The ISC Int'l Journal of Information Security

October 2025, Volume 17, Number 3 (pp. 1–12)

http://www.isecure-journal.org

Presented at the ISCISC'2025 in Tehran, Iran.

An Efficient ECC-Based Multi-Server Authentication Scheme for 5G Environment without Online Registration Server **

Seyede Marzieh Sadat Madani [1], Hamid Mala [1,*] and Mehrad Jaberi [1]

[1] *Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran.*

ARTICLE INFO.

Keywords:
Multi-Server Authentication, Key Agreement, 5G Networks, Security protocols, Elliptic Curve Cryptography.

Type:

doi:

Abstract

Multi-Server Authentication and Key Agreement (MAKA) protocols in 5G networks play a pivotal role in securing communications due to their widespread applications in domains such as drones, cellular networks, and secure communications. We propose a novel and efficient protocol for multi-server authentication and key agreement in 5G networks, based on Elliptic Curve Cryptography (ECC). The proposed protocol is secure against attacks such as user and server impersonation, password guessing, insider attacks, tracking, session key disclosure, replay, denial-of-service, and man-in-the-middle attacks. Additionally, distinctive features such as user anonymity, avoidance of bilinear pairing, key confirmation, perfect forward secrecy, and the ability to perform authentication without an online registration server make the proposed scheme more efficient and secure, compared to previous schemes. Formal analysis using Proverif cryptographic protocol verifier, confirms the protocol's confidentiality and authentication properties, while its computational and communication efficiency demonstrates relative superiority over comparable schemes.

1 Introduction

The fifth-generation (5G) mobile networks have brought significant advancements in wireless communications, enabling high-speed data transfer, ultra-low latency, and connectivity for a vast number of smart devices. Today, nearly two-thirds of the global population uses mobile devices, and the evolution from 2G through 4G to 5G aims to connect almost all smart devices, fostering ecosystems in areas such as security, artificial intelligence, big data, and pervasive wireless coverage [1].

According to the latest Ericsson Mobility Report (June 2024), global 5G subscriptions reached 1.6 billion by the end of 2023, marking a 71% increase from the previous year. Projections estimate this number will rise to 5.6 billion by 2029, emphasising the urgent need for scalable and flexible network architectures to handle this growth [2].

The 5G ecosystem consists of multiple entities, including base stations, core network infrastructures, and user equipment (UE). 5G base stations are smaller, more powerful, and denser than their 4G counterparts, connecting via standardized 5G

* Corresponding author.

**The ISCISC'2025 program committee effort is highly acknowledged for reviewing this paper.

Email addresses: m.sadatmadani@eng.ui.ac.ir, h.mala@eng.ui.ac.ir, mehrad.jaberi@eng.ui.ac.ir

ISSN: 2008-2045

protocols to ensure seamless communication.

Authentication and key agreement (AKA) protocols form the backbone of secure communication across mobile generations. Standards such as 5G-AKA, EAP-AKA, and EAP-TLS [8]—two symmetric key-based and one asymmetric key-based protocols—guarantee entity authentication and session key security [3].

The rapid expansion of 5G networks has led to the adoption of multi-server architectures to handle the increasing number of users and services efficiently. Multi-server authentication protocols address scalability and security challenges inherent in distributed 5G environments [4, 5].

In heterogeneous 5G networks, multi-server authentication is vital for ensuring seamless and secure access across multiple service providers, preventing unauthorized access, and mitigating various cyber-attacks [6].

To overcome the limitations of single-server architectures, 5G networks utilise multi-server designs where network functionalities are distributed among several servers. Users register once within the home network and can access multiple service providers without re-registering. This necessitates the development of multi-server authenticated key agreement (MAKA) protocols [7], which are critical in scenarios such as vehicular ad hoc networks (VANETs), UAV delivery systems, and cellular networks where security and uninterrupted access are paramount.

This paper proposes a novel Elliptic Curve Cryptography (ECC), ECC-based MAKA protocol designed for heterogeneous 5G environments. The main contributions include:

- Employing ECC and avoidance of bilinear pairing to provide lightweight multi-server authentication;
- Supporting user anonymity, key confirmation, perfect forward secrecy, offline registration server and resistance against user impersonation, server impersonation, password or Identifier guessing, insider, tracking, session key disclosure, replay, denial-of-service and man-in-the-middle attacks;
- Enabling single registration for access to multiple servers and offline capability of the home network server;
- Ensuring computational and communication efficiency suitable for 5G network environments.
- The security of the proposed scheme has been formally verified using the ProVerif tool.

The remainder of this paper is organised as follows. Section 2 reviews related authentication protocols in 5G environments and highlights their limitations. Section 3 describes the proposed protocol in detail, including system setup, user and service provider registration, the authentication and key agreement phase, and the service request subprotocol. Section 4 presents the security and performance evaluation of the protocol. Informal analysis, formal verification using ProVerif, and efficiency assessment are discussed in this section. Finally, Section 5 concludes the paper and suggests future research directions.

2 Related Work

The progression of authentication protocols in wireless communication, particularly within multi-server environments, has followed the evolution of mobile network architectures from 2G and 3G to the modern 5G landscape. While initial standards such as those from 3GPP [8] established basic protocols like 5G-AKA, EAP-AKA, and EAP-TLS to enable mutual authentication and key generation, these protocols primarily targeted single-server deployments. However, the introduction of distributed service providers and multi-server architectures in 5G networks necessitated significant protocol adaptation to ensure scalability, security, and efficiency in authentication processes.

