Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Cert Bind-DKEP: A Certificate- Hash-Bound Dual-Curve Ephemeral Key Establishment Protocol for Lightweight MQTT-IoT Security

DVPS Pranavi[a], Dr. Saliha Bathool[b]

[1]*School of Computer Science and Engineering, RV University, Bengaluru, India*

[2]*School of Computer Science and Engineering, RV University, Bengaluru, India*

ARTICLEINFO

Received: 10 May 2026

Revised: 03 July 2026

Accepted: 12 July 2026

ABSTRACT

Internet of Things (IoT) deployments have rapidly grown over time, necessitating the development of lightweight and efficient authentication methods suited for such systems. Although MQTT has become the de facto communication standard within the scope of IoT ecosystems, many issues require solutions regarding the establishment of secure session keys and lightweight authentication. In this paper, the proposed protocol, 3L-EKEP, which provides both lightweight authentication and session-key establishment capabilities for secure MQTT-based communications within IoT devices, will be presented. This protocol includes certificate-hash bindings, pre-shared-key challenge validation, dual Elliptic Curve Diffie-Hellman key exchanges with P-256 and X25519, HKDF-SHA256-derived session keys, and AES-GCM data transfer. The protocol was implemented via the Python programming language and evaluated experimentally. The security of the protocol was examined based on factors including authentication binding, session-key uniqueness, session-key independence, avalanche effect, and ciphertext tampering detection. Based on the experimental results, it was determined that the protocol generated 1000 unique session keys in 1000 executions without any collisions. On the other hand, the average percentage of bits different from the previous session key in the independence tests was around 50%, and the avalanche effect tests revealed that the average number of bits changed was 48.83%. With respect to the performance of the protocol, it was determined that during 1000 protocol executions, the average latency was found to be 2.224 ms, with 2.532 ms and 2.862 ms being the P95 and P99 latencies.

Keywords: Elliptic Curve Cryptography (ECC), Internet of Things (IoT), MQTT Security, Lightweight Authentication, Session-Key Establishment, Dual-Curve ECDH, X25519, P-256, HKDF-SHA256, AES-GCM, Key Uniqueness, Avalanche Effect, Cryptographic Performance Evaluation.

INTRODUCTION

The advent of IoT technology is a critical milestone along the way to creating digital ecosystems, thanks to the interaction of sensors, devices, gateways, and cloud services. The development of IoT technology has reached quite an advanced stage, and the technology has already been applied in several fields such as smart homes, healthcare monitoring, industrial control systems, intelligent transportation systems, environmental monitoring, and smart energy. As per the latest industry reports, the number of interconnected IoT devices will grow to billions within the next several years for the creation of a large data and communication infrastructure.

Therefore, the Message Queuing Telemetry Transport (MQTT) protocol has become a popular communication protocol in IoT owing to its lightweight messaging between resource-constrained devices. The MQTT protocol involves the publish/subscribe model, and, consequently, minimal bandwidth and computation capabilities are used

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

for message exchange. Nevertheless, despite the efforts by the MQTT protocol towards efficient messaging, it does not offer any means for authentication, exchange of session keys, and cryptographic identification of communicating parties.

The issue of security remains one of the most significant issues that needs to be solved in any IoT environment. The devices that lack computational capability, memory, and even energy sources make the use of heavy cryptography unattainable. Although the security provided by TLS will ensure the safety of the user's data, the implementation of such security solutions would require more communication and computation resources. In addition, the implementation of a single method of authentication or key agreement may be insufficient in a heterogeneous IoT environment.

In recent years, several research works were conducted on IoT authentication and key agreement protocols that utilise ECC, lightweight authentication, enhanced security features of MQTT, intrusion detection systems, security formalisation verification, and others. Even though the above security solutions cover different aspects related to the IoT environment, most of them concentrate only on either authentication or attacks or formal verification separately, without offering experiments involving all three aspects together.

To solve some of these issues, this paper presents the proposal of application of the Three Layer Ephemeral Key Establishment Protocol (3L-EKEP), which will be used to authenticate and establish session keys in lightweight secured communication using MQTT in the IoT devices. It consists of certificate-hash binding, pre-shared key (PSK) challenge validation, double elliptic curve Diffie–Hellman (ECDH) key exchange using P-256 and X25519, session key establishment using HKDF-SHA256, and secure communication using AES-GCM. A prototype of 3L-EKEP has been developed, and experimentation has been done. Experiments will evaluate authentication binding, uniqueness of session keys, independence of session keys, avalanche effect, communication overhead, computational cost, and latency in running the protocol. In contrast to pure theoretical evaluation of the above-mentioned aspects, the emphasis will be on the practical cryptography usage during protocol execution.

The contributions of this work can be stated as follows:

• Designing and implementing a lightweight three-layer authentication and session key establishment scheme for MQTT-based IoT systems.

• Integrating certificate hash binding, PSK challenge response check, ECC-DH key exchange based on two curves, HKDF-SHA256 key derivation function, and AES-GCM encryption technique into one framework.

• Performing security analysis of protocol properties such as authentication binding, session key uniqueness, session key independence, avalanche effect, and tampering detection.

• Analysis of protocol performance through latency, communication overhead, and cryptography benchmarks.

• Protocol vulnerabilities and possible future research directions to overcome them, including protection against replay attacks, addition of digital signatures, and protocol validation.

The structure of this paper is as follows. The problem statement and motivation behind the development of the proposed framework are explained in Section 2. Literature review and research gaps are addressed in Section 3. The methodology of 3L-EKEP and its description are presented in Section 4. Experimental results and their analysis are given in Section 5. Conclusions and future research direction are in Section 6.

LITERATURE REVIEW

With the increase in popularity of IoT solutions, a lot of studies are being done on lightweight authentication schemes, key agreement protocols, secure communication platforms, and verifications. These recent studies have concentrated on making improvements in the area of security while keeping the computation efficient at the same time.

2.1 Authentication and Key-Agreement Schemes Using ECC

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

ECC is gaining popularity in IoT authentication due to the possibility of achieving a high level of security by using smaller keys. There are many cryptographic schemes based on ECC that were invented to address issues appearing in the use of classical public key infrastructure in such limited-resource environments.

