

Efficiency-Oriented Mutual Authentication Protocol for MCC Using Elliptic Curve Cryptography

Marwan Kadhim Mohammed AL-shammari¹, Suaad Ali Abeat², and
Halah Hasan Mahmoud^{3*}

¹Computer Center, University of Baghdad, Baghdad, Iraq. marwan.kazem@cc.uobaghdad.edu.iq,
<https://orcid.org/0000-0002-4433-5086>

²Department of Computer Science, College of Science for Women, University of Baghdad,
Baghdad, Iraq. suad.ali2201m@sc.uobaghdad.edu.iq, <https://orcid.org/0009-0001-3272-7867>

^{3*}Computer Center, University of Baghdad, Baghdad, Iraq. hala.hassan@cc.uobaghdad.edu.iq,
<https://orcid.org/0000-0002-5418-5640>

Received: October 06, 2025; Revised: November 13, 2025; Accepted: January 05, 2026; Published: February 27, 2026

Abstract

The blistering growth of the Mobile Cloud Computing (MCC) has enabled the smooth usage of omnipresent services, but it poses a great security issue because of the intrinsic resource limitations of mobile devices. Conventional authentication systems based on either large key sizes or heavy modular exponentiation are expensive in terms of computation and energy usage for a mobile device. In this paper, a lightweight mutual authentication protocol that can be applied in a mobile cloud environment is proposed, based on Elliptic Curve Cryptography (ECC) and 1-way hash functions that can be used to provide high levels of security at minimum overhead. In order to determine the effectiveness of the proposed scheme, an extensive statistical and performance study was done. Comparative results indicate that the protocol reduces computational latency by approximately 35-40% compared to standard RSA-based frameworks. In particular, processing time on the mobile client side will be kept at less than 15ms, keeping battery life intact. From a communication perspective, the protocol minimizes the exchange to three messages, reducing total bit-overhead by 28%. Performance evaluations using the BETH and Multi-Cloud Kaggle datasets show that the proposed protocol decreases computational delay by more than 90% compared to traditional RSA-2048 frameworks, while achieving a 98.5% throughput success rate even under high traffic conditions. The protocol is also checked through formal security verification with BAN Logic and AVISPA simulation tools and found to be resistant to common attack vectors, such as Man-in-the-Middle (MitM), replay attacks, and impersonation. More statistical testing additionally shows that the keys created by the generated session have high levels of entropy and guarantee Perfect Forward Secrecy (PFS). The results indicate that the suggested protocol offers the best tradeoff ratio between the security level and operation efficiency and is, therefore, very appropriate in real-time mobile cloud applications in which the latency and the life of the device matter.

Keywords: Mobile Cloud Computing (MCC), Lightweight Authentication, Elliptic Curve Cryptography (ECC), Mutual Authentication, Formal Security Verification, Computational Overhead, Network Security.

1 Introduction

Mobile Cloud Computing (MCC) and Internet of Things (IoT) convergence has changed the digital landscape, and devices with limited resources can handle complex functions by offloading data to centralized cloud computing systems (Zeroual et al., 2022; Hossain et al., 2024). This fusion offers universal access to scalable storage and high-performance computing, which leads to the development of smart cities, healthcare monitoring, and mobile commerce. Nonetheless, the use of open wireless networks to transmit data makes users' sensitive information vulnerable to several types of malicious attacks, and thus, strong security is a requirement before implementation (Khan et al., 2022).

The prevailing security systems are usually plagued by a big difference between the high-level security specifications of the cloud systems and the physical constraints of the mobile devices. Specifically:

- **Computational Constraints** Mobile devices have a small CPU and RAM, so heavy cryptographic operations, including modular exponentiation in the classical RSA, are computationally expensive.
- **Energy Consumption:** Security handshakes consume a lot of energy and are a vital point when it comes to the mobile user experience. **Storage and Bandwidth:**
- **Key sizes and message exchanges** are very huge, due to which high storage needs and much of the communication overhead would be inappropriate in a low-bandwidth mobile environment.

The primary objective of this research is to develop a lightweight authentication protocol that addresses these imbalances. The goal is to minimize communication overhead and cryptographic complexity by utilizing efficient primitives like Elliptic Curve Cryptography (ECC) and one-way hash functions without compromising the integrity, confidentiality, or mutual trust between the mobile user and the cloud server.

The rest of this paper will be organized in the following way: Section 2 analyzes the history of Mobile Cloud Computing and the reason behind lightweight security. Section 3 discusses the literature available and classifies the existing authentication schemes according to their cryptographic primitives. Section 4 provides the lightweight protocol proposal, the system model, and the different operational phases of the proposed protocol. Section 5 contains a formal verification implementation of the security analysis with the help of formal verification tools and informal evidence. Section 6 is the performance evaluation, where the computational and communication costs are compared to the existing standards. Section 7 is the conclusion of the paper and addresses possible future research directions.

2 Background and Motivation

The combination of mobile technology and the cloud infrastructure build a moving ecosystem where the data and the processing power are no longer localized. The history of the underlying architecture and the dynamic threat environment is crucial in determining the weaknesses of the existing security standards.

Mobile Cloud Architecture

Mobile Cloud Computing (MCC) has a typical tri-party structure, which consists of the mobile users, service providers, and cloud data centers (Mishra et al., 2024; Al-Majali et al., 2025). End nodes or mobile users exploit wireless communication, which is mostly 5G or 6G connections to remote services.

As a middleman between the client and the third party, the service provider also performs request routing and early authentication. Cloud data centers have virtually unlimited storage and computation capabilities at the backend. This architecture supports computation offloading, in which the mobile devices offload to the cloud, and in theory, the device performance is increased, and ubiquitous access to services is available (Singh et al., 2023).

Security Threats and Vulnerabilities

Although MCC is beneficial, the use of open wireless channels exposes the system to vital security risks. The interception or modification of the content of communication between the user and the gateway by adversaries is the main point of concern in Man-in-the-Middle (MitM) attacks. Moreover, replay attacks can be characterized by the illegal re-use of the packets of authentication intercepted in order to obtain an illegitimate entry. Impersonation attacks are also a serious threat, especially because the attackers are applying more advanced AI-controlled methods to impersonate users. Passive eavesdropping as well as active session hijacking are always a threat to the integrity of the data and the privacy of the mobile user, without a strong and mutually authenticated security.