The foundational research by Li *et al.* [9] introduced the concept of using neural networks for password-based authentication in multi-server systems. Their architecture, although innovative, proved vulnerable to offline password guessing and server spoofing due to single-factor reliance. Yang and Yang [10] later incorporated biometrics with smart cards to enhance authentication robustness. Nevertheless, their approach remained susceptible to key forgery attacks and demonstrated limited flexibility in biometric processing.

Further advances by Islam [11] employed pairing-based cryptography to facilitate mutual authentication between mobile devices and multi-server. While this method addressed some security limitations of prior schemes, it introduced performance bottlenecks arising from complex bilinear pairing operations and failed to protect user anonymity adequately due to identity exposure in plaintext during exchanges. Amin *et al.* [12] sought to alleviate scalability challenges by implementing a dual-registration server mechanism, enhancing the distribution of authentication tasks. However, as demonstrated in [7], their protocol fails to meet three critical security and efficiency requirements: perfect forward secrecy, biometric update capability, and compatibility with smart card memory constraints.

Efficiency improvements continued with Kumar *et al.* [13], who proposed a lightweight authentication mechanism using hash functions combined with biometric data. Although this strategy reduced computational demands, the persistent use of constant session identifiers compromised unlinkability and session privacy.

In parallel, research methodologies evolved from the proposal of novel schemes to the embedding of rigorous cryptanalysis as a design cornerstone. Wu *et al.* [14] employed identity-based encryption to support flexible, scalable authentication in 5G multi-server networks. However, Roy *et al.* [15] identified several practical vulnerabilities, such as susceptibility to desynchronization attacks and the absence of an effective password update mechanism.

To address computational constraints, Xiao and Gao [16] developed the 5GAKA-LCCO protocol, targeting reduced overhead in computation and communication. Their efforts prioritized operational efficiency, yet Modiri *et al.* [17] revealed significant deficiencies in privacy protections, especially with respect to safeguarding permanent user identifiers and concealed session information.

In recent years, rigorous adversarial analyses have underscored vulnerabilities in state-of-the-art MAKA protocols. Jaberi *et al.* [19] critically analyzed Wang *et al.* protocol [18], uncovering server spoofing risks resulting from inadequate inter-server authentication measures. Furthermore, their evaluation of Palit et al.'s protocol [4] revealed vulnerabilities to desynchronization and denial-of-service (DoS) attacks, which stemmed from ineffective session management and the omission of message freshness validation.

Across these developments, persistent threats such as user impersonation, session key disclosure, insider attacks, replay attacks, and traceability issues remain persistent challenges. The literature consistently highlights the difficulty in harmonising formal security assurances with practical deployment requirements, especially in energy-constrained 5G environments where computational efficiency and bandwidth conservation are critical.

The persistent cycle of protocol proposals, cryptanalytic critiques, and incremental improvements underscores the need for robust, scalable, and lightweight authentication mechanisms that not only withstand advanced network attacks but also align with the scalability and privacy demands of future 5G multi-server ecosystems.

In the current paper, propose a novel MAKA protocol tailored to address these multi-dimensional challenges, emphasising resistance against desynchronization, DoS, and impersonation attacks while supporting efficient resource utilization, privacy protection, and scalability across heterogeneous service providers.

3 Proposed Protocol

All the notations used throughout this paper are summarised in Table 1. The proposed protocol is designed for 5G networks and involves three main entities:

- **User Equipment** (UE_i): Devices used by the user, such as smartphones, tablets, or IoT devices, that access services provided by service networks in a 5G environment. This entity communicates with the service network for authentication and service access.
- **Home Network** (HN): The home network can be considered a mobile operator that maintains a database of its users and is responsible for authenticating them. It also serves as a trusted registration center, providing registration services for all users and service providers in the system.
- **Service Provider Network** (SN_j): Depending on the network infrastructure, SN_j can offer various services. If authentication is successful and SN_j can provide the requested services, it provides the service directly using the established session key. Otherwise, the authenticated user receives services from the home network through the service provider network.

We assume that the channel between the user (UE_i) and the service provider (SN_j) is not secure, while the channel between the service provider (SN_j) and the home network (HN) is assumed to be secure. The protocol consists of three phases: system setup, registration, and authentication and key agreement.

3.1 System Setup

The setup phase is performed by the HN to generate its private key and public parameters. The HN first selects an elliptic curve $E_p(a, b)$ over the finite field $\mathbb{F}_p$ with prime p and where the integer points on the curve form a cyclic additive group G with prime order q. It is assumed that P is a generator of the group G. Then, the HN randomly selects a value $s \in \mathbb{Z}_q^*$ as its private key. The HN's public key is computed as $P_{pub} = s \cdot P$. Subsequently, the HN selects two hash functions as follows.

$$h_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$$

$$h_2 : G \rightarrow \{0, 1\}^l$$

The value l is set to $2 \log_2(p + 128)$, assuming that the user and service provider identifiers are 128 bits. Finally, the HN securely stores the value

s and publicly releases the parameters $PP = \{E, G, P, p, q, P_{pub}, h_1, h_2\}$ in the system.

3.2 Registration Phase

Users and service providers register with the home network via a secure channel. The user registration process is shown in Figure 1, and the service provider registration process is shown in Figure 2.

Table 1. Notations Used in the current paper

Notation	Description
UE_i	The i-th user equipment.
HN	Home Network (Registration Center).
SN_j	The j-th service provider network.
E	Elliptic curve E defined over the finite field $\mathbb{F}_p$, where p is a large prime.
G	Cyclic additive group over the curve.
P, q	Generator and prime order of the group.
s	HN's private key.
P_{pub}	HN's public key.
h_1, h_2	Collision-resistant hash function.
$\{E, G, P, p, q, P_{pub}, h_1, h_2\}$	Public parameters published in the system.
ID_{UE_i}	Unique user identifier.
ID_{SN_j}	Unique service provider identifier.
d_{UE_i}, X_{UE_i}	User's private and public keys.
d_{SN_j}, X_{SN_j}	Service provider's private and public keys.
$\oplus$	XOR operation.
$\|\|$	Concatenation operation.
Δ	Requested services.
∇	Provided services.