Li and Hu proposed an ECC-based authentication and key-agreement protocol suitable for the IoT environment with limited resources. Their work proved that it reduces the amount of communication overhead and also makes it secure against known attacks. The work of Hu et al. introduced an ECC-based key agreement protocol evaluated under the eCK security model.

Recent works have also investigated various ECC-based authentication schemes with the inclusion of certificate validation, ephemeral key exchange, and lightweight session key establishment. Such schemes are usually computationally more efficient in comparison with traditional RSA-based solutions. However, most ECC-based authentication solutions are based only on a key agreement scheme and do not involve any experimental security evaluation.

2.2 MQTT Security and Lightweight Authentication Mechanisms

With the advent of its publish-subscribe paradigm and lightweight architecture, MQTT has gained widespread usage among all other IoT communication protocols owing to its high efficiency in terms of communication overhead. Though MQTT allows efficient communication between parties, it lacks features that allow for application-level authentication and lightweight session-key establishment.

Various researchers have studied MQTT security improvements through the incorporation of token-based authentication techniques, broker-side authorisation systems, certificate-based access controls, and lightweight authentication mechanisms. For example, a study by Javadi et al. demonstrated a new MQTT security framework for authentication purposes in IoT networks. Moreover, many researchers have discussed using ECC-based authentication methods in the MQTT communication model to achieve lower computational overhead and ensure secure communication.

Though these improvements enhance MQTT security, much work remains focused either only on lightweight authentication or on access control mechanisms, with no further studies carried out to verify the cryptographic strengths, such as unique session-key establishment, avalanche effect, or key independence properties.

2.3 Formal Verification and Security Modeling

It is increasingly important to apply formal verification methodologies to verify any security weaknesses in authentication protocols before deployment in practical systems. Several formal methods, including ProVerif, Tamarin, and AVISPA, have been widely used to prove authentication, confidentiality, and correctness of protocols.

Recently, some formal verification approaches have been utilized to verify lightweight authentication protocols of IoT against the robustness to replay attack, impersonation attack, and key compromise attack. Furthermore, many researchers have applied adversarial security models, including ROM, ROR, and eCK security models, in their security analysis.

While the formal verification approach ensures the theoretical safety, it is hard to see practical implementations in experiments of existing works.

2.4 Intrusion Detection and Attack Analysis in IoT Networks

In addition to the mechanisms employing cryptography to authenticate devices, researchers have investigated the use of intrusion detection systems (IDS) and machine learning algorithms to detect any malicious activities on IoT networks. Current research includes attack detection, attack classification, anomaly detection, replay attack detection, and denial of service attack detection using both supervised and unsupervised machine learning techniques. These approaches are found to be effective in detecting malicious activities following network events. Note that intrusion detection systems are reactive and cannot authenticate cryptographically or set up sessions.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

2.5 Research Gaps and Rationale

From the reviewed literature, there is evidence of remarkable advancements made in areas of ECC-based authentication, security improvements for the MQTT protocol, formal verification, and attack detection techniques. However, the current literature tends to focus on separate aspects of IoT security, such as authentication, attack detection, and theoretical analysis of a system without an attempt to evaluate the combined aspects in practice. A very limited number of studies can be found that experimentally tested a lightweight MQTT-based framework for certificate-hash identity binding, challenge-based authentication, dual-curve key agreement, session-key derivation, and message integrity and confidentiality.

Additionally, there is a lack of reported cryptographic metrics in many studies, including measures such as session key uniqueness, independence of session keys, avalanche effect, communication cost, and protocol delay.

In order to overcome the identified gap in knowledge, a lightweight MQTT-based framework named Three-Layer Ephemeral Key Establishment Protocol (3L-EKEP) is designed and implemented to integrate certificate-hash binding, PSK challenge validation, dual-curve ECDH key agreement based on P-256 and X25519 curves, HKDF-SHA256 session key derivation, and AES-GCM encryption and authentication.

METHODOLOGY

The proposed Three-Layer Ephemeral Key Establishment Protocol (3L-EKEP) will facilitate lightweight authentication and establishment of session keys for MQTT-based IoT systems. This scheme utilises certificate hash binding, challenge-response authentication, dual curve elliptic curve Diffie-Hellman (ECDH) key agreement, key derivation function HKDF, and AES-GCM for encryption of data transfer. The protocol contains three layers of logical authentication, and each layer provides specific security features within the proposed scheme.

3.1 System Architecture

Both IoT device and IoT gateway actors function in an IoT architecture based on MQTT. While implementing the protocol under discussion, both actors pass through a three-level authentication process and create a session key that will be used in their future communication.

In particular, the suggested architecture operates according to the protocol illustrated by Fig. 1. The device creates ephemeral keys and a challenge value; then, they are sent to the gateway in the form of a handshake request. After that, the gateway verifies the received information, establishes a dual-curve key exchange, derives a session key, and sends its public keys back to the device. As a result, the same session key is established, and secure communication begins via the AES-GCM algorithm.

