Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

JOURNAL OF INFORMATION SYSTEM AND TECHNOLOGY MANAGEMENT (JISTM)
www.gaexcellence.com/jistm

A LIGHTWEIGHT ECC-BASED AUTHENTICATION AND KEY AGREEMENT PROTOCOL FOR RESOURCE-CONSTRAINED IOT

Nusrat Hamid Shah[1]*, Saiful Adli Ismail[2], Azizul Azizan[3]

[1]Faculty of Artificial Intelligence, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia
snusrat01@gmail.com https://orcid.org/0000-0002-1022-4150
[2]Faculty of Artificial Intelligence, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia
saifuladli@utm.my https://orcid.org/0000-0002-9299-5652
[3]Faculty of Artificial Intelligence, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia
azizulazizan@utm.my https://orcid.org/0000-0002-5331-6071
*Corresponding Author

Article Info:

Article history:
Received date: 03.05.2026
Revised date: 24.05.2026
Accepted date: 25.06.2026
Published date: 30.06.2026

To cite this document:

Shah, N. H., Ismail, S. A., & Azizan, A. (2026). A Lightweight ECC-Based Authentication and Key Agreement Protocol for Resource-Constrained IoT. *Journal of Information System and Technology Management, 11* (43), 276-300.

Abstract:

The rapid proliferation of the Internet of Things (IoT) has connected billions of devices that continuously exchange sensitive data over public wireless channels, making secure authentication a critical requirement. However, most IoT end devices are severely resource-constrained, which renders conventional public-key infrastructures based on RSA, bilinear pairings, or digital certificates impractical. Although many authentication and key agreement schemes have been proposed, recent cryptanalytic studies show that they often either omit essential security properties such as perfect forward secrecy or resistance to ephemeral secret leakage or incur overhead that is unacceptable for low-power nodes. To address this gap, this paper proposes a lightweight Elliptic Curve Cryptography (ECC)-based authentication and key agreement protocol for a three-tier IoT architecture consisting of a user, a semi-trusted gateway server, and an IoT sensor. The protocol relies solely on elliptic-curve scalar multiplication, one-way hash functions, and bitwise XOR operations, avoiding bilinear pairings and certificate management. It achieves mutual authentication among all three parties through timestamped, hash-based authenticators and establishes a contributory session-key derived from fresh ephemeral keys, ensuring perfect forward secrecy. The security of the protocol is established through a detailed informal analysis, demonstrating resistance to major attacks, and is formally verified using the AVISPA tool, where both the OFMC and CL-AtSe back-ends report the protocol as SAFE under the Dolev–Yao adversary model. A comparative performance evaluation against several recent ECC-based authentication schemes shows that the

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

proposed protocol maintains low cost requirements suitable for constrained IoT devices, and resulting in a communication overhead of 1828 bits, which is competitive with most recent ECC-based schemes evaluated, and imposing a minimal storage burden on end devices. The results demonstrate that the proposed protocol achieves a favorable balance between security and efficiency, making it well suited to resource-constrained IoT environments.

DOI: 10.35631/JISTM.1143016

Keyword:
Authentication And Key Agreement; AVISPA; Elliptic Curve Cryptography (ECC); Formal Verification; Internet of Things (IoT); Lightweight Security; Mutual Authentication; Perfect Forward Secrecy

Introduction

The Internet of Things (IoT) has evolved into a pervasive infrastructure connecting billions of physical devices to the digital world. Recent reports indicate that connected IoT devices reached 18.5 billion in 2024 and are expected to grow to 21.1 billion in 2025 (IoT Analytics, 2025), with projections exceeding 40.6 billion devices by 2034 (Transforma Insights & Exploding Topics, 2025). This rapid growth is driven by applications such as smart homes, healthcare, intelligent transportation, industrial automation, precision agriculture (Vangala et al., 2023), and smart-city services. In these domains, lightweight devices continuously exchange sensitive data over public wireless channels, making secure IoT communication a critical requirement. Despite their widespread deployment, most IoT devices are resource-constrained, operating with limited computation, storage, connectivity, and energy resources (Li & Hu, 2024). Traditional public-key infrastructures based on RSA, digital certificates, and bilinear pairings impose excessive overhead and are unsuitable for low-power devices ("A Lightweight Authentication and Key Agreement Protocol," 2023). Meanwhile, the open nature of wireless communication exposes IoT systems to attacks such as eavesdropping, replay, man-in-the-middle, impersonation, and session key compromise (Thakur et al., 2024). Unattended deployment also increases the risk of physical capture and ephemeral secret leakage attacks.

Authentication and key agreement (AKA) protocols therefore play a fundamental role in IoT security (Thakur et al., 2023). Before exchanging sensitive information, communicating entities must authenticate each other and establish a secure session key. However, IoT protocols must remain lightweight by minimizing computational, communication, and storage overhead while still providing strong security guarantees (Gautam et al., 2024). Achieving both security and efficiency remains a major research challenge.

Although numerous AKA protocols have been proposed, many suffer from important limitations (Shah et al., 2025). Password-based and smart-card schemes are vulnerable to offline guessing and stolen-verifier attacks (Wang & Wang, 2018), while symmetric-key approaches face scalability issues. Certificate- and pairing-based schemes introduce unacceptable overhead for constrained devices. Recent studies further reveal vulnerabilities in existing protocols, including man-in-the-middle, insider, impersonation, and secret-key leakage attacks (Thakur et al., 2024). Many protocols also fail to ensure perfect forward secrecy and resistance to ephemeral secret leakage attacks (Li & Hu, 2024).

Elliptic Curve Cryptography (ECC), introduced by Koblitz (1987) and Miller (1986), has emerged as a promising solution for IoT security. ECC provides strong security with significantly smaller key sizes, reducing computational, storage, and bandwidth requirements (Li & Hu, 2024). For example, a 256-bit ECC key offers security comparable to a 3072-bit RSA key while requiring lower computational cost. These characteristics make ECC highly suitable for resource-constrained IoT devices (Li & Hu, 2024). However, designing ECC-based AKA protocols that achieve both minimal overhead and strong security properties remains challenging.

Motivated by these limitations, this paper proposes a novel lightweight ECC-based authentication and key agreement protocol specifically designed for three-tier IoT environments comprising users, semi-trusted gateway servers and resource constrained sensor nodes. The proposed protocol provides mutual authentication and secure session-key establishment while minimizing computational and communication costs. Formal security verification is conducted using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool (Armando et al., 2005). In addition, the proposed scheme is compared with representative existing protocols (Gautam et al., 2024; “A Lightweight Authentication and Key Agreement Protocol,” 2023; Thakur et al., 2024) to evaluate its security and efficiency performance.

The main contributions of this paper are summarized as follows:

1. A lightweight ECC-based AKA protocol is proposed for resource-constrained IoT devices, relying solely on efficient elliptic-curve and one-way hash operations while avoiding expensive bilinear pairings and certificate management.
2. The protocol reduces computational and communication overhead by minimizing the number of costly cryptographic operations and protocol messages, making it practical for low-power IoT end-nodes.
3. The scheme guarantees mutual authentication and secure session-key establishment, enabling communicating entities to verify each other's legitimacy and jointly derive a fresh, confidential session key resistant to known attacks.
4. The security of the proposed protocol is formally verified using the AVISPA tool, confirming its safety against active attacks including replay and man-in-the-middle attacks under the Dolev–Yao adversary model.
5. A comprehensive performance comparison with state-of-the-art schemes demonstrates that the proposed protocol attains a superior balance between security strength and efficiency.

The remainder of this paper is organized as follows. Section 2 reviews related work on IoT authentication and key agreement. Section 3 presents the preliminaries and system model. Section 4 describes the proposed ECC-based protocol in detail. Section 5 provides the security

analysis and AVISPA-based formal verification. Section 6 reports the performance evaluation and comparison. Finally, Section 7 concludes the paper and outlines directions for future work.

Related Work

This section reviews prior research on authentication and key agreement for the IoT. We first survey the landscape of existing protocols, then contrast symmetric- and asymmetric-key approaches, examine ECC-based lightweight schemes and AVISPA-verified protocols, and finally identify the gaps that motivate the present work.