*Note: In the proposed protocol, the superscript * on variables indicates that it is not yet confirmed whether the computed or received value matches the correct value.*

3.2.1 User Registration

(1) The user UE_i sets its identifier ID_{UE_i} and sends it to the HN via a secure channel.

(2) The HN receives and stores ID_{UE_i}, then computes the following values:

$$x_i \in_R \mathbb{Z}_q^*$$

$$X_{UE_i} = x_i \cdot P$$

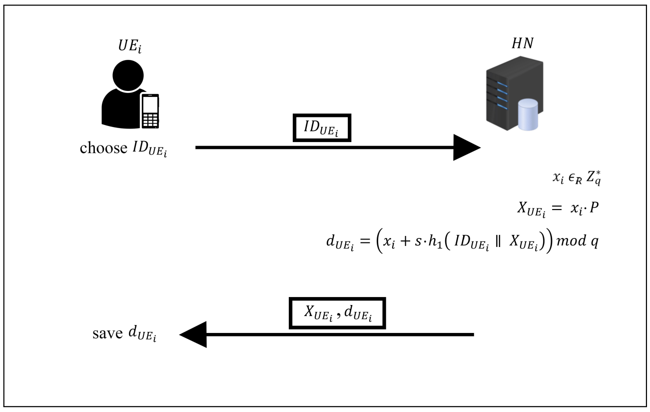

Figure 1. User registration phase of the proposed protocol

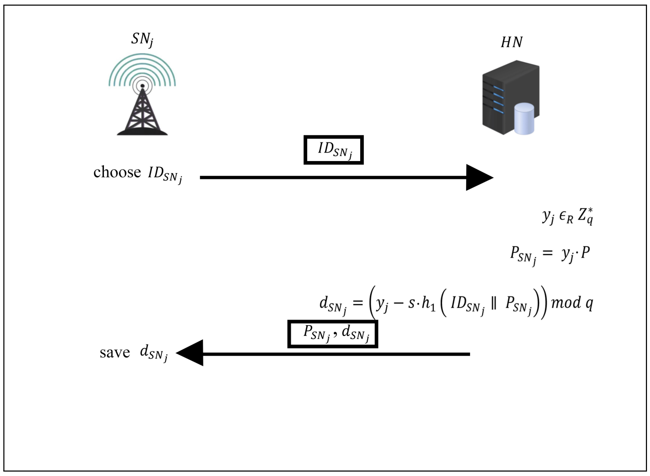

Figure 2. Service provider registration phase of the proposed protocol

$$d_{UE_i} = (x_i + s \cdot h_1(ID_{UE_i} \| X_{UE_i})) \mod q$$

(3) HN sends the user's private and public keys, d_{UE_i} and X_{UE_i}, to UE_i via a secure channel.

(4) The user UE_i receives its public and private keys from the Home Network and securely stores its private key, d_{UE_i}.

3.2.2 Service Provider Registration

Each service provider must register with the Home Network. This allows users to access various services from both SN_j and HN without needing to register separately with each server.

(1) The service provider SN_j creates its unique identifier ID_{SN_j} and sends it to the Home Network.

(2) The Home Network receives and stores the identifier, then computes the following values:

$$y_j \in_R \mathbb{Z}_q^*$$

$$P_{SN_j} = y_j \cdot P$$

$$d_{SN_j} = (y_j - s \cdot h_1(ID_{SN_j} \| P_{SN_j})) \mod q$$

(3) The Home Network sends the private and public keys, d_{SN_j} and P_{SN_j}, to SN_j via a secure channel.
(4) The service provider SN_j receives its public and private keys from the Home Network, securely stores its private key d_{SN_j}, and publishes its public key and identifier.

3.3 Authentication and Key Agreement Phase

This phase, illustrated in Figure 3, enables the user UE_i to authenticate with the service provider SN_j and establish a shared session key over an insecure channel.

After registration, the user equipment UE_i attempts to access a service provider, such as SN_j. It must follow the steps outlined in Figure 3 to achieve authentication and session key agreement with SN_j. In the proposed protocol, the superscript * on variables indicates that it is not yet confirmed whether the computed or received value matches the correct value.

(1) The user UE_i selects a random value $r_i \in \mathbb{Z}_q^*$ and computes the following values.

$$r_i \in_R \mathbb{Z}_q^*$$
$$R_i = r_i \cdot P$$
$$K = r_i \cdot \left(P_{SN_j} - h_1(ID_{SN_j} \| P_{SN_j}) \cdot P_{pub}\right)$$
$$v_i = h_2(K) \oplus (ID_{UE_i} \| X_{UE_i})$$
$$\Phi_i = r_i^{-1} \cdot (h_1(R_i \| K \| v_i \| T_i) + d_{UE_i}) \mod q$$