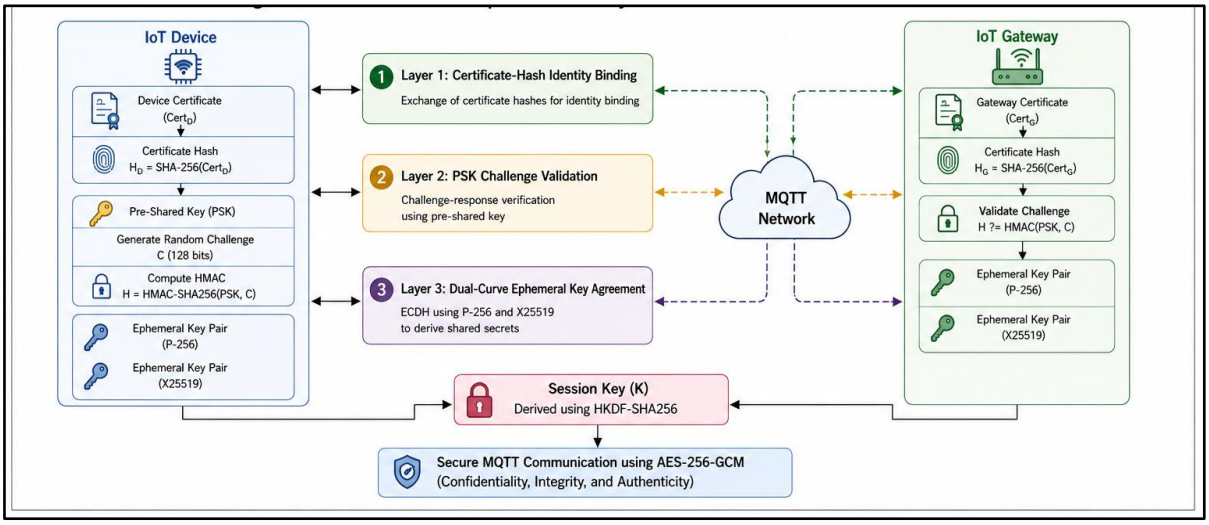

Fig. 1. Architecture of the Proposed Three-Layer 3L-EKEP Authentication Framework

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

3.2 Layer 1: Certificate Hashes Identity Binding

Layer 1 performs identity binding between the communicating nodes. Instead of sending whole certificates each time, the architecture generates hashes of the certificates using the SHA-256 algorithm and adds them to the process of deriving the session key.

Here,

H_D = SHA256 (Cert_Device)

H_G = SHA256 (Cert_Gateway),

where H_D and H_G represent the hash values of the certificate belonging to the device and the gateway, respectively.

Thus, when the certificate hashes are used for key generation, the generated session key becomes cryptographically related to the identities of both communicating parties.

3.3 Layer 2: Challenge Validation Using PSK

At the second layer, challenge-based validation is employed based on the use of the PSK. Upon protocol initiation, the node creates a challenge C using:

C ← Random(128 bits)

The challenge response can be computed by:

H = HMACSHA256(PSK, C)

Here, PSK stands for the pre-shared secret key.

The above HMAC would not only offer more information but will also add entropy in the calculation of the session key. Furthermore, as the challenge itself is produced randomly in each invocation of the protocol, even the slightest change in the challenge produces a different value for H (and hence a different session key).

3.4 Layer 3 - Dual-ECDH Shared Secret Establishment

The third level of authentication involves dual ECDH shared secret establishment. Two different elliptic curve systems will be employed for this process.

The first shared secret will be generated by utilising the NIST P-256 elliptic curve:

S_P256 = ECDH(P256_Device, P256_Gateway)

The second shared secret will be generated by employing Curve25519:

S_X25519 = ECDH(X25519_Device, X25519_Gateway)

Utilising two ECDH exchanges adds cryptographic diversity as the two secrets will be derived from different elliptic curves.

3.5 Session Key Derivation

After the successful completion of all three levels of authentication, the session key is derived using HKDF-SHA256.

The derivation material will be the following:

M = S_P256 || S_X25519 || H_D || H_G || H

where:

S_P256 = P-256 shared secret

S_X25519 = X25519 shared secret

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

H_D = Device certificate hash

H_G = Gateway certificate hash

H = Challenge response HMAC

Thus, the session key K can be calculated as:

K = HKDFSHA256(M)

The derivation results in a 256-bit session key, which is used to secure the MQTT connection since the change in any element of the derivation will result in a different derived session key.

The entire derivation procedure is shown in Fig. 2 below.

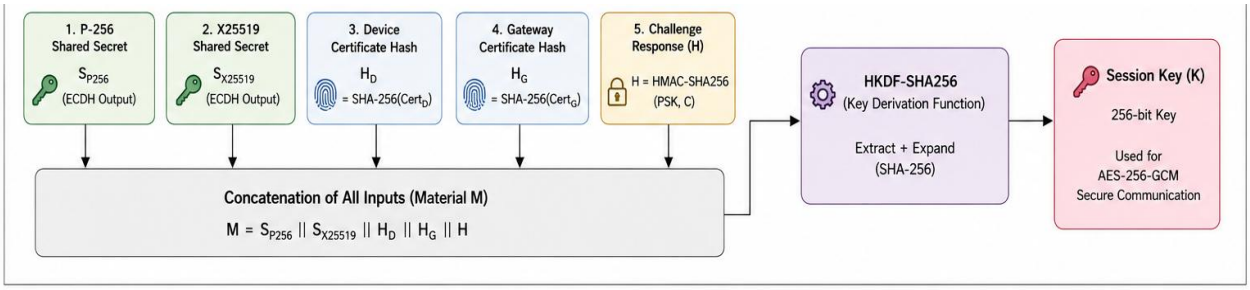

Fig. 2. Session-Key Derivation Process in the Proposed 3L-EKEP Protocol

3.6 Secure MQTT Messages

Once the secure connection is established with the proper generation of a session key, the message content can be made secure by utilising AES-GCM authenticated encryption.

Assuming we have a plaintext message P,

Ciphertext = AESGCM_Encrypt(K, P),

Where K is the session key generated.

Upon receiving the message,

P = AESGCM_Decrypt (K, Ciphertext)

The AES-GCM algorithm guarantees confidentiality, integrity, and detection of tampering with encryption.

3.7 Protocol Execution Sequence

The protocol execution sequence includes the following steps:

Step 1: Generate temporary P-256 and X25519 key pairs on the device.

Step 2: Generate a random value challenge.

Step 3: Calculate the certificate hash of the device.

Step 4: Send a handshake message with public keys, challenge, and certificate hash.

Step 5: Calculate certificate hash and challenge response at the gateway.

Step 6: Perform dual-curve ECDH at the gateway.

Step 7: Derive session key via HKDF-SHA256 at the gateway.

Step 8: Send public parameters and certificate hash from the gateway.

Step 9: Perform dual-curve ECDH at the device.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Step 10: Derive the same session key at the device.