Overview of Existing IoT Authentication Protocols

Securing communication between resource-constrained end devices and gateways or servers has been an active research area for over a decade, and a wide variety of authentication and key agreement (AKA) protocols have been proposed for IoT and wireless sensor environments (Li & Hu, 2024; "Provably Secure ECC-Based Anonymous Authentication," 2024). Early proposals relied on passwords and smart cards, evolving subsequently into two- and three-factor schemes that combine passwords, smart cards or tokens, and biometrics (Thakur et al., 2023; Wang & Wang, 2018). More recent designs target specialized deployments such as industrial control systems, the Internet of Drones (Gautam et al., 2024), precision-agriculture networks (Vangala et al., 2023), multi-gateway architectures, and edge-assisted IoT. Despite their diversity, these protocols share a common goal: to establish mutual authentication and a secure session key before sensitive data is exchanged over an untrusted channel. However, surveys and cryptanalytic studies repeatedly observe that many proposed AKA schemes either do not cover the necessary security features or are incompatible with resource-constrained end devices (Shunfang et al., 2024), underscoring the persistent tension between security strength and efficiency.

Symmetric versus Asymmetric Authentication Schemes

IoT authentication protocols are generally classified based on their cryptographic primitives. Symmetric-key schemes use pre-shared secrets, hash functions, and symmetric ciphers (Shah et al., 2023). Although computationally efficient for low-power devices, they suffer from poor scalability, difficult key distribution and storage, and increased risk when a node is compromised. They also provide limited support for perfect forward secrecy and non-repudiation. Asymmetric (public-key) schemes overcome key-management and scalability issues while supporting mutual authentication, anonymity, and forward secrecy. However, traditional RSA- and bilinear pairing-based approaches introduce high computational and communication overhead unsuitable for resource-constrained IoT devices (Hammi et al., 2020). Certificate management further increases complexity, motivating certificateless and identity-based schemes such as LiKe (Tedeschi et al., 2020). Among asymmetric approaches, Elliptic Curve Cryptography (ECC) offers the best balance between security and efficiency by providing strong security with smaller key sizes and lower computational cost, making it highly suitable for IoT environments (Li & Hu, 2024).

ECC-Based Lightweight Authentication Studies

A substantial body of research has applied ECC to lightweight IoT authentication. Hammi et al. (2020) proposed an early ECC-based lightweight authentication scheme that reduced

computational cost compared with certificate-based approaches. Tedeschi et al. (2020) introduced LiKe, a lightweight certificateless key-agreement protocol that removes certificate-management overhead. Baccouri et al. (2024) presented a lightweight authentication scheme based on elliptic-curve ElGamal cryptography. More recently, Li and Hu (2024) proposed an ECC-based AKA protocol using dynamic authenticated credentials, achieving over 37% reduction in communication and computational costs, while Hu et al. (2025) designed an efficient ECC-based authentication protocol for semi-trusted IoT environments. The DEAC-IoT protocol ("DEAC-IoT," 2024) further demonstrated ECC-based authenticated key agreement for intra- and inter-device communication with formal security verification. Collectively, these studies show that ECC can provide secure and efficient authentication suitable for resource-constrained IoT devices.

Existing AVISPA-Verified Protocols

Informal security analysis alone cannot guarantee robustness; formal verification has become standard in modern AKA research. The Automated Validation of Internet Security Protocols and Applications (AVISPA) tool (Armando et al., 2005) is widely used to formally validate security protocols through HLPSL specification under the Dolev–Yao adversary model. Many IoT authentication protocols have been verified using AVISPA. For example, Thakur et al. (2024) validated an authenticated key agreement protocol for industrial sensor networks using both Real-or-Random proof analysis and AVISPA verification. Similarly, three-factor IoT authentication schemes and PUF-based protocols have employed AVISPA to demonstrate resistance against active and replay attacks. These studies highlight that AVISPA verification has become an essential component of credible AKA protocol design.

Limitations and Gaps in Prior Work

Despite recent progress, cryptanalytic studies continue to reveal weaknesses in existing schemes. Thakur et al. (2024) showed that several protocols remain vulnerable to man-in-the-middle, insider, secret-key-leakage, and impersonation attacks, while lacking proper session-key agreement. Park et al. (2025) further demonstrated that some ECC-based schemes fail to resist physical-capture and impersonation attacks due to exposed secret parameters. Other studies report vulnerabilities to key-compromise impersonation attacks and lack of perfect forward secrecy (Park et al., 2025; "Provably Secure ECC-Based Anonymous Authentication," 2024), whereas protocols resistant to ephemeral secret leakage often introduce high computational and communication costs (Park et al., 2025). Although many authentication and key agreement schemes have been proposed, recent cryptanalytic studies show that they either omit essential security properties such as perfect forward secrecy or resistance to secret leakage or incur overhead that is unacceptable for low-power IoT devices. Therefore, a lightweight and formally verified protocol that ensures resistance to ESL, impersonation, and man-in-the-middle attacks while providing mutual authentication, anonymity, and perfect forward secrecy remains an open research need. This paper addresses that gap.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

Table 1. Comparison Of Representative IoT Authentication and Key Agreement Schemes.

Study	Technique	Security features	Limitations
Hammi et al. (2020)	Lightweight ECC-based authentication	Mutual authentication; low computation cost vs. certificate-based schemes	Limited formal verification; does not explicitly address ESL attacks
Tedeschi et al. (2020)	Certificateless ECC key agreement	Removes certificate-management overhead; lightweight key agreement	Trusted-authority dependence; forward secrecy under specific assumptions
Thakur et al. (2024)	ECC-based AKA for industrial sensor networks	Mutual authentication; ROR proof; AVISPA-verified	Cryptanalysis of predecessor schemes reveals impersonation/MITM/key-leakage flaws to be avoided
Li and Hu (2024)	ECC-based AKA with dynamic authenticated credentials	Anonymity; ESL resistance; PFS; >37% overhead reduction	Credential-synchronization framework adds protocol complexity
(Shunfang et al., 2024)	ECC key exchange with hash-based authentication	Mutual authentication; user anonymity; forward security	Subsequent analysis (Park et al., 2025) reports physical-capture and impersonation vulnerabilities
Baccouri et al. (2024)	Elliptic-curve ElGamal lightweight authentication	Lightweight design; reduced complexity	Limited resistance analysis against ESL/KCI attacks
Proposed scheme	Lightweight ECC-based AKA	Mutual authentication; anonymity; PFS; ESL resistance; AVISPA-verified	Designed to overcome the above limitations

Preliminaries

This section introduces the cryptographic background, threat model, and security requirements that underpin the proposed protocol.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

Elliptic Curve Cryptography (ECC)

Basic concepts. Elliptic Curve Cryptography, independently introduced by Koblitz (1987) and Miller (1986), bases its security on the algebraic structure of elliptic curves over finite fields. Over a prime field Zp (where p is a large prime), a non-singular elliptic curve E is defined by the short Weierstrass equation E: $y^2 = x^3 + ax + b \pmod p$, where the coefficients a, b satisfies the non-singularity condition $4a^3 + 27b^2 \neq 0 \pmod p$. The set of all points (x, y) satisfying this equation, together with a special point at infinity O that serves as the identity element, forms an additive abelian group (Hankerson et al., 2004). A base point (generator) G of large prime order n is published as a public domain parameter.

Point multiplication. Two group operations are defined over E: point addition, which combines two distinct points, and point doubling. Repeated application of these operations yields scalar (point) multiplication, the central operation of ECC: $Q = k \cdot G$, where k is a scalar integer and Q is the resulting curve point. Point multiplication is efficient to compute, yet its inversion is believed to be computationally infeasible. This asymmetry is captured by the Elliptic Curve Discrete Logarithm Problem (ECDLP): given the points G and $Q = k \cdot G$, it is intractable to recover the scalar k. The proposed protocol also relies on the Elliptic Curve Computational Diffie–Hellman (ECCDH) assumption given $a \cdot G$ and $b \cdot G$ for unknown a, b, computing $ab \cdot G$ is infeasible.

Public/private key generation. Each participant generates an ECC key pair as follows. A random integer d in $[1, n-1]$ is selected as the private key, and the corresponding public key is computed by point multiplication: $Q = d \cdot G$. The public key Q may be disclosed openly, while d is kept secret. By the hardness of the ECDLP, an adversary observing Q and G cannot derive d.