The value P_{SN_j} is published by the Home Network during the registration phase. Additionally, the value P_{pub} is publicly available to the user. Furthermore, when the user is in a specific region, it performs a handshake with the selected SN_j and gains access to ID_{SN_j} (the ID_{SN_j} values are also published by the Home Network during registration). Using the above, the user computes v_i. T_i is a timestamp, and Φ_i is a value that incorporates the private key of UE_i.
(2) The user UE_i sends the message $M_1 = \{\Phi_i, T_i, R_i, v_i\}$ to SN_j.
(3) The service provider SN_j first verifies the freshness of the received timestamp T_i. If T_i is not fresh, it terminates the current session. Otherwise, SN_j proceeds with the next steps.
(4) The service provider SN_j verifies the received values. To do so, it first computes K by multiplying its private key d_{SN_j} by the received R_i. It then verifies the correctness of the received values as follows. It XORs the received v_i with the hash of the computed K^*. This yields $ID^*_{UE_i} \| X^*_{UE_i}$. It also computes $\Phi_i \cdot R_i$. If Equation 1 holds, SN_j proceeds with the next steps; otherwise, it rejects the request from UE_i:

$$K^* = d_{SN_j} \cdot R_i$$
$$(ID^*_{UE_i} \| X^*_{UE_i}) = v_i \oplus h_2(K^*)$$
$$\Phi_i \cdot R_i \stackrel{?}{=} h_1(R_i \| K^* \| v_i \| T_i) \cdot P + h_1(ID^*_{UE_i} \| X^*_{UE_i}) \cdot P_{pub} + X^*_{UE_i} \quad (1)$$

(5) If the above steps are successfully completed, SN_j randomly selects $r_j \in \mathbb{Z}_q^*$ and generates the following values. At this stage, SN_j establishes the session key with UE_i using the following values:

$$r_j \in_R \mathbb{Z}_q^*$$
$$R_j = r_j \cdot P$$
$$R = r_j \cdot R_i$$
$$SK_{ji} = h_1(ID^*_{UE_i} \| ID_{SN_j} \| R \| K^*)$$
$$MAC_1 = h_1(ID^*_{UE_i} \| ID_{SN_j} \| R \| K^* \| SK_{ji})$$

(6) SN_j sends the message $M_2 = \{R_j, MAC_1\}$ to UE_i.
(7) The user UE_i uses R_j to compute R. Using R, it can compute the session key and MAC_1. If the computed MAC_1^* matches the received MAC_1, UE_i completes the key agreement and authentication with SN_j. Otherwise, it may request another service or terminate the connection:

$$R = r_i \cdot R_j$$
$$SK_{ij} = h_1(ID_{UE_i} \| ID_{SN_j} \| R \| K)$$
$$MAC_1^* = h_1(ID_{UE_i} \| ID_{SN_j} \| R \| K \| SK_{ij})$$

From this point, the user equipment and the service provider can use the session key to encrypt subsequent messages. Additionally, the session key can be used to compute MAC_1 to verify the integrity of the parameters SK_{ij} and R.

The session key is used to encrypt subsequent communications. The protocol supports three service delivery modes, illustrated in Figure 4, Figure 5, and Figure 6, individually.

- **Mode 1**: SN_j directly provides services using the session key (Figure 4).
- **Mode 2**: If SN_j cannot provide the services, it forwards the request to HN, and HN provides the services through SN_j (Figure 5).
- **Mode 3**: The user directly requests services from HN without SN_j accessing the service details (Figure 6).