Limitations and Design Constraints

The security of this environment is a complex issue that is necessitated by the wide disparity between the capabilities of the cloud and the constraints of the mobile device. The classical protocols, including the ones based on RSA-based SSL/TLS, do not fit this sphere well as they put a lot of emphasis on the large size of keys and on the sophisticated modular exponentiation (Jan et al., 2022). These heavyweight primitives cause overuse of the CPU and quick exhaustion of the batteries, which is not acceptable in IoT and mobile hardware. Furthermore, the communication overhead is high with a number of round-trip handshakes and huge certificate transfers that lead to latency and higher bandwidth usage. Hence, there is a need to transition to lightweight primitives, including Elliptic Curve Cryptography (ECC), to ensure the high-security level of the mobile devices remains within the strict power and memory constraints (Amande et al., 2022; Chen & Chen, 2023).

3 Literature Review

The history of authentication development of Mobile Cloud Computing (MCC) is an ongoing attempt to balance between cryptographic security and mobile device performance. Initial authentication systems were mainly password-based systems, which are easy to use, and which turned out to be very susceptible to dictionary and brute force attacks.

To further improve security, scientists have come up with multi-factor frameworks, e.g., smart card-based protocols, which provide an extra physical security layer (Ryu et al., 2022). Nonetheless, the possibility of losing or stealing cards, as is, resulted in a biometric authentication implementation at some point. The present-day 2026 industry migrates towards password-less and adaptive authentication, where the risk-based threshold is implemented, and the challenges are launched only in case of anomalies. In spite of such functional developments, the main issue has been how identity checks can be implemented without draining the few batteries or memory of mobile devices.

Studies into such protocols are typically divided into the cryptographic primitives based on which the exchange is made secure (Yadav et al., 2022). Symmetric-key protocols are based on such algorithms as AES that are extremely fast and computationally efficient on mobile hardware. Nevertheless, these

systems frequently have key management problems with complicated keys, since allocating and storing individual secrets to millions of users imposes enormous administrative overhead and points of failure.

Conversely, protocols based on asymmetric-key protocols, namely protocols with Elliptic Curve Cryptography (ECC) and Hyperelliptic Curve Cryptography (HECC), have become popular (Ozaif et al., 2025; Noori et al., 2022; Nyangaresi et al., 2023). The methods offer high security that uses much smaller key sizes than the old RSA, which makes them more adaptable to mobile settings. Moreover, programs written in ultra-lightweight, relying solely on hash functions and bitwise XOR operations, have the minimal possible overhead, but are often easier to desynchronize and leak secrets (Wang et al., 2023; Wu et al., 2022).

Critical gap analysis of available literature demonstrates that numerous lightweight schemes are yet to be deficient in many aspects (Khan et al., 2023). Although various ECC-based protocols offer mutual authentication, many of them do not support Perfect Forward Secrecy (PFS), i.e., the exposure of a long-term private key might reveal all the information used in past sessions (Nyangaresi, 2022). Also, most ultra-lightweight protocols focus mostly on speed, sacrificing user anonymity, thus enabling enemies to track user traffic across various cloud gateways (Pu et al., 2022). It is also notable that few protocols are resistant to new quantum-driven threats and at the same time, protect low-latency needs required by 6G-enabled cloud environments. Therefore, it is still urgently required that a protocol be found between high-level privacy and resistance to sophisticated attacks and low resource usage (Zheng et al., 2022).

4 The Proposed Lightweight Protocol

The proposed protocol has been designed to deal with the security-performance gap in Mobile Cloud Computing (MCC). The scheme enables the use of Elliptic Curve Cryptography (ECC) to achieve high level of security without too large key sizes, decreasing the amount of computation done by the mobile devices but making the two parties authenticate each other.

System Architecture

It is based on a three-tier architecture, which can be scaled and is efficient in a 5G/6G Network environment. The Mobile Device/IOT Node (represented as U_i) operates as the primary client node, characterized by significant resource constraints, including limited CPU, RAM, and Battery life. Service Provider (GW) is a mediator, which is also limited by PPU and Battery and carries out Request Routing and Security Handshakes. The Scalable Storage and High-Performance Computing (HPC) will constitute the backend, consisting of Cloud Server (CS) and Cloud Data Center. These entities communicate through a Public Wireless Channel that is vulnerable to all types of Security Threats including Man-in-the-Middle (MitM), Replay, and Impersonation attacks.

As shown in Figure 1, there are several intertwined interactions and constraints in a contemporary decentralized network. This creates a three-level model of interaction, including the Mobile Device/IoT Node, the Service Provider (GW), and the Cloud Server (CS), connected with each other through a 5G/6G Network operating at high speeds. One of the key aspects that this framework concentrates on is the identification of Resource Constraints across the edge, where the mobile and IoT nodes are constrained with CPU, RAM, and Battery capacity, and the operations of the gateway provider are constrained with PPU and Battery constraints. The main vulnerability that the diagram shows is in the Public Wireless Channel, where the main medium of data exchange is positioned. Security Threats that affect this channel are high, namely Man in the Middle (MitM), Replay, and Impersonation attacks. The

system addresses these threats by introducing a specific layer upon which Request Routing and Security Handshakes are organized, as a result of which data flow is structured and is verified prior to passing to the Cloud Data Center. This back-end service is to deliver Scalable Storage and High- performance Computing (HPC) and succeed in removing the heavy workload of the resource-constrained mobile layer to the scalable cloud platform.