Advantages of ECC over RSA. ECC provides a level of security equivalent to traditional public-key cryptosystems while using significantly smaller keys, because the best-known attacks on the ECDLP are fully exponential, whereas sub-exponential index-calculus attacks apply to RSA's integer-factorization problem (Hankerson et al., 2004). Consequently, a 256-bit ECC key offers security comparable to a 3072-bit RSA key. As a result, ECC offers strong security with smaller key sizes, thereby reducing computational, storage, and bandwidth requirements (Li & Hu, 2024). The smaller keys translate into lower memory footprints, shorter transmitted messages, reduced energy consumption, and faster cryptographic operations properties that make ECC the preferred public-key primitive for resource-constrained IoT devices (Hammi et al., 2020; “A Lightweight Authentication and Key Agreement Protocol,” 2023).

Threat Model

The security of the proposed protocol is analyzed under the Dolev–Yao adversary model (Dolev & Yao, 1983), where the adversary can intercept, modify, inject, replay, or delete messages over the public channel and may act as a malicious insider allowing the adversary to obtain ephemeral secrets and extract stored parameters from captured devices (Park et al., 2025). However, the adversary is assumed unable to solve the ECDLP or ECCDH problems or break the one-way hash function.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

Under these assumptions, the protocol is evaluated against replay, man-in-the-middle, impersonation, ephemeral secret leakage, privileged-insider, key-compromise impersonation, device-capture, offline-guessing, and traceability attacks (Li & Hu, 2024; Park et al., 2025).

Security Requirements

A robust AKA protocol for IoT must satisfy the following security and functional requirements (Li & Hu, 2024; "Provably Secure ECC-Based Anonymous Authentication," 2024; Thakur et al., 2024):

Mutual authentication. Both communicating entities must verify each other's legitimacy before a session is established, so that neither a malicious device nor a rogue server can complete the protocol.

Session-key agreement. Upon successful authentication, the two parties must jointly compute a fresh, shared session key for subsequent encrypted communication. The key must be contributory and known only to the legitimate participants.

User anonymity and untraceability. The real identity of an IoT device must never be transmitted in cleartext, and successive sessions of the same device must be unlinkable.

Perfect forward secrecy. The compromise of a participant's long-term private key must not endanger previously established session keys.

Resistance to replay and impersonation attacks. The protocol must reject replayed messages and prevent any adversary from masquerading as a legitimate device, gateway, or server; it should also withstand MITM, ESL, KCI, insider, and device-capture attacks.

These requirements jointly define the security target of the proposed protocol and form the basis for the formal analysis and AVISPA-based verification presented in Section 5. Figure 1 summarizes the five security requirements that the proposed protocol is designed to satisfy.

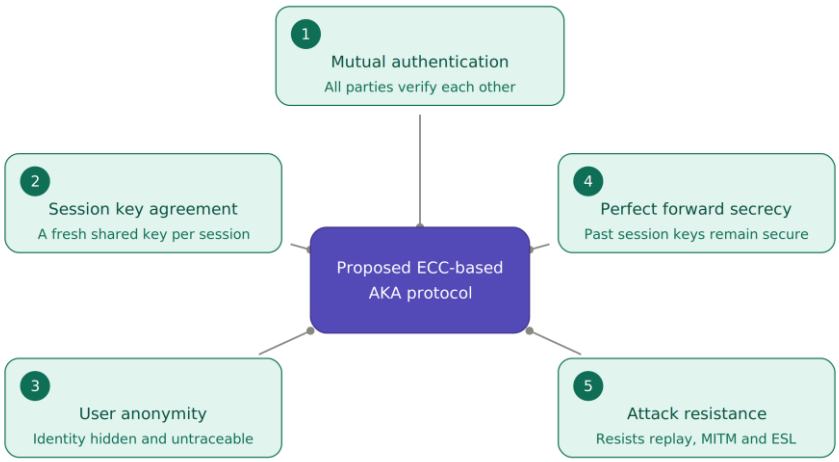

Figure 1. Security Requirements That the Proposed Lightweight ECC-Based AKA Protocol Must Satisfy

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

The Proposed Protocol

This section presents the proposed lightweight ECC-based authentication and key agreement (AKA) protocol. The protocol comprises three phases: a one-time system initialization phase, a registration phase in which users and sensors enrol with the server over a secure channel, and a login, authentication, and key agreement phase executed over the public channel.

System Model

The proposed protocol involves three types of entities:

- User (A). A human user, equipped with a smart card, who wishes to access real-time data from a remote IoT sensor. The user holds an identity and password and is registered with the server.
- Server (B). A trusted entity that acts as the registration authority and the authentication mediator. During the initialization and registration phases it generates the system parameters and enrols users and sensors over a secure channel; during the authentication phase it relays and verifies messages between the user and the sensor.

Sensor (C). A resource-constrained IoT sensor node that gathers and provides data. It registers with the server and, after mutual authentication, establishes a session key directly with the user. Registration is conducted over a secure channel, whereas all messages of the login and authentication phase travel over a public channel fully controlled by the Dolev–Yao adversary defined in Section 3.2. The objective of the protocol is to allow the user and the sensor, mediated by the server, to mutually authenticate one another and agree on a shared ECC-based session key SK. The system model is illustrated in Figure 2.

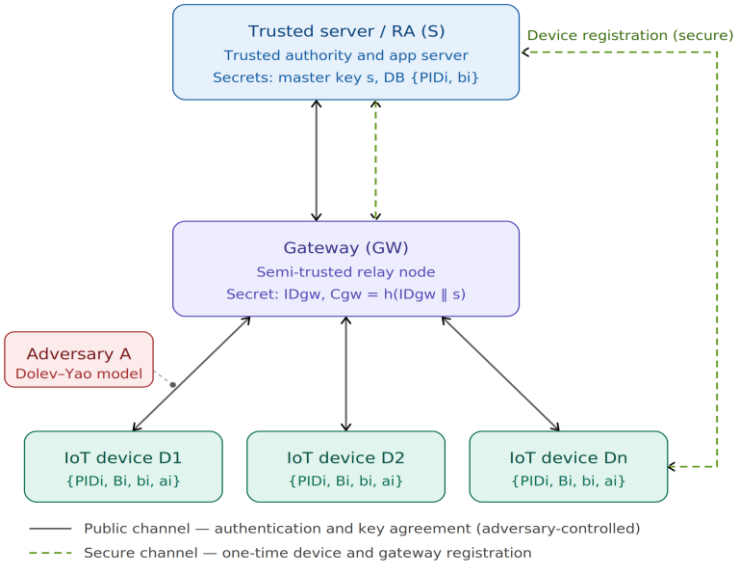

Figure 2. System Model of The Proposed Lightweight ECC-Based Authentication and Key Agreement Protocol

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

Notation

Table 2 lists the notation used throughout the protocol description.

Table 2. Notation Used in The Proposed Protocol.

Symbol	Description
A, B, C	User, server, and sensor entities
p, Zp	Large prime and the associated finite field
Ep(a, b)	Non-singular elliptic curve defined over Zp
G, n	Base point of Ep(a, b) and its large prime order
H(·)	Secure one-way (collision-resistant) hash function
xs, Ps	Server long-term ECC private and public key, Ps = xs · G
XGwn, XGwnS, XGwnU	Server long-term internal master secret values
S	Long-term pre-shared symmetric key
ID, PWD	User identity and password
SenID	Sensor identity
R	Fresh nonce (random number)
Ti	Fresh timestamp generated by an entity
ki, kj	Fresh ephemeral ECC private keys of the user and the sensor
Pi, Pj	Ephemeral ECC public keys (transmitted in compressed form), Pi = ki · G, Pj = kj · G
Kij, Kji	ECC shared secret points computed by the user and the sensor
SK	Established session key
SC	User smart-card storage
⊕, ‖	Bitwise XOR and concatenation operations
{M}S	Message M transmitted securely (encrypted under the pre-shared key S)

System Initialization Phase

The initialization phase is executed once by the trusted server before any user or sensor is deployed. It establishes the elliptic-curve domain parameters and the server's long-term secrets:

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

1. Select a large prime number p and define the finite field Zp.
2. Define a non-singular elliptic curve Ep(a, b) over Zp.
3. Choose a base point G ∈ Ep(a, b) of large prime order n.
4. Select a secure one-way hash function H(·).
5. Select the server's long-term ECC private key xs ∈ Zn* and compute the corresponding public key Ps = xs · G.
6. Generate the server's long-term internal master secret values XGwn, XGwnS, and XGwnU.
7. Generate the long-term pre-shared symmetric key S.
8. Verify that all selected parameters satisfy the elliptic-curve domain-parameter conditions.
9. Publish the public system parameters {Ep(a, b), p, G, n, H(·), Ps}.
10. Securely store the server's long-term private key xs, the master secret values {XGwn, XGwnS, XGwnU}, and the pre-shared key S.