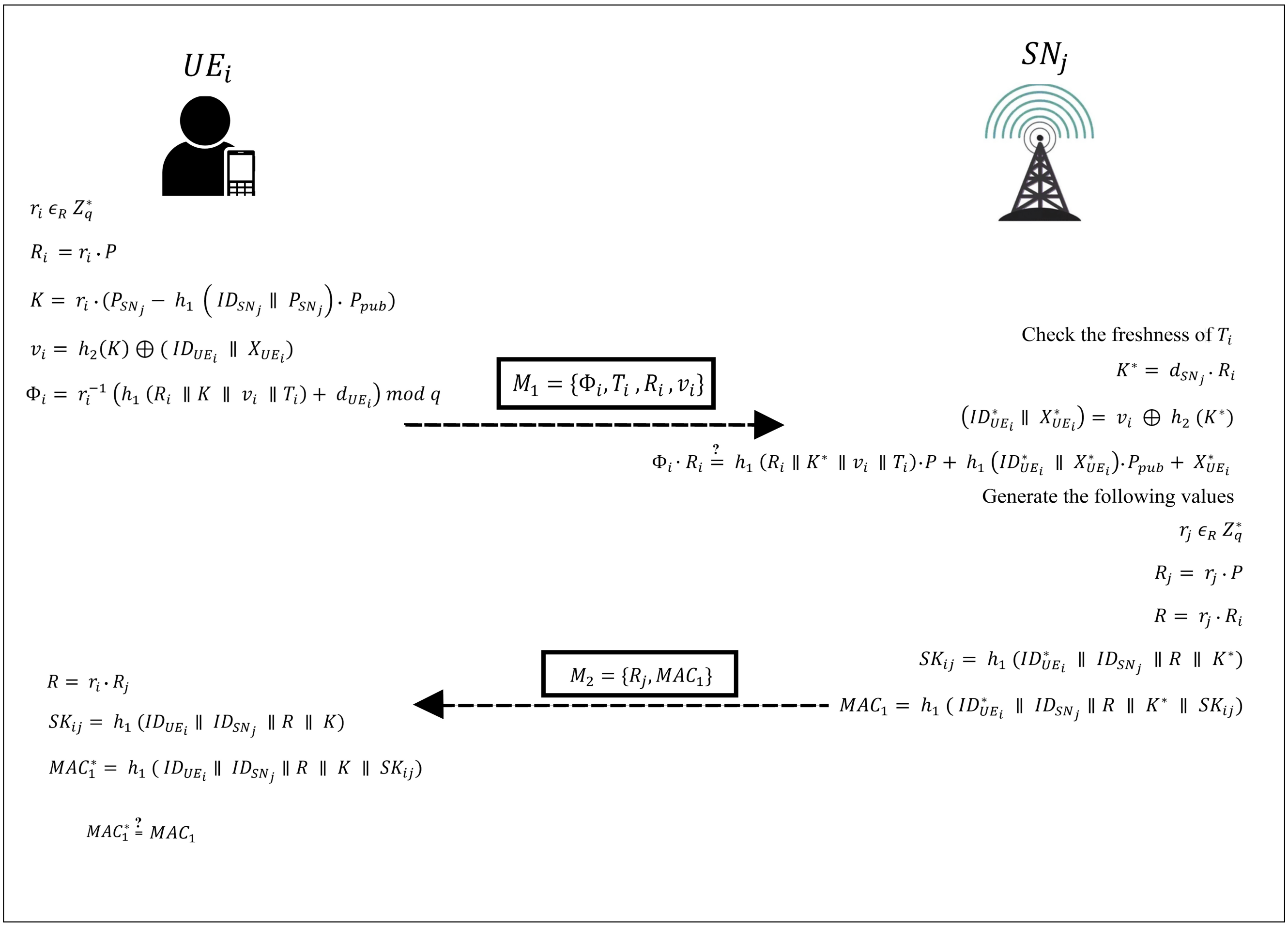

Figure 3. Authentication and Key Agreement Phase of the propsed protocol

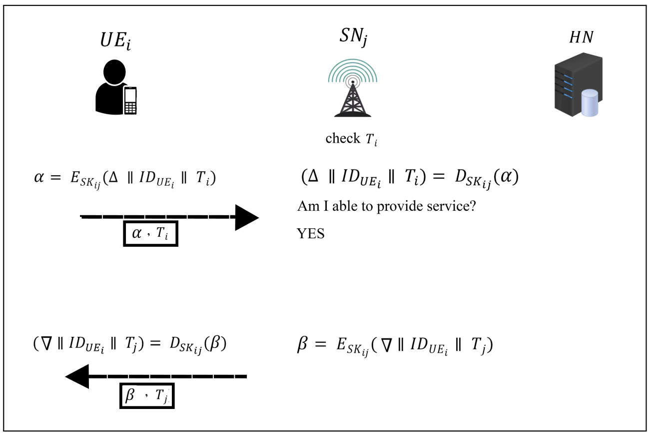

Figure 4. Service Delivery Mode 1

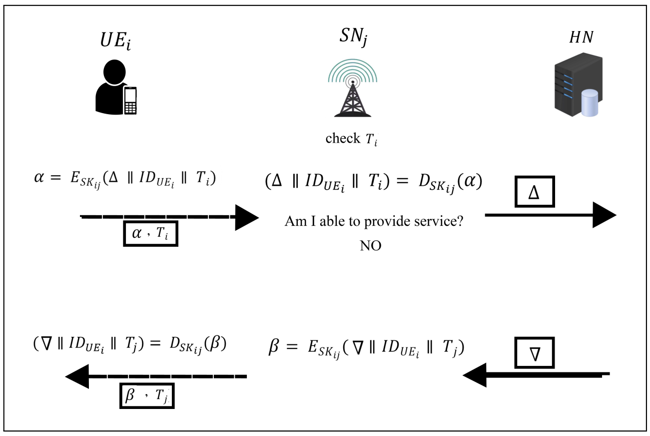

Figure 5. Service Delivery Mode 2

3.4 Service Request Subprotocol

When services are requested from HN through SN_j, the protocol ensures that SN_j remains unaware of the service details. This process is illustrated in Figure 6, which details the message exchange for service requests. The user encrypts the service list with the session key, and HN securely provides the services.

4 Security Evaluation

The proposed protocol has been evaluated against various attacks and security features. It is assumed that the communication channel between UE_i and SN_j is insecure, while the channel between SN_j and HN is secure. Furthermore, SN_j is considered semi-honest, and HN is fully trusted.