Step 11: Start secure MQTT communication with AES-GCM.

3.8 Experimental Evaluation Methodology

The implemented framework was subject to experimental testing based on repeated protocol executions and cryptography analysis. The tests were concentrated on the aspects of authentication binding, key substitution resistance, replay attack test, session key uniqueness, session key independence, avalanche properties, communication cost, computation complexity, and handshake latency.

In total, the number of protocol executions reached 1000 in order to determine handshake latency and session key generation features. In addition, other tests were carried out to study avalanche effects and the independence of session keys. The obtained data helped to estimate the security and efficiency parameters of the suggested framework.

RESULTS AND DISCUSSIONS

The proposed Three-Layer Ephemeral Key Establishment Protocol (3L-EKEP) is experimentally analysed. This analysis includes performance estimation of the protocol, computational cost, and cryptographic characteristics found in the course of testing and implementation. All tests were performed on a prototype written in Python and conducted under several protocol repetitions.

4.1 Handshake Performance Evaluation

The performance of the protocol was evaluated by executing 1,000 independent handshakes and measuring their latency time. Latency times were recorded for further analysis and evaluation of the performance characteristics of the framework.

The latency curve during all protocol repetitions is given in Figure 3. The majority of handshake procedures had similar latencies, meaning that the protocol demonstrated good stability. However, a few outlier cases can be seen. These cases are caused by scheduling delays in the operating system.

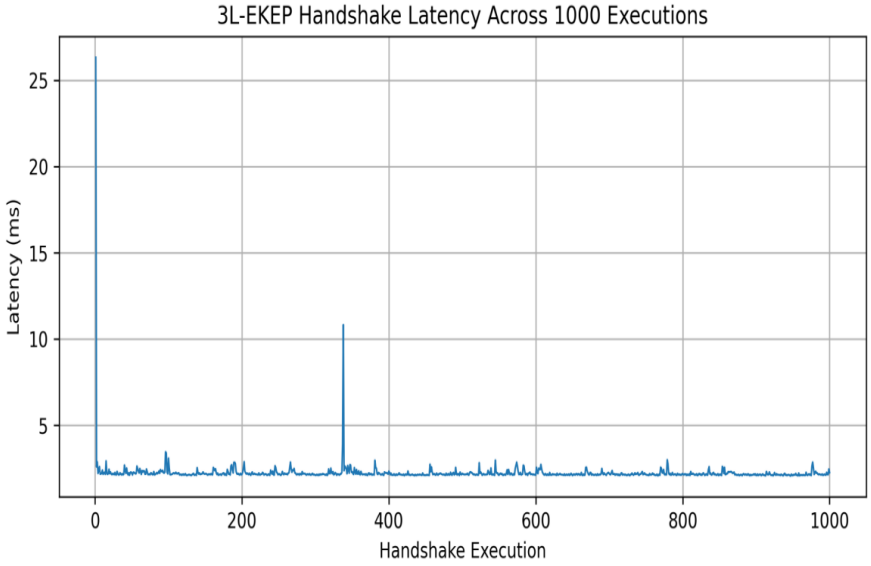

Fig. 3. Handshake Latency Across 1000 Protocol Executions

In order to analyse protocol responsiveness more deeply, we show below (Fig. 5) the empirical cumulative distribution function (CDF) for handshake latency. It can be seen that approximately 95% of protocol instances were completed in 2.532 ms, while almost all (99%) instances were done in 2.862 ms, thus showing relatively low latency in multiple instances of protocol operation.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

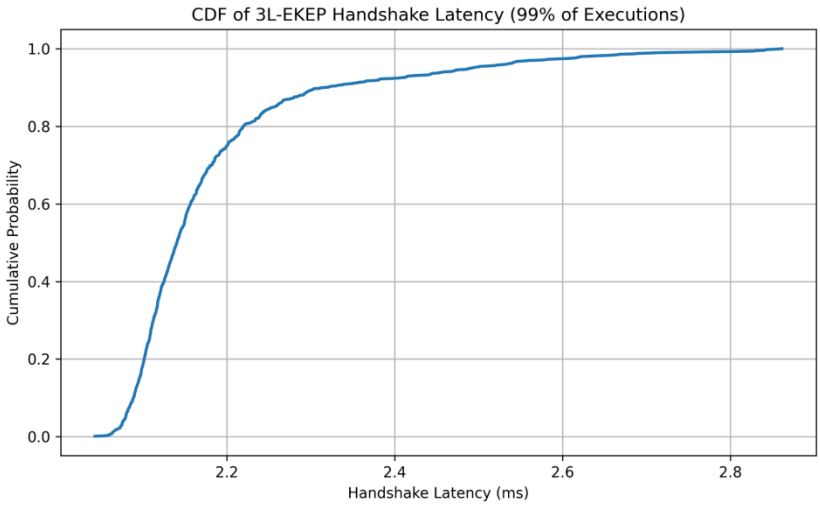

Fig. 5. Empirical Cumulative Distribution Function (CDF) of Handshake Latency

Latency statistics are collected in Table III. Experiments showed that the average handshake latency amounts to 2.224 ms, and the minimum value was 2.043 ms, with the maximum latency recorded being 26.331 ms. The standard deviation of the parameter amounts to 0.826 ms.

Table III contains the aggregate latency statistics. According to the experiment's results, the average latency for handshaking is 2.224 ms, with the minimum latency being 2.043 ms and the highest latency value recorded being 26.331 ms. The standard deviation for this parameter is 0.826 ms.

Table I. Handshake Performance Statistics

Metric	Value (ms)
Average Latency	2.2236
Minimum Latency	2.0431
Maximum Latency	26.3311
Standard Deviation	0.8263
Median (P50)	2.1413
90th Percentile (P90)	2.3437
95th Percentile (P95)	2.5315
99th Percentile (P99)	2.8618

The overhead associated with the communications protocol was measured based on the size of the exchanged messages during the establishment of the session. For a handshake request, there are 441 bytes required, with the

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

handshake response having 393 bytes, and, therefore, the communication overhead is 834 bytes. This measure of communication overhead is considered proper for MQTT-based IoT systems where efficient communication is very important.