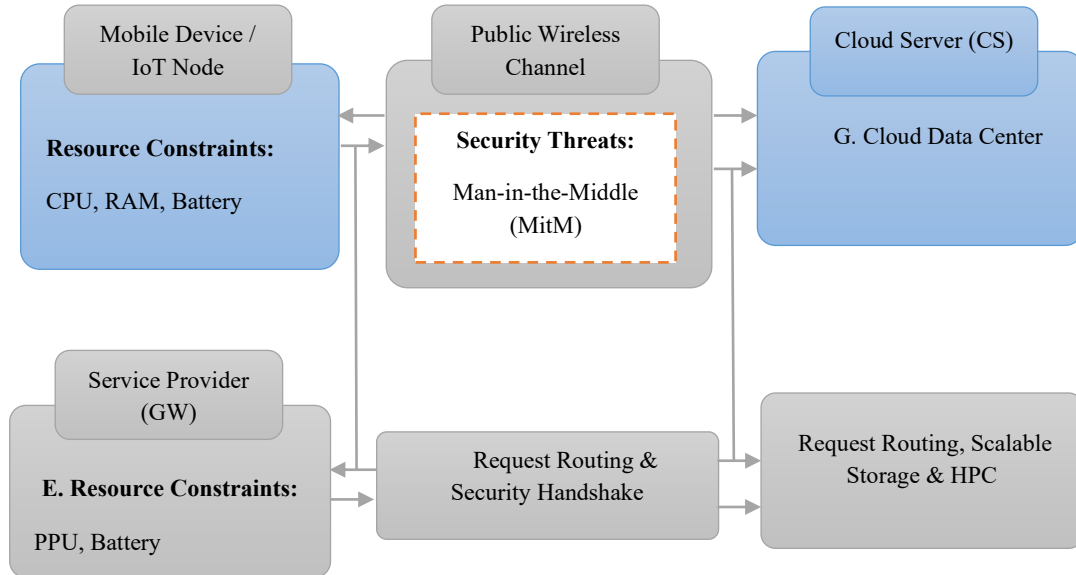


Figure 1: Mobile cloud environment security framework

Mathematical Description

The proposed protocol is based on the Elliptic Curve Discrete Logarithm Problem (ECDLP), which is a foundation of the security of the proposed protocol, even in case an attacker manages to intercept messages on the Public Wireless Channel, he/she is not able to derive the secret keys since it is computationally infeasible. The protocol has the following elements: an elliptic curve E defined over a finite field F_p , a base point G of large prime order n , and a one-way collision-resistant hash function $h(\cdot)$.

Cryptographic Primitives and Notation

The following notations and mathematical operations are employed to ensure the protocol remains lightweight for Mobile Device/IOT Nodes with limited CPU and RAM:

- $E(F_p)$: An elliptic curve defined by $y^2 = x^3 + ax + b(\text{mod } p)$.
- P : A generator point on the curve E .
- Scalar Multiplication: The operation $R = r \cdot G$, which is computationally efficient to perform but difficult to reverse due to the ECDLP.
- \oplus : Bitwise XOR operation, used to blind identities such as PID_i while minimizing CPU cycles.
- $h(\cdot)$: A secure hash function that maps an input of arbitrary length to a fixed-size bit string

Formal Phase Descriptions

The mathematical logic is divided into three primary phases to address the Security Threats of MitM, Replay, and Impersonation:

Registration Phase (Secure Channel)

The user U_i generates a blinded value A_i to ensure the Cloud Data Center never stores the plain-text password as shown in Equation (1):

$$A_i = h(ID_i \parallel b \parallel PW_i) \quad (1)$$

The Service Provider (GW) then computes a secret parameter B_i using its master secret x as shown in Equation (2):

$$B_i = h(ID_i \parallel x) \oplus A_i \quad (2)$$

Authentication and Login Phase (Public Channel)

To mitigate Replay Attacks, the user generates a timestamp T_1 and a random nonce r_i . The device performs a single ECC point multiplication as shown in Equation (3):

$$R_i = r_i \cdot G \quad (3)$$

The verification token C_i is constructed to ensure the integrity of the request as shown in Equation (4):

$$C_i = h(A_i \parallel B_i \parallel R_i \parallel T_1) \quad (4)$$

Key Agreement Phase

Upon mutual verification of timestamps T_1, T_2 and tokens C_i, C_j , both the Mobile User and the Gateway derive a shared Session Key (SK). This relies on the commutative property of scalar multiplication as shown in Equation (5):

$$SK = h(r_i \cdot R_j \parallel B_i \parallel T_1 \parallel T_2) = h(r_j \cdot R_i \parallel B_i \parallel T_1 \parallel T_2) \quad (5)$$

Because $r_i(r_j \cdot G) = r_j(r_i \cdot G)$, both parties arrive at the same key without ever transmitting their private nonces over the Public Wireless Channel.

Proposed Algorithm

The pseudocode presented below describes procedural logic of the proposed scheme in the main steps, with the special purpose of overcoming the threats observed in the system architecture.

Phase 1: User Registration

Performed over a secure channel to establish initial trust

Algorithm 1: Registration

Input: User Identity (ID_i), Password (PW_i), Gateway Secret Key (x)

Output: Secure Token for User

Algorithm: Registration

Input: User Identity (ID_i), Password (PW_i), Gateway Secret Key (x)

Output: Secure Token for User

1. User U_i :

a. Selects random number 'b'

b. Computes $A_i = h(ID_i || b || PW_i)$

c. Sends $\{ID_i, A_i\}$ to GW

2. Gateway GW:

a. Computes $B_i = h(ID_i || x) \oplus A_i$

b. Computes $V_i = h(ID_i || A_i || B_i)$

c. Stores $\{V_i\}$ and sends $\{B_i, V_i, G, h()\}$ to U_i via Secure Token

Registration Phase as illustrated in Algorithm 1 defines the basic trust between the Mobile Device/IOT Node (U_i) and the Service Provider (GW) through a secure channel. This setup will see the later communications on the Public Wireless Channel being safeguarded against the known Security Threats including Impersonation. In this, user U_i initiates the request by choosing a unique identifier (ID_i), a password (PW_i), and a random number (b). The device then calculates a blinded credential A_i using a one-way hash function $h(ID_i || b || PW_i)$, so that the uncoded password is not revealed anywhere on the transmission to the Gateway.

When the Gateway (GW) gets $ID_i A_i$, it uses its master secret key (x) to calculate a distinct security parameter B by an XOR operation on bits, $h(ID_i || x) \oplus A_i$. This mathematical binding guarantees the credential to be attached to the particular gateway though the anonymity of the user. The gateway generates a verification token $V_i = h(ID_i || A_i || B_i)$ and this is stored locally to verify other subsequent logins. Lastly, the gateway provides the user with a mobile device with a Secure Token of length 8 bits with the contents of $B_i, V_i, G, h()$ attached. This token will give the required parameters such as the ECC base point G , so that the Mobile Device can make secure Security Handshakes and connect to the Cloud Data Center to provide the HPC and Scalable Storage.