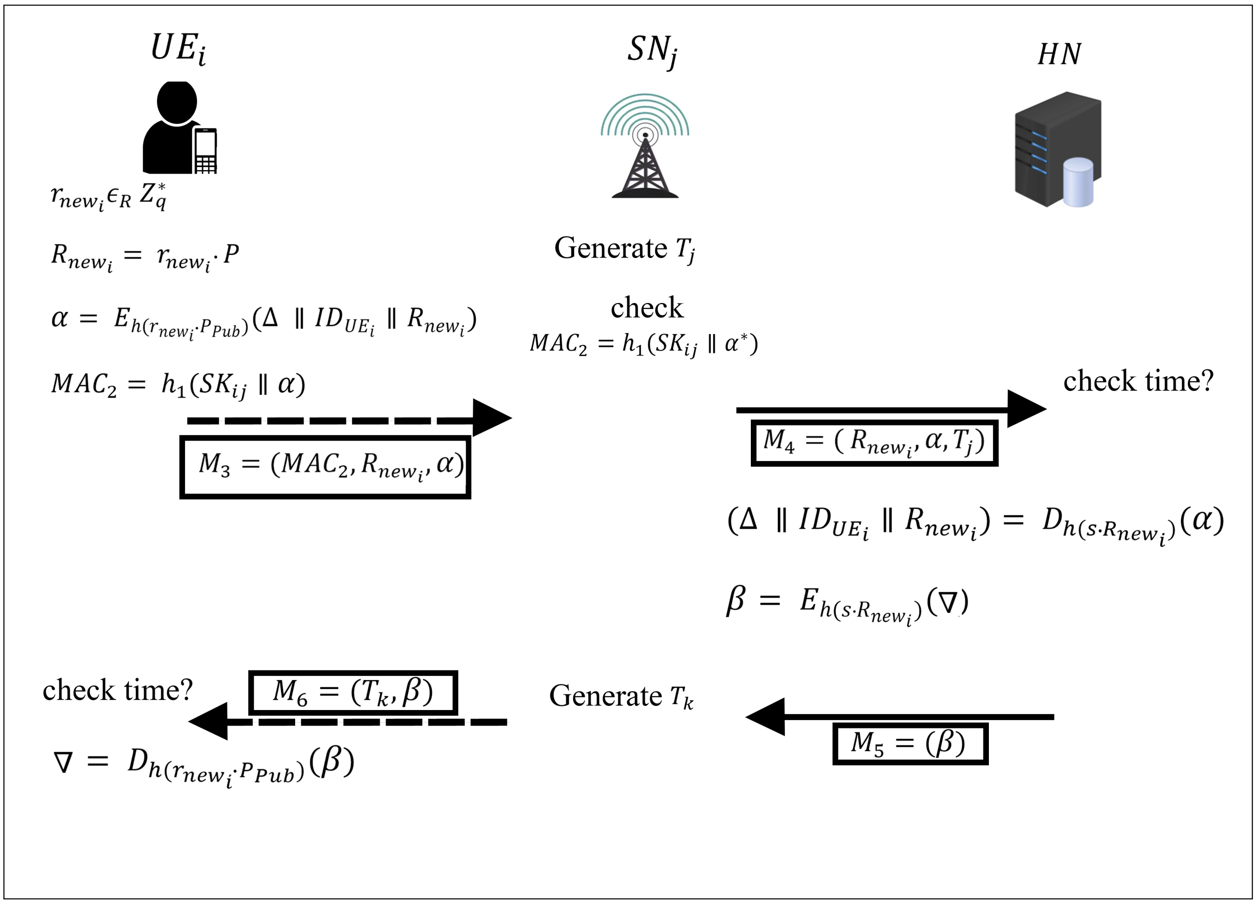

Figure 6. Service Delivery Mode 3

4.1 Informal Security Analysis

The proposed protocol is resistant to the following attacks and provides several security and performance features:

- **User Impersonation Attack**
  If an attacker attempts to impersonate a user, she/he must pose as a legitimate user and deceive the service provider. Suppose that the attacker decides to impersonate user UE_i. In this case, the attacker must correctly generate four values $\{\Phi_i, T_i, R_i, v_i\}$. The timestamp T_i is a value the attacker can create. If the attacker obtains the user's private key, d_{UE_i}, she/he can compute Φ_i and the remaining values (e.g., the service provider's public key). Given that the Home Network is trusted and offline during the authentication and key agreement process, and the user's private key is always kept hidden from other entities, it is challenging for the attacker to obtain d_{UE_i}. During registration, the Home Network generates and provides the user's private and public keys, and the attacker cannot obtain s, the Home Network's private key.
- **Server Impersonation Attack**

If an attacker attempts to impersonate a service provider, she/he must pose as the server. In this case, must first obtain the correct value of K as follows:

$$K = r_i \cdot \left(P_{SN_j} - h_1(ID_{SN_j} \| P_{SN_j}) \cdot P_{pub}\right) = d_{SN_j} \cdot R_i$$

There are three possible ways an attacker might impersonate SN_j to create a valid message M_2:

(1) The attacker obtains the private key d_{SN_j}. In the proposed protocol, private keys are protected from the attacker and other entities. Thus, the attacker cannot obtain the service provider's private key.

(2) The attacker obtains r_i, the user's random value: r_i is temporarily stored in the local database during the session and is never transmitted over an insecure channel; thus, the attacker cannot compute K this way. Additionally, since the attacker does not have r_{new_i}, cannot perform actions to access services.

(3) The attacker, correctly generates the message $M_2 = \{R_j, MAC_1\}$. The probability of this is negligible, as the attacker must si-

multaneously obtain the correct r_j to produce M_2 without knowing d_{SN_j} and r_i.

Therefore, the proposed protocol is resistant to server impersonation attacks.