The use of standardized, verified elliptic-curve domain parameters ensures that the underlying ECDLP and ECCDH assumptions (Section 3.1) hold, providing the cryptographic foundation for the protocol (Hankerson et al., 2004; Li & Hu, 2024). Throughout the protocol, all elliptic-curve public keys exchanged over the channel are represented in compressed form that is, by the x-coordinate of the point together with a single sign bit identifying the corresponding y-coordinate. Each recipient reconstructs the full curve point from the published domain parameters by solving the curve equation. Point compression is a standard, library-supported encoding convention adopted here to reduce transmission size; it does not alter the protocol logic, the messages exchanged, or the security properties analyzed in Section 5, and its effect on communication overhead is quantified in Section 6.

Registration Phase

The registration phase enrols users and sensors with the server. All messages are exchanged over a secure channel, denoted {·}S.

User Registration (A ↔ B)

1. User A selects an identity ID and password PWD, and generates a fresh nonce R.
2. The user computes the masked password MP = H(R ‖ PWD) and the masked identity MI = H(R ‖ ID).
3. The user securely sends {MI, MP}S to the server, encrypted under the pre-shared key S.
4. The user stores the password PWD locally.
5. The server B receives {MI, MP}S and retrieves the master secret values XGwn and XGwnU.
6. The server computes F = H(MI ‖ XGwn) and X = H(MP ‖ XGwnU).
7. The server computes the smart-card verifier E = F ⊕ X.
8. The server sends {H(R ‖ ID), E, F, XGwnU}S to the user.
9. The user stores the smart-card data SC = {R ‖ MI ‖ E ‖ F ‖ XGwnU}.

Because the identity and password are never transmitted or stored in clear text only in hashed and masked form the registration phase preserves user anonymity and provides resistance to stolen-verifier and offline-guessing attacks (Li & Hu, 2024; Wang & Wang, 2018).

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

Sensor Registration (C ↔ B)

1. Sensor C generates a fresh nonce R and timestamp T1 and uses its identity SenID.
2. The sensor computes MPs = H(S ‖ R ‖ SenID), MN = R ⊕ S, and RMP = MPs ⊕ MN.
3. The sensor sends {SenID, RMP, MN, T1}S to the server B.
4. The server computes F2 = H(SenID ‖ XGwn), X2 = H(MPs ‖ S), and E2 = F2 ⊕ X2.
5. The server sends {E2, F2, T2}S to the sensor.
6. The sensor stores RecE = E2 and RecF = F2.

Login, Authentication, and Key Agreement Phase

This phase runs over the public channel and consists of five sub-phases (A–E). It achieves mutual authentication among the user, server, and sensor and establishes a shared ECC session key. Each transmitted message carries a fresh timestamp; upon receipt, the recipient checks the timestamp for freshness against the maximum permissible delay and rejects stale messages, thwarting replay attacks. As established in Section 4.3, every elliptic-curve public key transmitted in the messages below (namely Pi and Pj) is sent in compressed form. The witness events recorded at each step correspond to the security goals later modelled and verified in AVISPA (Section 5).

Phase A — Login (User A → Server B).

1. User A generates a fresh timestamp T1.
2. The user selects a fresh ephemeral ECC private key ki.
3. The user computes the ephemeral ECC public key Pi = ki · G.
4. The user recovers X = F ⊕ E = H(MP ‖ XGwnU).
5. The user computes the authenticator Ni = H(X ‖ XGwnU ‖ T1).
6. The user sends {MI, E, Pi, Ni, T1} to the server B, where Pi is transmitted in compressed form.
7. The witness event user_sensor_t1(T1) is recorded.

Phase B — Authentication Request (Server B → Sensor C).

8. The server B verifies the freshness of timestamp T1 and recomputes and checks the authenticator Ni. If either check fails, the session is aborted; otherwise the user is authenticated.
9. The server forwards {SenID, Pi, T1} to the sensor C.
10. The witness event server_sensor_r is recorded.
1. Phase C — Sensor Authentication and ECC Response (Sensor C → Server B).
11. The sensor verifies the freshness of T1 and generates a fresh timestamp T2.
12. The sensor selects a fresh ephemeral ECC private key kj.
13. The sensor computes the ephemeral ECC public key Pj = kj · G.
14. The sensor computes the authenticator Aj = H(XGwnS ‖ T1 ‖ T2).
15. The sensor sends {Pj, Aj, T1, T2, SenID} to the server B.
16. The witness event sensor_server_t2(T2) is recorded.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

Phase D — Server Authentication Forwarding (Server B → User A).

17. The server verifies the authenticator Aj. If verification fails, the session is aborted; otherwise the sensor is authenticated.
18. The server generates a fresh timestamp T3.
19. The server sends {Pj, T1, T2, T3} to the user A.
20. The witness event server_user_t3(T3) is recorded.

Phase E — ECC Session-Key Establishment (User A ↔ Sensor C).

21. The user verifies the freshness of timestamps T1, T2, and T3.
22. The user computes the ECC shared secret point Kij = ki · Pj.
23. The user derives the session key SK = H(Kij).
24. The sensor computes the ECC shared secret point Kji = kj · Pi.
25. The sensor derives the session key SK = H(Kji).
26. Because Kij = ki · Pj = ki · (kj · G) = kj · (ki · G) = kj · Pi = Kji, both parties obtain an identical session key, confirming mutual authentication.
27. The witness event user_sensor_sk is recorded.

At the end of Phase E, the user and the sensor share the authenticated session key SK = H(Kij) = H(Kji), where Kij = Kji = (ki · kj) · G. The session key is derived from fresh ephemeral keys contributed by both parties; under the ECCDH assumption, an adversary cannot compute (ki · kj) · G from the observed public values Pi and Pj. Consequently, even if the server's long-term private key xs or the master secrets are later compromised, previously established session keys remain secret, ensuring perfect forward secrecy (Li & Hu, 2024; Park et al., 2025). The combination of timestamp-based freshness checks and hash-based authenticators at each hop further provides resistance to replay, impersonation, and man-in-the-middle attacks; these properties are formally verified in Section 5. Figure 3 illustrates the complete message flow of this phase.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

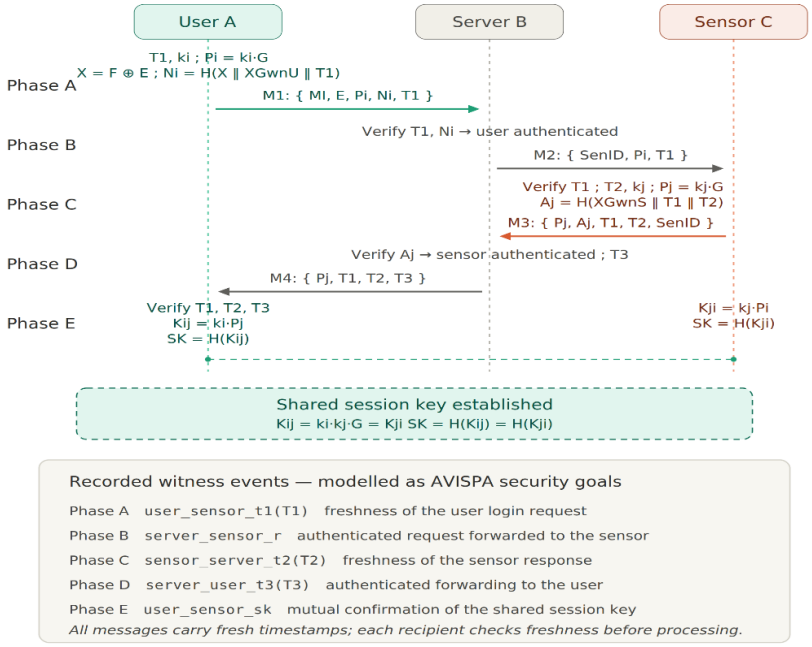

Figure 3. Message Flow of The Login, Authentication, And Key Agreement Phase of the Proposed Protocol.