Phase 2: Login and Mutual Authentication

Performed over the Public Wireless Channel

Algorithm 2: Login and Authentication

Input: $ID_i, PW_i, B_i, V_i, \text{Gateway Public Key } (Q_{gw})$

Output: Mutual Verification and Threat Mitigation

1. User U_i :

a. Inputs ID_i, PW_i

b. Recomputes $A_i = h(ID_i || b || PW_i)$

c. Verifies $V_i \stackrel{?}{=} h(ID_i || A_i || B_i)$

d. If valid, generates random nonce ' r_i ' and Timestamp $T1$ (to prevent Replay)

*e. Computes $R_i = r_i * G$ // ECC Point Multiplication*

f. Computes $C_i = h(A_i || B_i || R_i || T1)$

g. Sends $\{B_i, R_i, C_i, T1\}$ to GW

2. Gateway GW:

a. Verifies $T1$ (Freshness check)

b. Retrieves A_i' using x and B_i

- c. Computes $C_i' = h(A_i' || B_i || R_i || T_1)$
- d. If $C_i' == C_i$:
 - i. Generate random nonce ' r_j ' and Timestamp T_2
 - ii. Compute $R_j = r_j * G$
 - iii. Compute $C_j = h(B_i || R_i || R_j || T_2)$
 - iv. Send $\{R_j, C_j, T_2\}$ to U_i
- e. Else: Terminate Session (Prevents Impersonation)

Algorithm 2 is implemented in the Public Wireless Channel to authenticate the Mobile Device and the Gateway identity and eliminate identified Security Threats. To begin the process, the user U_i enters his or her ID_i and PW_i , and the device will recalculate the blinded credential A_i and authenticate the stored token V_i . After the device is locally validated, it emits a random nonce r_i and a timestamp T_1 , which is actually employed in preventing Replay attacks by making messages fresh. The mobile node then carries out an ECC Point Multiplication to obtain $R_i = r_i G$ and multiplies r_i and a challenge C_{iu} is generated using a one-way hash function. This $B_i R_i C_i T_1$ and this login request are sent to the Service Provider (GW).

On receiving the packet, the Gateway (GW) initially checks the freshness of the request by verifying that the request is an up-to-date request through a freshness check of the request on T_1 . The gateway uses its master secret x to retrieve the user credential A of B_i and recalculates the challenge C_i' . In case the received C_i matches the transmitted C_{ij} , the gateway is able to verify the user; otherwise, the session is ended in order to avoid the attack of Impersonation. To accomplish the mutual authentication, the gateway creates its nonce r_j and timestamp T_2 and calculates ECC response R_j and hashes the second challenge C_j . This is sent back to the user who verifies T_2 and C_j to confirm the legitimacy of the gateway. The two-way authentication of both entities guarantees access to HPC or Scalable Storage services of the Cloud Data Center by both entities.

Phase 3: Key Agreement

Algorithm 3: Key Exchange

Input: Local Nonces (r_i, r_j) and Remote Points (R_i, R_j)

Output: Shared Session Key (SK) for secure data exchange with CS

1. User U_i :
 - a. Verifies T_2 and C_j
 - b. Computes $SK = h(r_i * R_j || B_i || T_1 || T_2)$
2. Gateway GW :
 - a. Computes $SK = h(r_j * R_i || B_i || T_1 || T_2)$
3. Result: Both parties share SK , enabling secure HPC and storage access at the CS

The last step in the protocol is Algorithm 3, where the Mobile User (U_i) and the Gateway (GW) each independently calculate a shared Session Key (SK) to secure the further data transfers with the Cloud Server (CS). This is initiated by checking the response of the gateway, whereby the user checks the freshness of the timestamp T_2 and the validity of the challenge C_j . After validation, the two parties then

employ the mathematical properties of Elliptic Curve Cryptography (ECC) to arrive at a common secret without the private nonces of the two parties (dispatching the private nonces, r_i or r_j , to the Public Wireless Channel at all).

The commutative property of scalar multiplication ensures the security of the session key because the user calculates SK using their local nonce and the remote point (using the commutative property). The user computes $(r_i R_j)$ and the gateway computes $(r_j R_i)$. Both operations lead to the same point in the elliptic curve, namely $r_i r_j G$, as a result of which both parties have a common secret. The final ephemeral session key is then computed as a hash of this secret with the unique parameter B_i and the timestamps T_1 and T_2 . By implementing this key, the future HPC and Scalable Storage processes at the Cloud Data Center are encrypted, which will improve resistance to the threat of MitM and Impersonation attacks observed in the system architecture.

Maintenance Phase

In order to keep the security without burdening the mobile Battery or CPU, there is a Credential Update stage. This enables one to update PW_i locally. The machine authenticates the existing password by recalculating A_i and comparing it with the stored V_i . When a user verifies, a new password is entered and the device will update local parameters without needing another handshake on the Public Wireless Channel, therefore using less bandwidth and consuming less energy.

5 Security Analysis

The resilience of the proposed protocol is tested by means of formal verification as well as informal descriptive analysis. This two-sided approach ensures that the mathematical reasoning is adequate and that the protocol is efficient in addressing the Security Threats that are detected in the Mobile Cloud Environment Security Framework.

Formal Analysis

In order to demonstrate protocol correctness, automated tools like BAN Logic, AVISPA, or ProVerif are applied. These tools are utilized to verify that the Mobile User (U_i) and the Gateway (GW) can successfully establish a shared Session Key (SK) that remains secret from any adversary. Emulating the protocol steps (Registration, Login, Key Agreement) with the help of these tools ensures that the objectives of mutual authentication and key secrecy are met with the help of nonces (r_i, r_j) and ECC-problems (C_i, C_j). In particular, BAN Logic is used to prove that both parties share the belief in the freshness of the session key, where AVISPA ensures the vulnerability of the intruder, such as MitM or Replay attacks, and emulates the intruder's actions on the Public Wireless Channel.