- **Password or Identifier Guessing Attack** The protocol is resistant to guessing the user's real identifier, as even if the attacker correctly guesses ID_{UE_i}, cannot construct the message M_1 without r_i. Thus, cannot generate $\{\Phi_i, K, R_i, v_i\}$ to perform a user impersonation attack. Additionally, since the attacker lacks r_i, they cannot take further actions.
- **Insider Attack** Suppose the attacker has access to the service provider's private key, d_{SN_j}. In this case, the attacker cannot impersonate the user. In the proposed protocol, after receiving the message $M_1 = \{\Phi_i, T_i, R_i, v_i\}$, the service provider, having d_{SN_j}, computes K^* and derives $ID_{UE_i} \| X_{UE_i}$, but fails to verify the correctness of $\Phi_i \cdot R_i$, as this equality holds only if the attacker own the user's private key, which is not feasible. Thus, the proposed protocol is resistant to insider attacks.
- **Tracking Attack** In the proposed protocol, during authentication, the user UE_i sends four values $\{\Phi_i, T_i, R_i, v_i\}$ to SN_j over an insecure channel. The values R_i and T_i reveal no user information. The user's identifier is concealed in v_i using K. In constructing K, r_i is a random value that also makes v_i random. Similarly, Φ_i is randomised via K (due to r_i). Additionally, the attacker cannot find a link between v_i and Φ_i through analysis, as there is no linear relationship between their structures. In the second message, two values $\{R_j, MAC_1\}$ are sent to the user. By intercepting these messages, the attacker cannot obtain user information, as R_j is a random value chosen by SN_j and unrelated to the user's identity. The MAC_1 value is a hash of two independent and random values K and R_j, which cannot reveal user information. Thus, the presence of a single user cannot be detected across two sessions, as all values change in subsequent sessions. Consequently, the proposed protocol supports unlinkability and is resistant to tracking attacks.
- **Session Key Disclosure Attack** In this attack, the attacker tries to obtain the session key. Assume the attacker intercepts all values on the insecure channel, thus possessing $\{\Phi_i, T_i, R_i, v_i, R_j, MAC_1\}$. To construct the session key $SK_{ij} = h_1(ID_{UE_i} \| ID_{SN_j} \| R \| K)$, the attacker should obtain four values ID_{UE_i}, ID_{SN_j}, R and K. ID_{SN_j} is distributed by SN_j. Even with R_i and R_j, the attacker cannot derive the correct R, as r_i and r_j are random and unobtainable during the protocol. The user's real identity is also concealed during authentication, and thus, the attacker cannot access K, as r_i is hidden. Additionally, without access to the private key of SN_j, K cannot be computed. Therefore, the proposed protocol is resistant to session key disclosure attacks.
- **Replay Attack** In the proposed protocol, the message M_1 includes a timestamp T_i, which is encrypted using Φ_i, incorporating the user's private key. Thus, the attacker cannot use previously intercepted messages from sessions to deceive the service provider. Even if an attacker attempts to replay an old valid tuple (Φ_i, R_i, v_i) with a fresh timestamp T_i', the verification will fail because T_i is also incorporated into the hash computation h_1, ensuring that the verification equation cannot be satisfied with an altered timestamp. As long as the R_i generated by the user differs in each communication round, the attacker cannot use intercepted messages to gain the user's trust. Therefore, the proposed protocol is resistant to replay attacks. The protocol relies on timestamps for freshness to prevent replay attacks, assuming reasonable clock synchronization among network entities, which is a standard requirement in 5G networks as per 3GPP standards [8]. For enhanced robustness in future implementations, addressing potential clock skews in distributed environments could be explored.
- **Denial-of-Service Attack** A denial-of-service attack aims to temporarily or permanently disrupt service (here, the execution of the MAKA protocol). In MAKA protocols, an attacker may cause message desynchronization between entities, preventing services from HN or SN_j from reaching the user. In the proposed protocol, if an attacker sends incorrect messages, the service provider detects miscellaneous messages by verifying the correctness of $\Phi_i \cdot R_i$, as the attacker cannot correctly construct this value unless user impersonation occurs, which was addressed above. Since values such as MAC_1 or timestamps must be verified in each case, the attacker cannot desynchronize messages. Thus, the protocol is resistant to denial-of-service attacks. Additionally, if many users access the system, the service provider, based on the values it must compute, can distinguish between malicious and legitimate users. Legitimate users each have a unique identifier and corresponding public key.

- **Man-in-the-Middle Attack**
  In a successful man-in-the-middle attack, the attacker can intercept messages between the user and the service provider and alter them without the entities noticing any changes or defects. In the proposed protocol, upon receiving values from the user, SN_j computes $K^* = d_{SN_j} \cdot R_i$ and $v_i \oplus h_2(K^*)$. It also performs $Phi_i \cdot R_i$ Φ_i. Consequently, SN_j detects any tampering or message alteration, as it will not obtain the correct user identifier and public key. Additionally, SN_j computes R and MAC_1 and sends them to the user. The user must first derive the correct R and then compute MAC_1 with its own values; if the received MAC_1 differs from the computed MAC_1, the user detects message tampering. Thus, the proposed protocol is resistant to man-in-the-middle attacks.
- **User Anonymity**
  In the proposed protocol, the user's real identity is concealed by hashing K. The value K is defined as follows:
  $$K = r_i \cdot \left(P_{SN_j} - h_1(ID_{SN_j} \| P_{SN_j}) \cdot P_{pub}\right)$$
  $$= d_{SN_j} \cdot R_i$$
  Since the private key of SN_j, d_{SN_j}, and r_i are kept hidden from other users and service providers, the attacker cannot compute K without access to r_i and d_{SN_j}. Therefore, the proposed protocol supports user anonymity against other users.
- **Avoidance of Bilinear Pairing**
  The use of bilinear pairing increases computational overhead. The proposed protocol employs asymmetric key cryptography based on elliptic curves, resulting in reduced computational overhead compared to protocols using bilinear pairing.
- **Key Confirmation**
  In the proposed protocol, it must be ensured that the user and the service provider obtain the same session key. After receiving M_1, SN_j first verifies the user's identity and then constructs the session key as $SK_{ji} = h_1(ID^*_{UE_i} \| ID_{SN_j} \| R \| K^*)$. On the other hand, only an authenticated user can derive the correct R. Thus, it can construct $SK_{ij} = h_1(ID_{UE_i} \| ID_{SN_j} \| R \| K)$. Additionally, SN_j computes MAC_1 and sends it to the user. The user compares the received value with its own computed MAC_1. Therefore, the protocol provides key confirmation.
- **Perfect Forward Secrecy**
  If an attacker obtains the private key of the user d_{UE_i} or service provider d_{SN_j} after passively intercepting all previous messages over the insecure channel, they can compute $K = d_{SN_j} \cdot R_i$. However, without knowledge of the correct ephemeral secret r_i or r_j, the attacker cannot compute the shared secret $R = r_j \cdot R_i = r_i \cdot R_j$, due to the hardness of the Elliptic Curve Diffie-Hellman (ECDH) problem. As a result, the session key remains secure and cannot be reconstructed, even if long-term private keys are compromised after the session. Therefore, the proposed protocol achieves perfect forward secrecy.
- **Offline Registration Server**
  Mutual authentication and key agreement between the user and the service provider do not rely on the Home Network. Therefore, the proposed protocol ensures protocol execution without requiring the Home Network (registration server) to be online. Note that if the service provider cannot provide the requested service, the Home Network delivers the service, but this does not create a single point of failure in the protocol.

