong L. Anonymous authentication scheme for smart home environment with provable security. *Computers & Security*. 2019;86:132–146.
- Suganthi S, Anitha R, Sureshkumar V, Harish S, Agalya S. End to end light weight mutual authentication scheme in IoT-based healthcare environment. *Journal of Reliable Intelligent Environments*. 2020;6(1):3–13.
- Sureshkumar V, Amin R, Vijaykumar V, Sekar SR. Robust secure communication protocol for smart healthcare system with FPGA implementation. *Future Generation Computer Systems*. 2019;100:938–951.
- Wang W, Yan B, Chai B, Shen R, Dong A, Yu J. EBIAS: ECC-enabled blockchain-based identity authentication scheme for IoT device. *High-Confidence Computing*. 2024;p. 100240.
- Wang X, Fan K, Yang K, Cheng X, Dong Q, Li H, et al. A new RFID ultra-lightweight authentication protocol for medical privacy protection in smart living. *Computer Communications*. 2022;186:121–132.
- Wang Y, Liu Y. RC2PAS: Revocable Certificateless Conditional Privacy-Preserving Authentication Scheme in WBANs. *IEEE Systems Journal*. 2022;16(4):5675–5685.
- 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). <https://www.mdpi.com/2227-7390/11/17/3701>, <https://doi.org/10.3390/math11173701>.
- Xiang A, Zheng J. A situation-aware scheme for efficient device authentication in smart grid-enabled home area networks. *Electronics*. 2020;9(6):989.

Xu Z, Xu C, Chen H, Yang F. A lightweight anonymous mutual authentication and key agreement scheme for WBAN. *Concurrency and computation: Practice and experience*. 2019;31(14):e5295.

Zheng L, Song C, Cao N, Li Z, Zhou W, Chen J, et al. A new mutual authentication protocol in mobile RFID for smart campus. *IEEE Access*. 2018;6:60996–61005.