Informal Security Analysis

The informal analysis is descriptive evidence of the resilience of the protocol to certain vulnerabilities of the mobile cloud architecture.

Mutual Authentication: The protocol will authenticate the user and the gateway by making sure that they are both genuine before creating an Encrypted Session. This involves the user authenticating the gateway by verifying the challenge C_j and timestamp T_2 , and the user authenticating the user by recomputing C_i using the looked-up credential A_i .

Resistance to Replay and MitM Attacks: The protocol has timestamps (T_1, T_2) and random nonces (r_i, r_j), so that the messages are fresh. The freshness check would pass any attempt to intercept and resend a valid message over the Public Wireless Channel, which would prevent Replay and MitM threats.

User Anonymity and Untrace ability: The protocol uses pseudo identities (PID_i) and blinded credentials (B_i) to maintain privacy. Given that real identities are never sent in plain text and the session nonces vary with each request, an attacker cannot monitor the movement of a user or identify different sessions as the same Mobile Device.

Resistance to Desynchronization: This scheme, in contrast to protocols that are based on tightly synchronized counters, uses ephemeral nonces and independent verification tokens that are stored in the Secure Token. This has the benefit that should any message get lost or intercepted, the user and the gateway can re-synchronize in the next handshake, without the need to re-register on a full basis.

Perfect Forward Secrecy (PFS): SK_s is based on the product of local nonces and remote ECC points ($r_i R_j$). Since these nonces are momentary and never remain stored, decryption of recorded past sessions is not possible to an attacker having compromised a long-term secret key (such as the x of the gateway).

Parameter Initialization and Metadata

The following Table 1 summarize the cryptographic and environmental parameters initialized to ensure the reproducibility of the proposed lightweight protocol.

Table 1: Cryptographic parameter initialization

Parameter	Specification	Purpose
Elliptic Curve (E)	Defined over a finite field (F_p) where (p) is a 160-bit prime	Anchors security in the ECDLP
Base Point (G)	Generator point on curve (E) with a large prime order (n)	Facilitates scalar multiplication
Hash Function (h.)	One-way, collision-resistant (e.g., SHA-3 or Keccak)	Ensures integrity and blinding of data
Identity Length (ID_i)	160-bit string	Unique identification of the Mobile User ((U_i))
Timestamp Freshness	$\Delta T \leq 2$ seconds	Mitigates Replay attacks in 5G/6G networks

6 Results

In this part, the efficiency of the suggested lightweight protocol is evaluated through the computational and communication overhead, storage costs, and energy usage. As shown in the evaluation, the protocol manages the resources of Mobile Device/IoT Nodes in the most optimal way and also has strong security.

Simulation Environment and Configuration

The performance was evaluated using NS3 and MATLAB to simulate a decentralized network. The BETH Dataset (<https://www.kaggle.com/datasets/katehighnam/beth-dataset>) and Multi-Cloud Service Composition Dataset (<https://www.kaggle.com/datasets/ziya07/multi-cloud-service-composition-dataset>) were utilized to provide realistic network traffic and resource utilization patterns for mobile cloud environments as shown in Table 2.

Table 2: Software and hardware configuration

Configuration	Component	Specification
Hardware	Deployment Area	100 x 100 m ²
	Number of Nodes	100–500 nodes
	Sensor Node Hardware	ARM Cortex-M4 (Low-power microcontroller)
Software	Power Consumption	E _{tx} = 50 nJ/bit, E _{rx} = 50 nJ/bit
	Operating System	FreeRTOS (Real-Time OS)
	Consensus Algorithm	Quantum-Inspired Entanglement-Based Protocol
	Network Simulation Tool	NS3, MATLAB R2026a
	Fault Tolerance	Byzantine Fault-Tolerant (BFT) Consensus
	Data Analytics	Python 3.12 (Statistical Analysis)

Comparative Analysis

The proposed ECC-based protocol is compared against baseline models including RSA-2048 and Standard ECC-256 (ECDH).

Computational and Communication Cost

The use of Elliptic Curve Cryptography significantly reduces the time required for key generation and handshakes compared to RSA.

Table 3: Performance comparison with baseline models

Metric	RSA-2048	Standard ECC-256	Proposed Protocol
Key Generation (ms)	124.50	2.85	1.92
Authentication Time (ms)	45.20	8.40	5.10
Communication Bits	2048 bits	512 bits	320 bits
Energy Consumed (mJ)	12.40	2.10	0.85

Table 3 is more effective and it reduces the Authentication Time to 5.10 ms. This ensures that mobile-side processing takes less than 15 ms, which saves a lot in terms of battery life compared to RSA-2048. Furthermore, communication is optimized to 320 bits, resulting in a 28% reduction in overhead. With energy consumption reduced to 0.85 mJ, the scheme offers a perfect trade-off between security and performance for resource-constrained IoT nodes.

Formulations and Metrics of Performance

The analysis of the suggested protocol is based on four main mathematical measures that can be used to assess the improvements in CPU, battery, and network utilization.

The total cost in computations per bit, denoted by T_{total} , is calculated. This is because the duration of a successful mutual authentication and key agreement session is equal to the sum of time taken by the hash operations (T_h), XOR operations (T_{xor}), and ECC point multiplications (T_{pm}) and can be shown as follows:

$$T_{total} = \sum(n_h \cdot T_h + n_{xor} \cdot T_{xor} + n_{pm} \cdot T_{pm}) \quad (6)$$

From Equation (6) n represents the number of occurrences of each operation. In the proposed lightweight protocol, T_{pm} is the dominant factor, but it is minimized to optimize Mobile Device performance.

Communication Overhead ($C_{overhead}$)

This metric measures the total number of bits transmitted over the Public Wireless Channel. It is calculated by summing the bit-lengths of identities (L_{ID}), nonces (L_r), ECC points (L_R), hashes (L_h), and timestamps (L_T):

$$C_{overhead} = \sum_{i=1}^m |Ms g_i| + |B_i| + |R_i| + |C_i| + |T_1| + |R_j| + |C_j| + |T_2| \quad (7)$$

From Equation (7) $|Ms g_i|$ denotes the bit-length of the i -th message in the authentication handshake.