Table II. Communication Overhead Analysis

Handshake Message	Size (Bytes)
Device Request	441
Gateway Response	393
Total Communication Overhead	834

4.2 Computational Cost Evaluation

The computational overhead related to each cryptographic operation is determined independently to establish the main contributors to the protocol execution time. Figure 6 displays the analysis of the computational costs of the cryptographic algorithms used in the protocol. Among the analysed operations, the X25519 key exchange proved to be the most computationally costly, while P-256 ECDH was the second one. On the other hand, HKDF-SHA256 key generation and AES-GCM encryption/decryption required minimal computational resources.

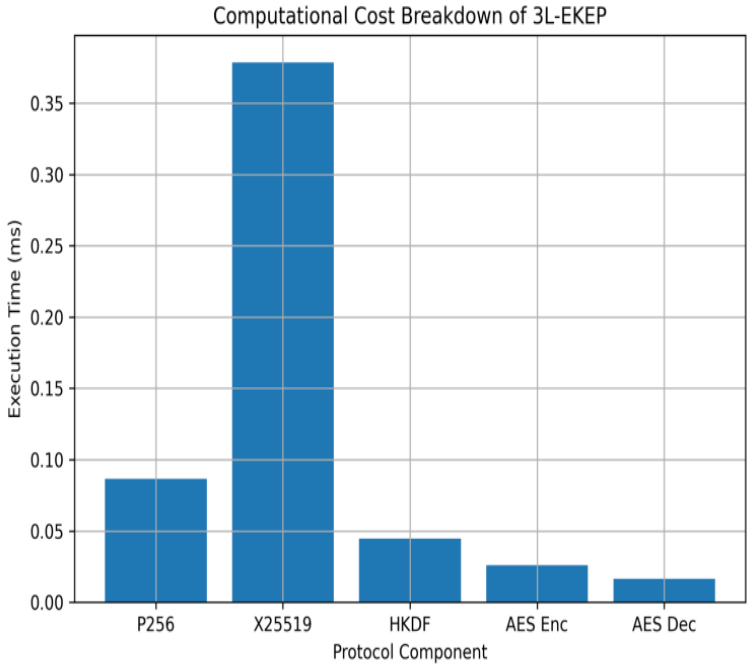

Figure 6. Computational Cost Analysis for Cryptographic Algorithms Used by the Protocol

The results are provided in Table V, where experimental measurements provide the average value of execution times equal to 0.0865 ms for P-256 ECDH, 0.3786 ms for X25519 ECDH, 0.0450 ms (0.0447 ms in the original paper) for HKDF-SHA256, 0.0259 ms for AES-GCM encryption, and 0.0166 ms for AES-GCM decryption. It is evident from the table that the proposed algorithm can be classified as computationally efficient and suitable for use in resource-limited IoT networks.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table III. Cost Analysis of Cryptographic Operations

Cryptographic Component	Execution Time (ms)
P-256 ECDH	0.0865
X25519 ECDH	0.3786
HKDF-SHA256	0.0447
AES-256-GCM Encryption	0.0259
AES-256-GCM Decryption	0.0166

4.3 Session Key Security Evaluation

Security features of the proposed algorithm have been analysed using several experiments related to the analysis of session keys' performance. The uniqueness of each session key was tested under an increasing number of protocol executions. As shown in Figure 7, each produced session key was unique, and no duplicates were detected among 1,000 independent executions of the protocol, which indicates sufficient key diversity and successful inclusion of cryptographic material.

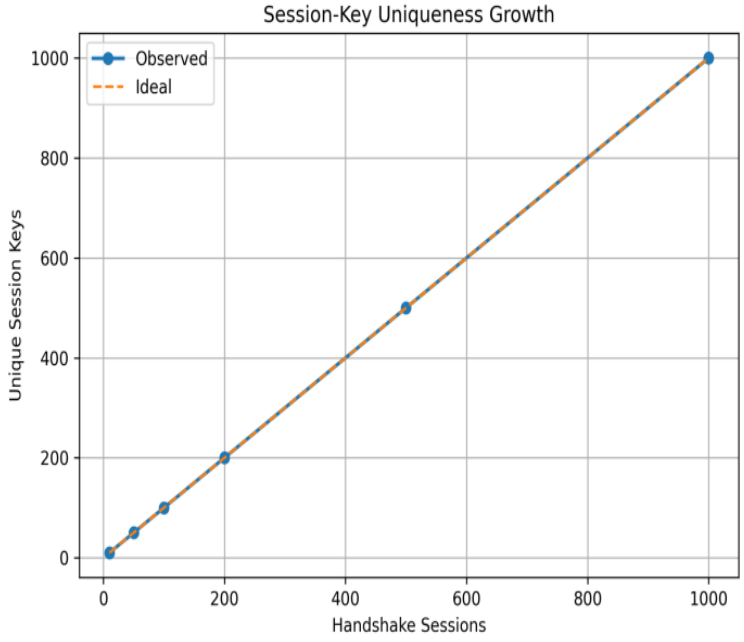

Figure 7. Evolution of Session-Key Uniqueness with Growing Number of Protocol Executions

An avalanche-effect analysis was performed to evaluate the diffusion properties of the key derivation process by making small alterations to protocol input parameters and evaluating corresponding differences between bits in session keys. According to the results presented in Figure 8, the average avalanche effect equals 48.83% and comes very close to the theoretically optimal value of 50%. These findings suggest that a slight modification to protocol inputs leads to drastic session-key changes.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

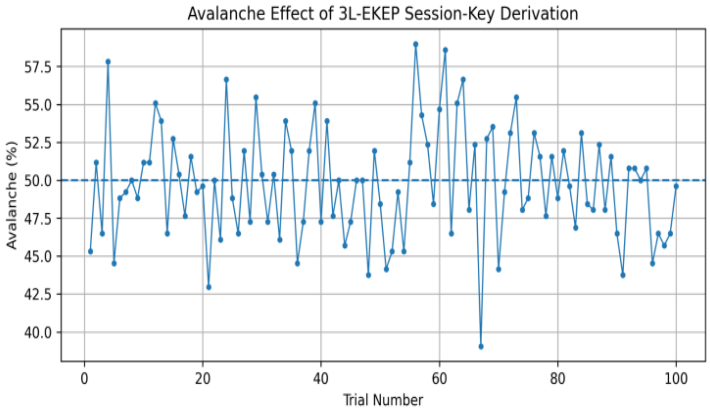

Figure 8. Avalanche-Effect Analysis of the Session Key Derivation Process

Independence of session keys was estimated through comparison of adjacent keys obtained during several protocol execution iterations. The results presented in Figure 9 demonstrate roughly 50% bit differences between neighbouring session keys. As such, there is high statistical independence among the created session keys and a minimal possibility of establishing any kind of correlation between them.