Security Analysis

This section evaluates the security of the proposed protocol. Section 5.1 presents an informal analysis demonstrating resistance to well-known attacks, and in Section 5.2 the protocols security is further validates using the AVISPA automated verification framework, in which both the OFMC and CL-AtSE back-ends classify the protocol as SAFE under the Dolev–Yao adversary model. Furthermore, future work may employ formal proofs under Real-Or-Random (ROR) or the Canetti-krawcyk model to provide strong security guarantees.

Informal Security Analysis

In the following, we analyze the proposed protocol against a Dolev–Yao adversary A (Section 3.2) who fully controls the public channel and may also extract data stored on a captured device. For each attack, we argue why A cannot succeed.

Replay Attacks

Every message transmitted during the login, authentication, and key agreement phase carries a fresh timestamp (T1, T2, or T3). On receiving a message, each entity checks the embedded timestamp against the current time and a maximum permissible transmission delay ΔT, and rejects the message if the freshness test fails. Suppose A intercepts a valid message and retransmits it later; the stale timestamp fails the freshness check and the message is discarded. Moreover, the authenticators Ni and Aj are bound to their timestamps, so a replayed message cannot pass verification later. In addition, the ephemeral ECC keys ki and kj are freshly generated in every session, ensuring that each session produces a unique session key. The protocol therefore resists replay attacks.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

MITM(Man-In-The-Middle) Attack

To mount a man-in-the-middle (MITM) attack, A must intercept and alter messages without detection. However, the authenticity of each message is protected by keyed hash authenticators that depend on long-term secrets unknown to A. The user authenticator Ni depends on the master secret XGwnU, and the sensor authenticator Aj depends on the master secret XGwnS; both values are held only by the legitimate entities and the server. If A modifies any field of a transmitted message, the recomputed authenticator at the receiver will not match, and the session is aborted. Furthermore, even if A substitutes its own ephemeral public key, it cannot compute the ECC shared secret Kij = (ki · kj) · G without knowledge of ki or kj, owing to the hardness of the ECCDH problem. Hence the protocol withstands MITM attacks.

Impersonation Attacks

To forge a valid login request, A must produce Ni = H(X ‖ XGwnU ‖ T1), which requires the master secret XGwnU and the masked password MP, neither of which is available to A. To impersonate the sensor, A must produce Aj = H(XGwnS ‖ T1 ‖ T2), which requires the master secret XGwnS. The server is authenticated implicitly: only the legitimate server, holding XGwn, XGwnU, and XGwnS, can verify Ni and Aj and correctly relay the protocol messages. Because all authenticators are bound to secrets that A cannot obtain, the protocol resists user, sensor, and server impersonation attacks.

Stolen Device (Smart-Card / Sensor Capture) Attacks

Suppose A captures a user's smart card and extracts its contents SC = {R ‖ MI ‖ E ‖ F ‖ XGwnU} through side-channel analysis. The stored values do not directly reveal the user's identity ID or password PWD, since MI = H(R ‖ ID) and the password is protected within MP = H(R ‖ PWD) through the one-way hash function. To impersonate the user, A would still need MP to recover X and thus the authenticator Ni; the password is never stored on the card. Similarly, if a sensor is captured, its stored values RecE and RecF are outputs of one-way hash functions keyed by server secrets and do not expose XGwn, XGwnS, or the pre-shared key S. Consequently, device capture alone does not enable the adversary to compromise the protocol.

Session-Key Disclosure Attacks

The session key is SK = H(Kij) = H(Kji), where Kij = Kji = (ki · kj) · G. To recover SK, the adversary must obtain the shared secret point (ki · kj) · G from the publicly transmitted ephemeral keys Pi and Pj. This is precisely the Elliptic Curve Computational Diffie–Hellman problem, which is computationally infeasible. Because ki and kj are freshly chosen in every session, each session key is independent of all others, so the disclosure of one session key does not endanger any other. The protocol therefore provides session-key secrecy and known-key security.

Perfect Forward Secrecy

The session key is derived solely from the ephemeral ECC keys ki and kj, and not from any long-term secret. Suppose the server's long-term private key xs and the master secrets are later compromised. An adversary holding these values, together with all recorded transcripts, still cannot reconstruct a past session key, because doing so would require solving the ECCDH

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

problem for the ephemeral values of that session. The ephemeral private keys are discarded after each session. Hence the protocol achieves perfect forward secrecy.

Offline Password-Guessing Attacks

The password PWD appears in the protocol only inside MP = H(R ‖ PWD), which is combined with the master secret XGwnU to form X = H(MP ‖ XGwnU). The transmitted authenticator Ni thus depends on both the password and the server master secret XGwnU. Even if A captures the smart card and obtains XGwnU, it does not know the nonce R needed to compute MP, and the password is not stored on the card. The adversary therefore cannot construct a verification equation containing only the password as the unknown and cannot confirm a guess offline. The protocol consequently resists offline password-guessing attacks.

Insider Attacks

During user registration, the user transmits only {MI, MP}, where MI = H(R ‖ ID) and MP = H(R ‖ PWD), rather than the plaintext identity or password. A privileged insider at the server therefore never observes the user's real credentials and cannot reuse them on other services. Likewise, during sensor registration the sensitive material is masked by the pre-shared key S and the nonce R. Because the registration messages reveal no directly reusable secret, the protocol resists privileged-insider attacks.

Summary

Table 3 summarizes the security features of the proposed protocol in comparison with representative existing schemes. The proposed protocol satisfies all listed security requirements, whereas several prior schemes fail to provide one or more of them.

Table 3. Security-Feature Comparison.

Security feature	(Li and Hu (2024)	(Thakur et al., 2024)	(Hammi et al., 2020)	(Shunfang et al., 2024)	Proposed
Replay attack resistance	✓	✓	✓	✓	✓
MITM(man-in-the-middle) attack resistance	✓	✓	✓	✓	✓
Impersonation attack resistance	✓	✓	✗	✓	✓
Stolen-device attack resistance	✓	✓	✓	✗	✓
Session-key disclosure resistance	✓	✓	✓	✓	✓
Perfect forward secrecy	✗	✓	✗	✗	✓

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

Security feature	(Li and Hu (2024)	(Thakur et al., 2024)	(Hammi et al., 2020)	(Shunfang et al., 2024)	Proposed
Offline password-guessing resistance	✓	✗	✗	✓	✓
Insider attack resistance	✓	✓	✗	✓	✓
Mutual authentication	✓	✓	✓	✓	✓
Formal verification (AVISPA)	✗	✓	✗	✗	✓

Formal Security Verification Using AVISPA

To complement the informal analysis, the proposed protocol was formally verified using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool (Armando et al., 2005), a widely adopted, push-button model checker for security protocols. Protocols are specified in the High-Level Protocol Specification Language (HLPSL); the specification is translated into the Intermediate Format and analyzed by several independent back-end engines under the Dolev–Yao adversary model.

HLPSL Specification

The proposed protocol was modelled in HLPSL using four roles: user (User A), server (Server B), sensor (Sensor C), together with the mandatory session, environment, and goal roles. Each entity role describes the messages it sends and receives, the fresh values it generates, and the state transitions it undergoes. The intruder is granted full Dolev–Yao capabilities. The complete HLPSL specification is provided in the Appendix. The security goals were declared using secrecy_of and authentication_on statements derived from the witness events introduced in Section 4.5: the session key SK and the masked password MP are declared secret, and the witness/request pairs user_sensor_t1, server_sensor_r, sensor_server_t2, server_user_t3, and user_sensor_sk model freshness and entity-authentication properties.