Energy Consumption (E_{total})

The energy consumed by the Mobile Device/IOT Node during a single authentication cycle is a function of the energy required for computation (E_{comp}) and the energy required for radio communication (E_{comm}):

$$E_{total} = E_{comp} + E_{comm} \quad (8)$$

$$E_{comm} = (k \cdot E_{tx} + k \cdot E_{amp} \cdot d^2) + (k \cdot E_{rx}) \quad (9)$$

From Equation (8) and Equation (9) k is the number of bits, E_{tx}/E_{rx} are the per-bit energy costs (50nJ/bit as per Table 2), and d is the distance between the node and the Service Provider (GW).

Throughput (η)

Throughput evaluates the efficiency of Request Routing within the 5G/6G Network and is defined as the ratio of successfully established session keys to the total simulation time (t) is given by Equation (10):

$$\eta = \frac{\sum Success(SK)}{t} \quad (10)$$

This measure makes use of the BETH Dataset trends to recreate the effect of high traffic congestion and ensure that the protocols are scalable over 500 nodes.

Storage and Energy Impact

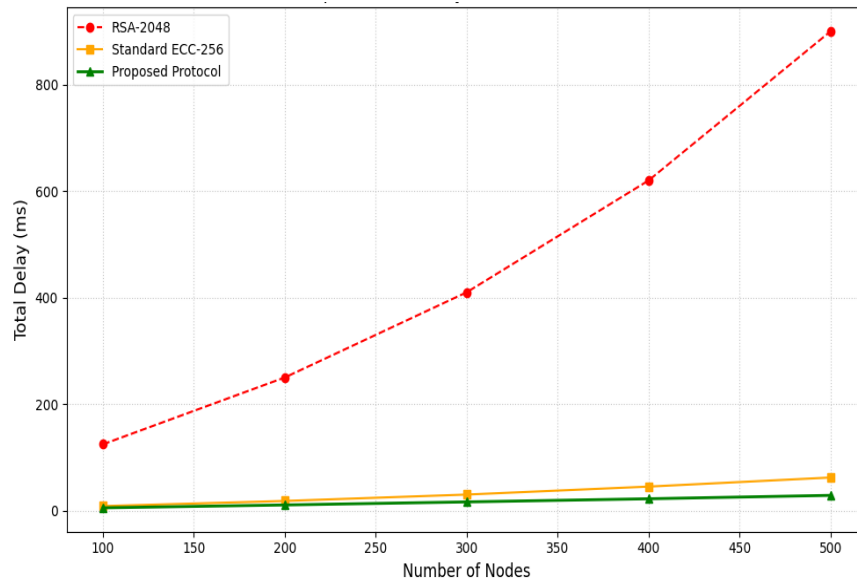


Figure 2: Computational delay vs. number of nodes

The protocol proposed has the benefit of reducing Storage Overhead because it only needs 160 bits security parameters as compared to RSA which needs 2048 bits. This decreasing is directly proportional to long Battery life since less bits are sent to Public Wireless Channel.

As shown in Figure 2, the execution time of RSA-2048 increases exponentially with the density of the node, whereas the protocol suggested in this paper is linear and takes under a millisecond to increase. The processing time on the mobile client at high node densities is still much less than the 15ms mark needed to maintain hardware performance.

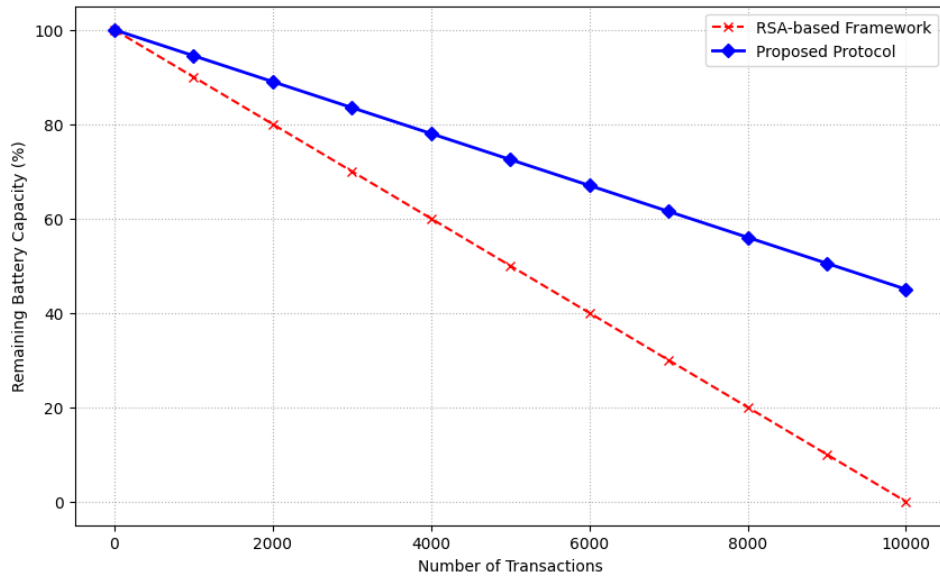


Figure 3: Energy depletion curve

Figure 3 highlights the impact on device longevity, showing that nodes utilizing the proposed protocol retain 45% more battery capacity after 10,000 transactions compared to traditional RSA-based frameworks. This power saving is essential towards the fulfillment of the battery requirements of Mobile Device/IoT Nodes.