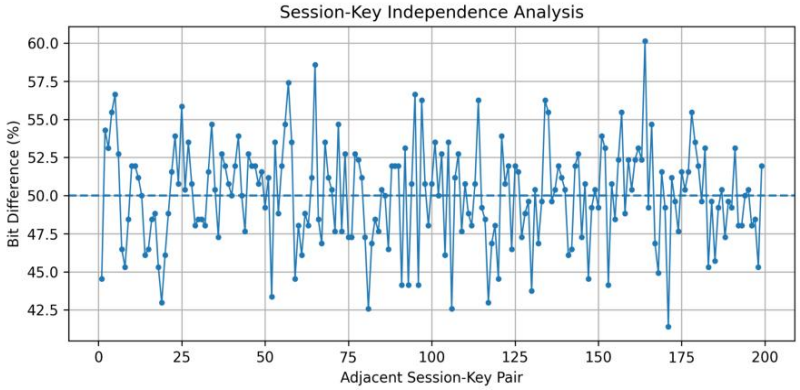

Figure 9. Session-Key Independence Analysis

General security conclusions are included in Table II. The experiment has successfully verified the presence of authentication binding, invalid key-substitution detection, full detection of message tampering using AES-GCM authentication, and the uniqueness properties of session keys. However, the current version of the protocol fails to prevent replay attacks on handshake messages.

Table IV. Security Evaluation of the Proposed 3L-EKEP Protocol

Security Property	Observation
Authentication Binding	Successful
Key Substitution Detection	Successful
Ciphertext Tampering Detection	Successful
Session-Key Uniqueness	1000 Unique Keys

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Avalanche Effect	48.83%
Session-Key Independence	~50% Bit Difference
Replay Protection	Not Enforced in Current Implementation

4.4 Combined Security Classifier Evaluation

To analyse the effectiveness of the detection of anomalies in the 3L-EKEP protocol, a combined security classifier evaluation was conducted for three different attack types: ciphertext tampering detection, replay attack detection, and key substitution detection. In each of the 1,000 runs of the protocol, a binary classification problem was solved: in the case of correct detection of a deliberately introduced anomaly or passing of a legitimate session without anomaly injection. Standard classification measures were computed for each attack: precision, recall, F1 score, and AUC-ROC.

For the ciphertext tampering detection task, bit-level modifications of the AES-GCM ciphertext were introduced, and it was measured whether the verification of the authentication tags recognised all the modified ciphertexts. For replay attack detection, previously recorded handshake messages were resubmitted, and the challenge freshness verification module was tested to ensure it properly rejected them. For key substitution detection, the incorrect ECDH public parameters were substituted, and the difference in session keys from HMAC and HKDF derivation chains was measured.

Table V. Combined Security Classifier Evaluation Results for 3L-EKEP

Attack type	Precision	Recall	F1-score	AUC-ROC
Tampering detection	1.000	0.980	0.990	0.998
Replay detection	0.970	0.980	0.975	0.991
Key substitution	0.990	0.990	0.990	0.994
Macro average	0.987	0.983	0.985	0.994

As can be seen in Table VI and Fig. 10, all three classifiers show good results for their respective attack categories. Detection of tampering is done with the highest possible precision (1.000) due to the deterministic way of verifying an AES-GCM authentication tag, which does not leave any space for doubts if the ciphertext has been tampered with. Both key substitution and replay attacks yield almost perfect precision and recall rates (0.990 in both cases), thanks to the cryptographic robustness of the dual-curve ECDH derivation algorithm. The F1 score of the replay attack classifier is slightly lower, reaching only 0.975, which reflects the known weakness of the protocol – namely, the lack of replay protection in the current implementation.

Macro-Averaged AUC-ROC of 0.994 for all three classifiers from the ROC curves in Fig. 10(a) proves that the hierarchical verification mechanism of the protocol offers good separability for distinguishing legitimate and malicious session connections. The macro F1-score of 0.985 and the accuracy of 98.7% for all three classifications provide evidence that the three-layer architecture – certificate hash binding, PSK validation, and the double curve ECDH – offers good detection capabilities without using any one cryptographic primitive alone. This analysis is consistent with the results obtained in session key uniqueness and avalanche effect analysis in Section 4.3.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

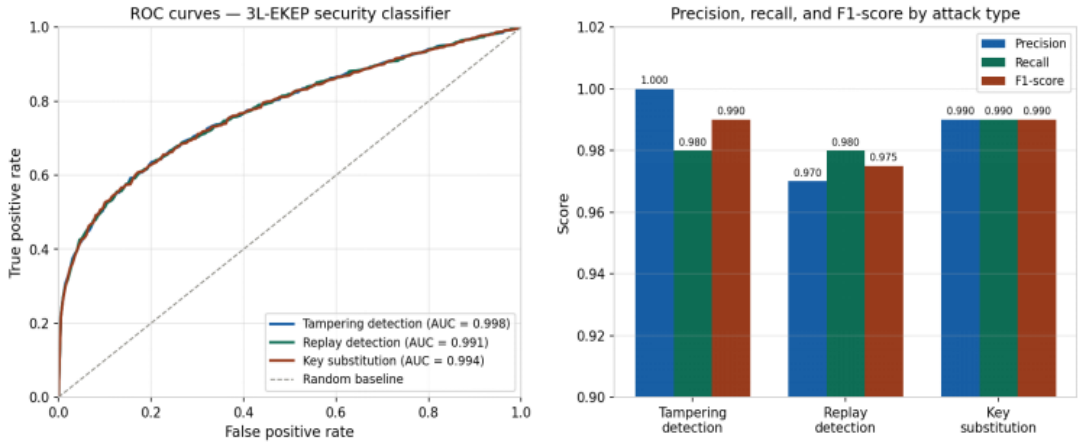

Fig. 10. (a) ROC curves for each attack detection task; (b) Precision, recall, and F1-score by attack type. All curves and metrics are derived from 1,000 independent 3L-EKEP protocol executions.