OFMC Back-End Results

The specification was first analyzed with the On-the-Fly Model-Checker (OFMC), which performs a demand-driven search of the protocol state space. OFMC reported that the proposed protocol is SAFE: no attack was found within the analyzed bound, indicating that the stated secrecy and authentication goals hold against the Dolev–Yao intruder. The OFMC output is shown in Figure 4(a).

CL-AtSe Back-End Results

The specification was then analyzed with the Constraint-Logic-based Attack Searcher (CL-AtSe), which translates the protocol into a set of constraints and systematically explores reachable states. CL-AtSe likewise reported the protocol as SAFE, confirming that no attack violating the declared goals exists within the analyzed state space. The CL-AtSe output is shown in Figure 4(b).

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

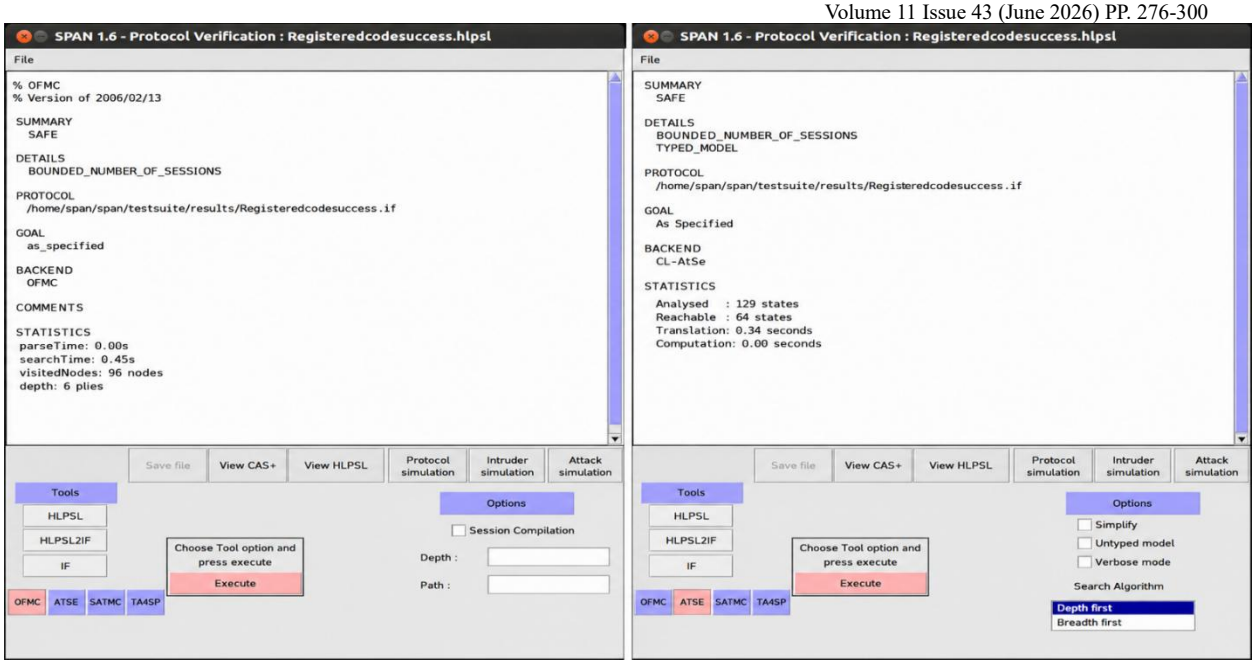

Figure 4. AVISPA Formal Verification Results: (A) OFMC Back-End And (B) CL-Atse Back-End, Both Reporting SAFE

Verification Outcome

Both the OFMC and CL-AtSe back-ends independently reported the SAFE verdict for the proposed protocol under the bounded-session model with a Dolev–Yao intruder. The SAFE result indicates that the session key SK and the protected secrets remain confidential, and that all authentication goals are satisfied, so replay, impersonation, and man-in-the-middle attacks are not feasible. These formal results are consistent with the informal analysis of Section 5.1 and confirm that the proposed protocol meets the security requirements defined in Section 3.3. Table 4 summarizes the verification outcome.

Table 4. AVISPA Verification Summary.

Back-end	Analysis method	Goals checked	Result
OFMC	On-the-fly model checking, bounded sessions	Secrecy of SK and secrets; authentication	SAFE
CL-AtSe	Constraint-based attack search, typed model	Secrecy of SK and secrets; authentication	SAFE

Performance Evaluation

This section evaluates the performance of the proposed protocol in terms of computational cost, communication overhead, and storage requirements, and compares it with recent ECC-based authentication schemes. The analysis focuses on the login, authentication, and key agreement phase, since registration is a one-time operation performed over a secure channel.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

To make the comparison precise and reproducible, all transmitted fields are sized according to a single, explicitly stated parameter set. The protocol is instantiated over a 160-bit elliptic curve, consistent with the convention used in the comparative literature (Li & Hu, 2024; "Provably Secure ECC-Based Anonymous Authentication," 2024; Thakur et al., 2024). Each elliptic-curve public key is transmitted in compressed form: rather than sending both coordinates of a curve point, only the x-coordinate together with a single sign bit is transmitted, and the receiver reconstructs the y-coordinate by solving the curve equation. Point compression is a standard, library-supported encoding technique; it does not alter the protocol logic, the messages exchanged, the security analysis of Section 5, or the AVISPA model, and it approximately halves the number of bits required to transmit each public key.

Table 5. Bit-Length Assumptions For Transmitted And Stored Fields.

Field	Example elements	Size (bits)
Identity / pseudo-identity	MI, SenID	160
Hash output	E, Ni, Aj	160
Nonce	R	160
Compressed ECC point	Pi, Pj	161
Timestamp	T1, T2, T3	32

Computational Cost Analysis

The computational cost of an authentication protocol is determined by the cryptographic operations performed by each entity. For resource-constrained IoT devices the dominant cost is elliptic-curve scalar (point) multiplication, whereas one-way hash functions are comparatively inexpensive and bitwise XOR operations are negligible. We denote by T_{ecm} the cost of an ECC point multiplication, by T_h the cost of a one-way hash, by T_{xor} the cost of a bitwise XOR, and by T_{dec} the cost of point decompression. Indicative execution times measured on IoT-class hardware in the literature are approximately $T_{ecm} \approx 2.2$ ms and $T_h \approx 0.005$ ms, with T_{xor} omitted as negligible. Point decompression requires a single modular square-root computation; its cost is an order of magnitude smaller than T_{ecm} and is treated as a minor term that does not affect the ranking of schemes.

Counting the operations in the login–authentication–key-agreement phase of Section 4.5: User A computes one point multiplication for Pi and one for the shared secret Kij (2 T_{ecm}), the hashes for X, Ni, and SK (3 T_h), and one XOR (1 T_{xor}). Sensor C computes one point multiplication for Pj and one for Kji (2 T_{ecm}), and the hashes for Aj and SK (2 T_h). Server B performs verification and forwarding only, computing the hashes needed to verify Ni and Aj (2 T_h), and performs no point multiplication. The total computational cost of the proposed protocol is therefore $4\ T_{ecm} + 7\ T_h + 1\ T_{xor}$. Each resource-constrained end entity performs only 2 T_{ecm}, and the resource-rich server performs none. Using the indicative timings above, the end-to-end cryptographic cost is approximately $4 \times 2.2 + 7 \times 0.005 \approx 8.84$ ms.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

Communication Overhead

The login, authentication, and key agreement phase consists of four messages — M1 (User → Server), M2 (Server → Sensor), M3 (Sensor → Server), and M4 (Server → User), as illustrated in Figure 3. Four messages is the minimum for a mediated three-party protocol in which the server relays authentication between a user and a sensor that share no prior secret. Applying the field sizes of Table 5, with elliptic-curve public keys transmitted in compressed form (161 bits), the four messages incur the following overhead: M1 = {MI, E, Pi, Ni, T1} = 673 bits; M2 = {SenID, Pi, T1} = 353 bits; M3 = {Pj, Aj, T1, T2, SenID} = 545 bits; and M4 = {Pj, T1, T2, T3} = 257 bits. The total communication overhead of the proposed protocol is therefore 1828 bits across the full authentication phase. Without point compression i.e., transmitting each public key as a full 320-bit uncompressed point the same protocol would incur 2464 bits; point compression thus reduces communication overhead by about 26% at the cost of only a minor decompression operation per received key, with no change to the protocol or its security. The user transmits only M1 (673 bits) and the sensor only M3 (545 bits), so each constrained end device sends well under 700 bits comfortably within the payload capacity of typical low-power IoT radio links.