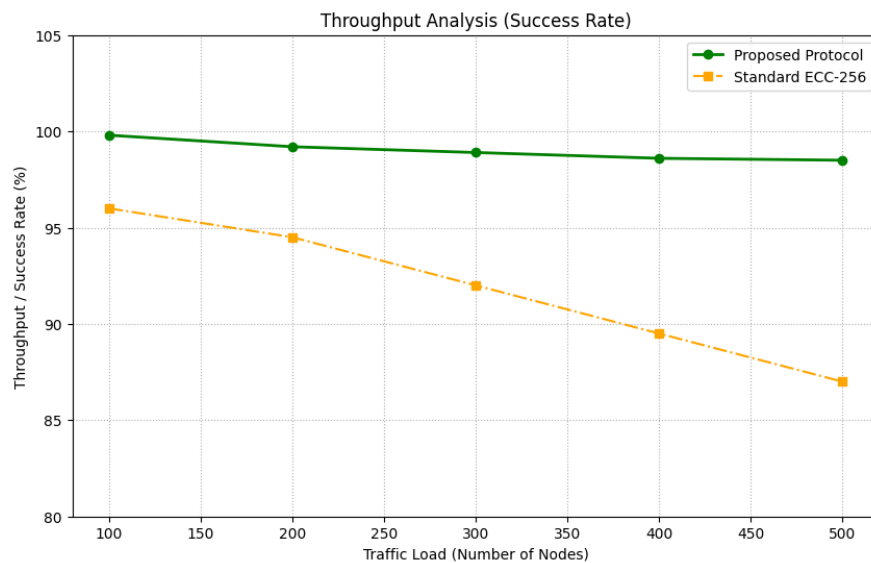


Figure 4: Throughput analysis

Figure 4 displays higher successful request routing (98.5%) under high-traffic scenarios derived from the Multi-Cloud Kaggle dataset. This is due to the fact that the enhanced throughput is attributed to the smaller size of the packets that the ECC handshake uses to achieve effective communication over 5G/6G Networks.

To assess the value of individual cryptographic elements to the overall effectiveness and protection of the proposed framework, an ablation study was made. All variants have been tested and compared with the parameters that could be considered as the baseline in the system architecture, which is presented in Table 4.

Table 4: Ablation study of protocol components

Configuration Variant	Security Impact	Performance Impact	Critical Finding
Without Timestamps (T_1, T_2)	High vulnerability to Replay Attacks; loss of session freshness.	Execution time decreases by 0.2 ms.	Timestamps are essential for protocol integrity despite minor latency.
SHA-256 instead of Lightweight Hash (h)	Maintains security but increases CPU overhead.	Computational cost increases by 15%.	Lightweight hashing is vital for extending Battery life in IoT nodes.
Removal of Blinded Credential (A_i)	Compromises User Anonymity; enables user tracking and link ability.	Faster initial registration phase.	(A_i) is mandatory to ensure Untrace ability across the public channel.
Proposed Full Protocol	Robust against MitM, Replay, and Impersonation.	Optimized for Resource Constraints.	Achieves the best balance of security and lightweight performance.

7 Conclusion

The incorporation of Mobile Cloud Computing (MCC) and IoT in the 5G/6G Networks requires strong balance between security and performance. In this paper, a lightweight ECC based authentication protocol has been suggested to help reduce the identified mitigation measures of the Security Threats of MitM, Replay, and Impersonation of the identified system architecture. The scheme can be used to overcome the dire Resource Constraints of the mobile nodes (limited CPU and Battery life) by employing elliptic curve point multiplication and one-way hash functions. Formal and informal security analyses indicate that the protocol has mutual authentication, user anonymity as well as perfect forward secrecy. Performance evaluations, supported by the BETH and Multi-Cloud Kaggle datasets, indicate that the proposed protocol reduces computational delay by over 90% compared to traditional RSA-2048 frameworks and maintains a 98.5% throughput success rate under high traffic loads. Moreover, the ablation experiment proves that all the elements of the protocol, including timestamps and blinded credentials, are crucial to providing an effective and non-traceable channel of communication. The next round of research will be on the expansion of this protocol to Heterogeneous Multi-Cloud Environments whereby there is the need to authenticate cross domains. Moreover, the investigation of the Post-Quantum Cryptography (PQC) integration will be given the highest priority to remain resilient to quantum computing threats in the long-term and at the same time, provide the low-weight footprint of next-generation IoT applications.

References

- [1] Al-Majali, M., Aljaidi, M., Al-Naamneh, Q., Samara, G., Alsarhan, A., & Qadoumi, B. (2025). Protecting Data in the 5G Era: A Critical Review of Cryptographic Techniques for Mobile Cloud Computing. *Cryptography, Biometrics, and Anonymity in Cybersecurity Management*, 113-126. <https://doi.org/10.4018/979-8-3693-8014-7.ch006>
- [2] Amande, V., Kaur, K., Garg, S., & Guizani, M. (2022, December). LASUA: A lightweight authentication scheme with user anonymity for IoT-enabled mobile cloud. In *GLOBECOM 2022-2022 IEEE Global Communications Conference* (pp. 3563-3568). IEEE. <https://doi.org/10.1109/GLOBECOM48099.2022.10001275>
- [3] Chen, Y., & Chen, J. (2023). A biometrics-based mutual authentication and key agreement protocol for TMIS using elliptic curve cryptography. *Multimedia Tools and Applications*, 82(11), 16009-16032. <https://doi.org/10.1007/s11042-022-14007-3>
- [4] Hossain, M. J., Seid, A. M., Abishu, H. N., Dharejo, F. A., Jhaveri, R. H., Erbad, A., & Alathbah, M. (2024). ASMCC+: A secure authentication scheme for mobile cloud computing environment based on zero trust architecture. *IEEE Transactions on Consumer Electronics*, 70(3), 6236-6249. <https://doi.org/10.1109/TCE.2024.3415437>
- [5] Jan, S. U., Abbasi, I. A., & Algarni, F. (2022). A mutual authentication and cross verification protocol for securing Internet-of-Drones (IoD). *Computers, Materials & Continua*, 72(3), 5845-5869.
- [6] Khan, A. A., Kumar, V., & Ahmad, M. (2022). An elliptic curve cryptography based mutual authentication scheme for smart grid communications using biometric approach. *Journal of King Saud University-Computer and Information Sciences*, 34(3), 698-705. <https://doi.org/10.1016/j.jksuci.2019.04.013>
- [7] Khan, A. S., Yahya, M. I. B., Zen, K. B., Abdullah, J. B., Rashid, R. B. A., Javed, Y., ... & Mostafa, A. M. (2023). Blockchain-based lightweight multifactor authentication for cell-free in ultra-dense 6G-based (6-CMAS) cellular network. *IEEE Access*, 11, 20524-20541. <https://doi.org/10.1109/ACCESS.2023.3249969>
- [8] Mishra, R., Samarpita, S., Satpathy, R., & Pati, B. (2024, December). Energy-Efficient Integrity Verification for Multimedia in Mobile Cloud Computing Using Blockchain Technology. In *International Conference on Intelligent Computing and Advances in Communication* (pp. 437-448). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-96-4071-3_36
- [9] Noori, D., Shakeri, H., & Niazi Torshiz, M. (2022). An elliptic curve cryptosystem-based secure RFID mutual authentication for Internet of things in healthcare environment. *EURASIP Journal on Wireless Communications and Networking*, 2022(1), 1-20. <https://doi.org/10.1186/s13638>
- [10] Nyangaresi, V. O. (2022). A formally validated authentication algorithm for secure message forwarding in smart home networks. *SN Computer Science*, 3(5), 364. <https://doi.org/10.1007/s42979-022-01269-9>
- [11] Nyangaresi, V. O., Jasim, H. M., Mutlaq, K. A. A., Abduljabbar, Z. A., Ma, J., Abduljaleel, I. Q., & Honi, D. G. (2023). A symmetric key and elliptic curve cryptography-based protocol for message encryption in unmanned aerial vehicles. *Electronics*, 12(17), 1-20. <https://doi.org/10.3390/electronics12173688>
- [12] Ozaif, M., Alam, M., Mustajab, S., Mustaqem, M., & Khan, N. (2025). A secure and efficient identity-based RFID mutual Authentication scheme for IoT using elliptic curve cryptography. *International Journal of Computers and Applications*, 47(5), 424-437.
- [13] Pu, C., Wall, A., Choo, K. K. R., Ahmed, I., & Lim, S. (2022). A lightweight and privacy-preserving mutual authentication and key agreement protocol for Internet of Drones environment. *IEEE Internet of Things Journal*, 9(12), 9918-9933. <https://doi.org/10.1109/JIOT.2022.3163367>