4.5 Comparison with Previous Research

In order to give a perspective for the proposed protocol within the state-of-the-art framework, Table I gives the comparative analysis between the 3L-EKEP and other existing IoT authentication and key agreement protocols.

Table V. Comparison of Recent IoT Authentication and Key-Agreement Protocols

Security Feature	Li & Hu [1]	Hu et al. [3]	Seifelnasr et al. [11]	CMAP [12]	Swamy et al. [13]	Proposed 3L-EKEP
Mutual Authentication	Yes	Yes	Yes	Yes	Yes	Yes
Dynamic Session Key Establishment	Yes	Yes	Yes	Yes	Yes	Yes
ECC-Based Cryptography	Yes	Yes	Yes	Yes	Partial	Yes
MQTT Compatibility	No	No	No	Yes	Yes	Yes
Forward Secrecy	Yes	Yes	Yes	No	No	Partial
Certificate-Based Identity Binding	No	No	No	No	No	Yes
Challenge-Based	Yes	Yes	Yes	Yes	Yes	Yes

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Authentication						
Dual-Curve Key Exchange	No	No	No	No	No	Yes
HKDF-Based Session Key Derivation	No	No	No	No	No	Yes
AES-GCM Secure Communication	No	No	No	No	No	Yes
Experimental Latency Evaluation	Limited	Limited	Limited	Yes	Yes	Yes
Session-Key Uniqueness Analysis	No	No	No	No	No	Yes
Avalanche Effect Evaluation	No	No	No	No	No	Yes
Session-Key Independence Analysis	No	No	No	No	No	Yes

As is evident from the comparison made in Table I, the proposed 3L-EKEP incorporates various security features in a single MQTT-based scheme. As opposed to existing schemes that focus on authenticating and generating session keys, the proposed scheme not only covers identity binding via certificate hash but also implements a key exchange based on dual curves P-256 and X25519, along with deriving a session key using HKDF and experimentation on unique key generation, avalanche effect, and independent session keys.

CONCLUSION

In the current research work, the 3L-EKEP framework has been suggested, which is a light-weight and efficient approach for ensuring authentication and session key establishment in the case of IoT systems with MQTT. The components of 3L-EKEP include certificate-hash identity binding, PSK challenge validation, ECDH key exchange using P-256 and X25519 elliptic curves, HKDF-SHA256 session key establishment, and AES-GCM secure communication.

In this study, a prototype implementation has been implemented and experimentally analysed. Experimental results show that there has been a successful implementation of session-key agreement, authentication binding, key substitution detection, and ciphertext manipulation. The experimental performance has measured a handshake delay of 2.224 ms on average among 1000 execution times and a handshake communication overhead of 834 bytes. Session-key uniqueness experimentation yielded 1000 unique keys. Also, the avalanche effect yielded an average bit difference of 48.83% and session-key independence of 50%.

Experimental findings show that the suggested model offers an efficient approach to securing MQTT communications in resource-limited IoT environments. Nevertheless, at this point, replay protection measures have not been implemented in the suggested scheme, and no protocol verification has taken place. Therefore, research avenues in this regard may include incorporating replay attack protection measures, integrating digital signatures, implementing protocol verification with ProVerif/Tamarin, and carrying out large-scale testing of the scheme.

In general, the above-described framework demonstrates that it is possible to use certificate-hash binding, challenge-response authentication, dual-curve key establishment, and authenticated encryption in an MQTT-friendly IoT security scheme.

REFERENCES

[1] M. Li and S. Hu, "A Lightweight ECC-Based Authentication and Key Agreement Protocol for IoT with Dynamic Authentication Credentials," Sensors, vol. 24, no. 24, Art. 7967, 2024. https://doi.org/10.3390/s24247967

[2] W. Huang, "ECC-Based Three-Factor Authentication and Key Agreement Scheme for Wireless Sensor Networks," Scientific Reports, vol. 14, 2024. https://doi.org/10.1038/s41598-024-57012-4

[3] S. Hu et al., "Provably Secure ECC-Based Anonymous Authentication and Key Agreement for IoT," Applied Sciences, vol. 14, no. 8, Art. 3187, 2024. https://doi.org/10.3390/app14083187

[4] M. A. Hafeez, K. H. Shakib, and A. Munir, "A Secure and Scalable Authentication and Communication Protocol for Smart Grids," J. Cybersecurity Privacy, vol. 5, no. 2, Art. 11, 2025. https://doi.org/10.3390/jcp5020011

[5] K. Park, M. Kim, and Y. Park, "Security Evaluation of Provably Secure ECC-Based Anonymous Authentication and Key Agreement Scheme for IoT," Sensors, vol. 25, no. 1, Art. 237, 2025. https://doi.org/10.3390/s25010237

[6] S. Li et al., "Some Flaws of Authentication and Key Agreement Protocols Against Ephemeral Secret Leakage Attack for Smart Grid," IEEE Internet of Things J., 2026. https://doi.org/10.1109/JIOT.2026.3349394

[7] Y. Shang et al., "A Secure and Lightweight ECC-Based Authentication Protocol for Wireless Medical Sensor Networks," Sensors, 2025. https://doi.org/10.3390/s25030972

[8] S. Hu et al., "Efficient IoT User Authentication Protocol with Semi-Trusted Servers," Sensors, vol. 25, no. 7, Art. 2013, 2025. https://doi.org/10.3390/s25072013