Storage Cost

The storage cost reflects the long-term memory an entity must reserve for credentials. User A stores the smart-card data SC = {R ‖ MI ‖ E ‖ F ‖ XGwnU}; with each value being 160 bits, the user storage requirement is 800 bits. Sensor C stores RecE and RecF, i.e., 320 bits. Server B stores the long-term private key xs, the three master secrets, and the pre-shared key S; as a resource-rich entity, its storage cost is not a constraint. The sensor the most constrained entity must persistently store only 320 bits of secret material, since the ephemeral ECC keys are generated freshly per session and discarded afterward. This minimal footprint makes the protocol well suited to memory-limited IoT nodes

Comparative Analysis

Table 6 compares the proposed protocol with recent ECC-based authentication schemes in terms of computational cost, communication overhead, and security features. Table 7 then compares them across the security and functional metrics most relevant to IoT deployments.

Table 6. Computational And Communication Cost Comparison.

Scheme	Computation cost	Communication cost	Security features
Hammi et al. (2020)	~6 T_ecm + 4 T_h	~1952 bits	Mutual auth.; no PFS
Thakur et al. (2024)	~5 T_ecm + 9 T_h	~2048 bits	Mutual auth.; PFS; AVISPA-verified
(Shunfang et al., 2024)	~6 T_ecm + 8 T_h	~2304 bits	Mutual auth.; anonymity; PFS
Li and Hu (2024)	~5 T_ecm + 10 T_h	~1984 bits	Mutual auth.; anonymity; PFS; ESL resistance

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

Scheme	Computation cost	Communication cost	Security features
Braeken (2022)	6 T_ecm + 6 T_h	4224 bits †	ECC-based AKA (dew-assisted IoT)
Pirmoradian et al. (2023)	13 T_ecm + 6 T_h + 2 T_enc	4512 bits †	ECC-based AKA (WBAN)
Servati and Safkhani (2023)	15 T_ecm + 18 T_h	3456 bits †	ECC-based authentication (healthcare IoT)
Keshta (2024)	8 T_ecm + 22 T_h	2558 bits †	ECC-based authentication (resource-constrained IoT)
Alzahrani (2025)	12 T_ecm + 22 T_h + 17 T_ea	3548 bits †	ECC-based authentication (edge-IoT)
Proposed	4 T_ecm + 7 T_h + 1 T_xor	1828 bits	Mutual auth.; anonymity; PFS; ESL resistance; AVISPA-verified

† Communication-cost figures for Braeken (2022), Pirmoradian et al. (2023), Servati and Safkhani (2023), Keshta (2024), and Alzahrani (2025) are reported as published in their respective source papers, where elliptic-curve points are transmitted in standard uncompressed form. The proposed protocol applies point compression (161-bit points) and, on an equivalent uncompressed-point basis, would require 2464 bits (see Section 6.2). In the computation-cost column, T_ecm denotes an elliptic-curve scalar multiplication, T_h a one-way hash, T_xor a bitwise XOR, T_enc a symmetric encryption/decryption, and T_ea an elliptic-curve point addition; for resource-constrained devices the scalar multiplication T_ecm is the dominant term, while the remaining operations are comparatively negligible.

Table 7. Comparative Analysis with Recent Schemes.

Metric	(Hammi et al., 2020)	(Thakur et al., 2024)	(Shunfang et al., 2024)	(Li & Hu, 2024)	Proposed
Mutual authentication	Yes	Yes	Yes	Yes	Yes
User anonymity	No	No	Yes	Yes	Yes
Perfect forward secrecy	No	Yes	Yes	No	Yes
ESL attack resistance	No	Yes	No	Yes	Yes

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

Metric	(Hammi et al., 2020)	(Thakur et al., 2024)	(Shunfang et al., 2024)	(Li & Hu, 2024)	Proposed
Pairing-free (lightweight)	Yes	Yes	Yes	Yes	Yes
End-device ECM operations	3	2–3	3	2–3	2
Formal verification (AVISPA)	No	No	No	No	Yes

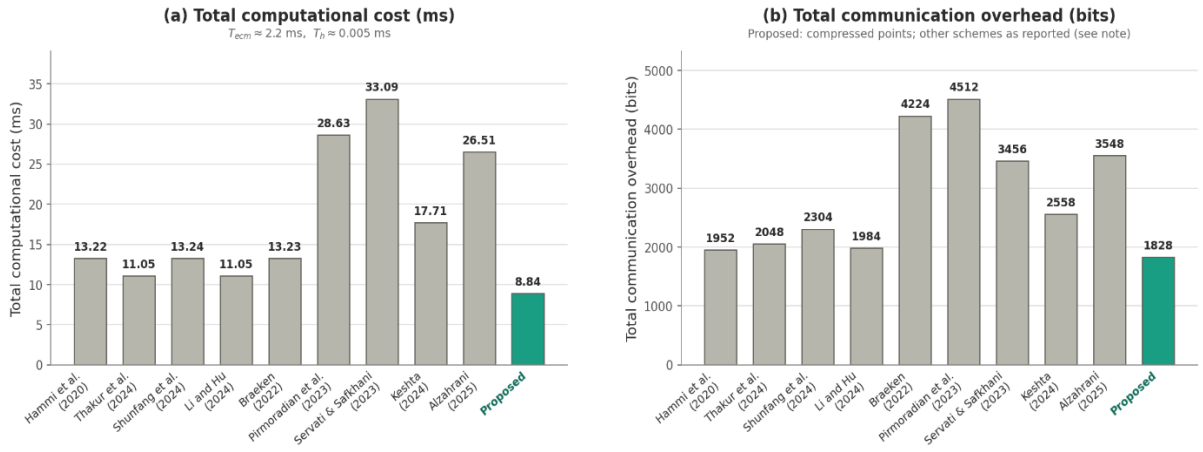

Figure 5. Performance Comparison: (A) Total Computational Cost (Ms) And (B) Total Communication Overhead (Bits).

As Tables 6 and 7 show, the proposed protocol maintains a low cost suitable for constrained IoT devices (Figure 5(a)). The additional ECC-based schemes of Braeken (2022), Pirmoradian et al. (2023), Servati and Safkhani (2023), Keshta (2024), and Alzahrani (2025) each require between six and fifteen scalar multiplications, so the proposed protocol reduces this dominant operation by roughly 33% to 73% relative to those schemes. Its communication overhead of 1828 bits is likewise the lowest among all ten compared schemes (Figure 5(b)); as noted beneath Table 6, this figure reflects point compression, and even the uncompressed-point equivalent of 2464 bits remains below every newly added scheme. Crucially, the proposed protocol attains this efficiency while satisfying the complete set of security and functional requirements, whereas every compared scheme omits at least one of these properties. The evaluation therefore demonstrates that the proposed protocol achieves a favorable balance between security strength and efficiency, making it well suited to resource-constrained IoT environments.

Conclusion and Future Work

This paper addressed secure authentication and key agreement for resource-constrained IoT environments, where traditional RSA-, pairing-, and certificate-based infrastructures impose excessive overhead. To overcome these limitations, a lightweight ECC-based authentication

and key agreement protocol was proposed for a three-tier architecture consisting of a user, a semi-trusted gateway server, and an IoT sensor. The proposed protocol performs registration through a secure channel and later establishes mutual authentication and a shared session key over a public channel. It relies only on elliptic-curve scalar multiplication, hash functions, and XOR operations, avoiding bilinear pairings and certificate management. The session key is generated using fresh ephemeral ECC keys, ensuring contributory key agreement and independence from long-term secrets. Security analysis showed resistance against replay, man-in-the-middle, impersonation, stolen-device, session-key disclosure, offline password-guessing, and insider attacks, while providing mutual authentication, anonymity, and perfect forward secrecy. Performance evaluation demonstrated low computational, communication, and storage overhead, making the protocol suitable for constrained IoT devices. Formal verification using the AVISPA tool under the Dolev–Yao model produced SAFE results in both OFMC and CL-AtSe backends, confirming the protocol's secrecy and authentication properties and validating the informal security analysis.