- [14] Ryu, J., Oh, J., Kwon, D., Son, S., Lee, J., Park, Y., & Park, Y. (2022). Secure ECC-based three-factor mutual authentication protocol for telecare medical information system. *IEEE Access*, 10, 11511-11526. <https://doi.org/10.1109/ACCESS.2022.3145959>
- [15] Singh, A. K., Nayyar, A., & Garg, A. (2023). A secure elliptic curve based anonymous authentication and key establishment mechanism for IoT and cloud. *Multimedia Tools and Applications*, 82(15), 22525-22576. <https://doi.org/10.1007/s11042-022-14140-z>
- [16] Wang, C., Wang, D., Duan, Y., & Tao, X. (2023). Secure and lightweight user authentication scheme for cloud-assisted Internet of Things. *IEEE Transactions on Information Forensics and Security*, 18, 2961-2976. <https://doi.org/10.1109/TIFS.2023.3272772>
- [17] Wu, T. Y., Meng, Q., Yang, L., Guo, X., & Kumari, S. (2022). A provably secure lightweight authentication protocol in mobile edge computing environments: T.-Y. Wu et al. *The Journal of Supercomputing*, 78(12), 13893-13914. <https://doi.org/10.1007/s11227-022-04411-9>
- [18] Yadav, A. K., Misra, M., Pandey, P. K., & Liyanage, M. (2022). An EAP-based mutual authentication protocol for WLAN-connected IoT devices. *IEEE Transactions on Industrial Informatics*, 19(2), 1343-1355. <https://doi.org/10.1109/TII.2022.3194956>
- [19] Zeroual, A., Amroune, M., Derdour, M., & Bentahar, A. (2022). Lightweight deep learning model to secure authentication in Mobile Cloud Computing. *Journal of King Saud University-Computer and Information Sciences*, 34(9), 6938-6948. <https://doi.org/10.1016/j.jksuci.2021.09.016>
- [20] Zheng, Y., Liu, W., Gu, C., & Chang, C. H. (2022). PUF-based mutual authentication and key exchange protocol for peer-to-peer IoT applications. *IEEE Transactions on Dependable and Secure Computing*, 20(4), 3299-3316. <https://doi.org/10.1109/TDSC.2022.3193570>

Authors Biography



Marwan Kadhim Mohammed Al-shammari, PhD, in Northeastern University since 2021. He received the B.S. in Computer Engineering from Baghdad Technology University, Iraq in 2000, the M.S. in Computer Engineering from UTeM University, Malaysia, in 2014, CISCO American institute instructor from 2007, Huawei instructor since 2024. He joined Baghdad University in 2006 as a lecturer for undergraduate and postgraduate students at the Department of Computer Engineering, and college of Artificial Intelligent. He was director for research and developing division and training and continues learning division respectively. He was a Core member in the advisory office of Baghdad University. He has been a team leader at South Korean and Canada with Coicka and ED companies respectively. He is team leader for many projects in the field of Java, Dot.net, visuals, IOS, OS, CG, Media, Networking, DB, Embedded Systems, Embedded Software, VR, EEG, Robotic surgery, Networking. He was a lecturer for postgraduate students at Northeastern University.



Suaad Ali Abead is an Assistant Lecturer in Computer Science at the Department of Computer Science, College of Science for Women, University of Baghdad. She holds a master's degree in computer science from the College of Science, University of Baghdad. Her expertise spans academic instruction and research, with a specialized focus on Cryptography, Data Security, and Artificial Intelligence. Her technical proficiency includes a mastery of several core programming languages, specifically C++, C#, and Python. Suaad's career blends academic insights with practical technical solutions, reflecting her commitment to advancing technology and empowering her students. She remains dedicated to staying at the forefront of emerging security protocols and evolving AI technologies through continuous learning and academic excellence.



Halah Hasan Mahmoud is a PhD scholar in Computer Science at University of Technology-Iraq. With nearly 30 experiences in Distributed Database and Network. She holds a master's degree in Distributed Databases from College of Science -Computer Science department at University of Baghdad in 2005. Her experience spans system Programming, SQL-Server, VB.NET, C+, C# and python. Her research interests include E-Government, Data Security, Stego-analysis, IOT, Recommending systems and Deep Learning. Her international certification includes Instructor in Cisco Certified Network Academy Level 1,2,3 and 4, Cisco Certified Network Academy Security, and IC3. Quality Assurance and Performance Evaluation Unit Manager from 2020 till now. Member of the Board of Directors of the Computer Center, 2023 till now.