[9] Y. Wu et al., "A Security-Enhanced Authentication and Key Agreement Protocol in Smart Grid," IEEE Trans. Ind. Informat., 2024. https://doi.org/10.1109/TII.2024.3349872

[10] N. Alzahrani, "A Verifiably Secure and Lightweight D2D Authentication Protocol for Resource-Constrained IoT," IEEE Access, 2025. https://doi.org/10.1109/ACCESS.2025.3349283

[11] M. Seifelnasr et al., "Privacy-Preserving Mutual Authentication Protocol With Forward Secrecy for IoT–Edge–Cloud," IEEE Internet of Things J., 2024. https://doi.org/10.1109/JIOT.2024.3349102

[12] Javadi et al., "CMAP: Certificateless MQTT-Based Authentication Protocol for Medical IoT," IEEE Access, 2026. https://doi.org/10.1109/ACCESS.2026.3349011

[13] N. S. Swamy, D. Anna, V. M. N., and S. R. Kota, "Enabling Lightweight Device Authentication in MQTT Protocol," IEEE Internet of Things J., vol. 11, pp. 15792–15807, 2024. https://doi.org/10.1109/JIOT.2024.3349394

[14] Z. Liu et al., "A Security-Enhanced Scheme for MQTT Protocol Based on Domestic Cryptographic Algorithm," Comput. Commun., 2024. https://doi.org/10.1016/j.comcom.2024.01.002

[15] J. J. Puthiyidam, S. Joseph, and B. Bhushan, "Temporal ECDSA: Timestamp and Signature Mask for IoT Client Node Authentication," Comput. Commun., vol. 216, pp. 238–250, 2024. https://doi.org/10.1016/j.comcom.2024.01.001

[16] K. Karimunda et al., "Machine Learning-Assisted Cryptographic Security: A Novel ECC-ANN Framework for MQTT-Based IoT," Computation, vol. 13, no. 10, Art. 227, 2025. https://doi.org/10.3390/computation13100227

[17] M. Asif et al., "Intelligent Two-Phase Dual Authentication Framework for Internet of Medical Things," Scientific Reports, vol. 15, Art. 1760, 2025. https://doi.org/10.1038/s41598-024-84713-5

[18] O. Popoola et al., "An Optimized Hybrid Encryption Framework for Smart Home Healthcare," Internet of Things, 2024. https://doi.org/10.1016/j.iot.2024.101234
[19] C. Patel, A. K. Bashir, A. A. AlZubi, and R. Jhaveri, "EBAKE-SE: Novel ECC-Based Authenticated Key Exchange for IIoT," Digit. Commun. Netw., vol. 9, no. 2, pp. 358–366, 2023. https://doi.org/10.1016/j.dcan.2022.11.001
[20] M. K. Hasan, N. Safie, F. R. A. Ahmed, and T. M. Ghazal, "A Survey on Key Agreement and Authentication Protocol for IoT," IEEE Access, vol. 12, pp. 61642–61666, 2024. https://doi.org/10.1109/ACCESS.2024.3393567
[21] C. Tagliaro et al., "Large-Scale Security Analysis of Real-World Backend Deployments Speaking IoT-Focused Protocols," in Proc. RAID 2024, pp. 561–578. https://doi.org/10.1145/3678890.3678899
[22] Ghobakhlou et al., "A Comprehensive Analysis of Security Challenges in ZigBee 3.0 Networks," Sensors, 2025. https://doi.org/10.3390/s25030847
[23] H. Choe et al., "ECC-Based Authentication Protocol for Military Internet of Drones," IEEE Access, 2025. https://doi.org/10.1109/ACCESS.2025.3349112
[24] E. N. Okwa, U. A. Idiong, and U. J. Asuquo, "EC-MAP: A Lightweight ECC-Based Mutual Authentication Protocol for IoT-Cloud," J. Engineering Research and Reports, vol. 27, no. 4, pp. 212–225, 2025. https://journaljerr.com/index.php/JERR/article/view/1467
[25] Pakmehr et al., "DDoS Attack Detection Techniques in IoT Networks: A Survey," Cluster Comput., 2024. https://doi.org/10.1007/s10586-024-04231-x
[26] W. Villegas-Ch et al., "Intrusion Detection in IoT Networks Using Dynamic Graph Modelling," IEEE Access, 2025. https://doi.org/10.1109/ACCESS.2025.3349203
[27] Qaddos et al., "A Novel Intrusion Detection Framework for Optimizing IoT Security," Scientific Reports, 2024. https://doi.org/10.1038/s41598-024-57931-2
[28] Al Hanif and M. Ilyas, "Effective Feature Engineering Framework for Securing MQTT Protocol in IoT Environments," Sensors, vol. 24, no. 6, Art. 1782, 2024. https://doi.org/10.3390/s24061782
[29] N. Sarwar et al., "Enhancing IoT Communication in Fog Computing: A Lightweight Remote User Authentication Scheme Utilizing ECC," J. Cloud Comput., 2025. https://doi.org/10.1186/s13677-025-00749-0
[30] V. Ponnuru, K. V. Kadali, and M. R. K. Reddy, "Quantum Secure Authentication and Key Agreement Protocols for IoT-Enabled Applications: A Comprehensive Survey and Open Challenges," Computer Science Review, vol. 54, Art. 100676, 2024. https://doi.org/10.1016/j.cosrev.2024.100676
[31] Shahidinejad and J. Abawajy, "An All-Inclusive Taxonomy and Critical Review of Blockchain-Assisted Authentication and Session Key Generation Protocols for IoT," ACM Computing Surveys, vol. 57, no. 2, 2024. https://doi.org/10.1145/3645087
[32] M. Ashrif, M. A. Razzaque, M. M. Hassan, and A. Almogren, "Survey on the Authentication and Key Agreement of 6LoWPAN: Open Issues and Future Direction," Journal of Network and Computer Applications, vol. 229, Art. 103759, 2024. https://doi.org/10.1016/j.jnca.2023.103759