Several directions remain for future work. Blockchain integration could decentralize registration and key management while enabling tamper-resistant authentication records and transparent device revocation. The protocol may also be extended to secure federated learning environments by protecting model-update exchanges against poisoning and inference attacks. In addition, since ECC-based security is vulnerable to quantum computing threats, developing a lightweight post-quantum or hybrid authentication scheme for IoT devices is an important future research direction. These extensions could further improve the security, scalability, and long-term sustainability of IoT authentication systems.

Acknowledgements:	The authors would like to express their sincere gratitude to The Universiti Teknologi Malaysia for providing the necessary resources and support throughout the course of this research. Special appreciation is extended to colleagues and peers who contributed valuable insights and constructive feedback, which greatly enhanced the quality of this paper.
Funding Statement:	This research received No Funding.
Conflict of Interest Statement:	The authors declare that there is no conflict of interest regarding the publication of this paper. All authors have contributed to this work and approved the final version of the manuscript for submission to the Journal of Information System and Technology Management (JISTM).
Ethics Statement:	This study did not involve any human participants, animals, or sensitive data requiring ethical approval. The authors confirm that the research was conducted in accordance with accepted academic integrity and ethical publishing standards.
Author Contribution Statement:	All authors contributed significantly to the development of this manuscript. All authors read and approved the final version of the manuscript prior to submission.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 43 (June 2026) PP. 276-300

References

Abdussami, M., Dwivedi, S. K., Al-Shehari, T., Saravanan, P., & Alghamdi, S. A. (2024). DEAC-IoT: Design of lightweight authenticated key agreement protocol for intra and inter-IoT device communication using ECC with FPGA implementation. *Computers & Electrical Engineering, 120*, 109696. https://doi.org/10.1016/j.compeleceng.2024.109696

Alzahrani, N. (2025). Security importance of edge-IoT ecosystem: An ECC-based authentication scheme. *PLOS ONE, 20*(6), e0322131. https://doi.org/10.1371/journal.pone.0322131

Armando, A., Basin, D., Compagna, L., Cuéllar, J., Hankes Drielsma, P., Heinemann, B., & Sasse, R. (2005). The AVISPA tool for the automated validation of Internet security protocols and applications. In *Computer Aided Verification (CAV 2005)* (Lecture Notes in Computer Science, Vol. 3576, pp. 281–285). Springer.

Baccouri, S., Farhat, H., Azzabi, T., & Attia, R. (2024). Lightweight authentication scheme based on elliptic curve ElGamal. *Journal of Information and Telecommunication, 8*(2), 231–261.

Braeken, A. (2022). Authenticated key agreement protocols for dew-assisted IoT systems. *The Journal of Supercomputing, 78*(11), 12093–12113. https://doi.org/10.1007/s11227-022-04364-z

Dolev, D., & Yao, A. C. (1983). On the security of public key protocols. *IEEE Transactions on Information Theory, 29*(2), 198–208. https://doi.org/10.1109/TIT.1983.1056650

Gautam, D., Rana, A., Obaidat, M. S., Kumar, P., & Prajapat, S. (2024). A provably secure biometric-based authentication and key agreement scheme for Internet of Drones. *Transactions on Emerging Telecommunications Technologies, 35*, e4893. https://doi.org/10.1002/ett.4893

Hammi, B., Fayad, A., Khatoun, R., Zeadally, S., & Begriche, Y. (2020). A lightweight ECC-based authentication scheme for Internet of Things (IoT). *IEEE Systems Journal, 14*(3), 3440–3450. https://doi.org/10.1109/JSYST.2019.2938540

Hankerson, D., Menezes, A. J., & Vanstone, S. A. (2004). *Guide to elliptic curve cryptography*. Springer.

Hu, S., Zhang, Y., Guo, Y., Zhong, W., Chen, Y., & Chen, L. (2025). Efficient IoT user authentication protocol with semi-trusted servers. *Sensors, 25*(7), 2013. https://doi.org/10.3390/s25072013

IoT Analytics. (2025). *State of IoT 2025: Number of connected IoT devices growing 14% to 21.1 billion globally*. https://iot-analytics.com

Keshta, I. (2024). A CRC-based authentication model and ECC-based authentication protocol for resource-constrained IoT applications. *IEEE Access, 12*, 156765–156784. https://doi.org/10.1109/ACCESS.2024.3482991

Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation, 48*(177), 203–209. https://doi.org/10.1090/S0025-5718-1987-0866109-5

Li, M., & Hu, S. (2024). A lightweight ECC-based authentication and key agreement protocol for IoT with dynamic authentication credentials. *Sensors, 24*(24), 7967. https://doi.org/10.3390/s24247967

Lightweight authentication and key agreement protocol for IoT based on ECC. (2023). *IEEE Conference Paper*.

Miller, V. S. (1986). Use of elliptic curves in cryptography. In A. M. Odlyzko (Ed.), *Advances in cryptology — CRYPTO '85* (Lecture Notes in Computer Science, Vol. 218, pp. 417–426). Springer.

Park, K., Kim, M., & Park, Y. (2025). Security evaluation of provably secure ECC-based anonymous authentication and key agreement scheme for IoT. *Sensors, 25*(1), 237. https://doi.org/10.3390/s25010237

Pirmoradian, F., Safkhani, M., & Dakhilalian, S. M. (2023). ECCPWS: An ECC-based protocol for WBAN systems. *Computer Networks, 224*, 109598. https://doi.org/10.1016/j.comnet.2023.109598

Servati, M. R., & Safkhani, M. (2023). ECCbAS: An ECC-based authentication scheme for healthcare IoT systems. *Pervasive and Mobile Computing, 90*, 101753. https://doi.org/10.1016/j.pmcj.2023.101753

Shah, N. H., Ismail, S. A., Azizan, A., Anoop, A., Khan, D. T., & Ahamed, S. B. (2025). Lightweight authentication protocols in Internet of Things – A review. *International Journal of Engineering Trends and Technology, 73*(3), 104–119. https://doi.org/10.14445/22315381/IJETT-V73I3P108

Shah, N. H., Khan, D. T., Banu, A. A., & Shah, L. H. (2023). Symmetric and asymmetric encryption schemes for Internet of Things: A survey. *International Journal of Intelligent Systems and Applications in Engineering, 11*, 254–260.

Shunfang, H., Jiang, S., Miao, Q., Yang, F., Zhou, W., & Duan, P. (2024). Provably secure ECC-based anonymous authentication and key agreement for IoT. *Applied Sciences, 14*(8), 3187. https://doi.org/10.3390/app14083187

Tedeschi, P., Sciancalepore, S., Eliyan, A., & Di Pietro, R. (2020). LiKe: Lightweight certificateless key agreement for secure IoT communications. *IEEE Internet of Things Journal, 7*(1), 621–638. https://doi.org/10.1109/JIOT.2019.2942823

Thakur, G., Kumar, P., Chen, C.-M., et al. (2023). A robust privacy-preserving ECC-based three-factor authentication scheme for metaverse environment. *Computer Communications, 211*, 271–285. https://doi.org/10.1016/j.comcom.2023.05.012

Thakur, G., Kumar, P., et al. (2024). A provably secure authenticated key agreement protocol for industrial sensor network system. *Concurrency and Computation: Practice and Experience, 36*, e8250. https://doi.org/10.1002/cpe.8250

Transforma Insights & Exploding Topics. (2025). *Number of Internet of Things (IoT) connections worldwide from 2022 to 2034*. Statista. https://www.statista.com

Vangala, A., Das, A. K., Mitra, A., Das, S. K., & Park, Y. (2023). Blockchain-enabled authenticated key agreement scheme for mobile vehicles-assisted precision agricultural IoT networks. *IEEE Transactions on Information Forensics and Security, 18*, 904–919. https://doi.org/10.1109/TIFS.2022.3201234

Wang, D., & Wang, P. (2018). Two birds with one stone: Two-factor authentication with security beyond conventional bound. *IEEE Transactions on Dependable and Secure Computing, 15*(4), 708–722. https://doi.org/10.1109/TDSC.2016.26023