4.2 Formal Security Analysis

ProVerif is an automated tool for analyzing the security properties of cryptographic protocols using formal methods. Using the Proverif tool, the protocol's confidentiality and authentication properties were verified. Queries were defined for the values ID_{UE_i}, d_{UE_i}, d_{SN_j}, r_i, r_j, K, K^*, SK_{ij}, and SK_{ji} to verify their confidentiality. Additionally, events were defined for authenticating messages exchanged between entities. The results, shown in Figure 7, confirm that all defined values remain secure. Furthermore, all messages exchanged between entities are authenticated.

4.3 Performance Evaluation

In this section, the performance of the proposed protocol is compared with the schemes of Wu et al. [14], Xiao *et al.* [16], the 5G-AKA standard [8], and Palit *et al.* [4] in terms of computational and communication costs.

The computational costs of all protocols, summarised in Table 2, are expressed parametrically using TH, TS, and TP, representing the time taken by hash operations, symmetric cryptographic operations, and asymmetric cryptographic operations, respectively. According to [4], these values were obtained using a Zolertia remote sensor device equipped with an ARM Cortex-M3 microcontroller operating at 32 MHz, with 32 KB RAM and 512 KB flash memory. Based on this setup, TH, TS, and TP were set to 0.03 ms, 0.12 ms, and 342.39 ms, respectively. It is important to note that the computational costs rep-

```
Verification summary:

Query not attacker(IDUEi[]) is true.

Query not attacker(dUEi[]) is true.

Query not attacker(dSNj[]) is true.

Query not attacker(ri[]) is true.

Query not attacker(rj[]) is true.

Query not attacker(K[]) is true.

Query not attacker(K'[]) is true.

Query not attacker(SKij[]) is true.

Query not attacker(SKji[]) is true.

Query attacker(SKij[]) ==> event(start_UE(dUEi[])) && event(end_UE(dUEi[])) is true.

Query attacker(SKji[]) ==> event(start_SN(IDSNj[],dSNj[])) && event(end_SN(IDSNj[],dSNj[])) is true.

Query inj-event(end_UE(x)) ==> inj-event(start_UE(x)) && inj-event(start_SN(y,z)) && inj-event(end_SN(y,z)) is true.
```

Figure 7. The results of the formal security verification of the propsed protocol using Proverif tool

resent the total time consumed by all cryptographic operations performed by all entities involved in the protocol within a single session, not just the user's operations.

Scheme	Computational Cost	Time (ms)
Proposed Protocol	$11TH + 4TS$	0.81
[14]	$42TH + 2TS$	1.56
[16]	$4TH + 2TS + 2TP$	685.14
[8]	$20TH + 2TS + 2TP$	685.62
[4]	$16TH + 2TS$	0.72

Table 2. Comparison of Computational Costs

The communication cost is evaluated based on the number of exchanged messages and the total number of bits transmitted per session between protocol participants. The bit counts are calculated based on the following assumptions: hash functions are 256 bits, random values are 64 bits, entity identifiers are 64 bits, timestamp fields are 16 bits, and exchanged text messages are 64 bits. Table 3 presents a detailed comparison of communication costs, including the number of bits for each message and the total bits transmitted. The proposed protocol demonstrates the lowest communication overhead, requiring only two messages and 912 bits in total.

To further evaluate the performance, we compare the security features of the proposed protocol with the referenced schemes based on a vulnerability analysis. Table 4 highlights key security attributes, indicating their presence (✓) or absence (×) in each scheme. The proposed protocol outperforms others in providing comprehensive security against common attacks while maintaining efficiency.

Security Feature	Proposed	[14]	[16]	[8]	[4]
User Impersonation Resistance	✓	×	✓	×	✓
Server Impersonation Resistance	✓	×	✓	✓	×
Password/Identity Guessing Resistance	✓	×	✓	×	✓
Insider Attack Resistance	✓	✓	×	×	✓
Tracking Resistance (Unlinkability)	✓	×	×	×	✓
Session Key Disclosure Resistance	✓	✓	✓	×	×
Replay Attack Resistance	✓	✓	✓	✓	×
Denial-of-Service (DoS) Resistance	✓	×	✓	✓	×
Man-in-the-Middle (MiTM) Resistance	✓	✓	×	✓	✓
Perfect Forward Secrecy (PFS)	✓	×	✓	×	✓
User Anonymity	✓	×	×	×	✓
No Bilinear Pairing	✓	✓	×	N/A	✓
Key Confirmation	✓	×	✓	×	✓
Offline Registration Server	✓	✓	✓	N/A	×

Table 4. Comparison of Security Features (Based on Vulnerability Analysis)

Regarding storage costs, the proposed protocol requires approximately 320 bits of storage per user entity for keys and parameters (e.g., identifiers and hashes), which is relatively low compared to similar schemes that often exceed 512 bits due to larger key sets or additional parameters. Note that not all parameters need permanent storage, as some are ephemeral during sessions.

The schemes of Wu et al. [14] and Xiao *et al.* [16] are unsuitable for 5G environments due to their reliance on an online registration server, incompatibility with distributed 5G systems, and excessive communication overhead (5776 bits and 3248 bits, respectively). The 5G-AKA standard [8] also suffers from security weaknesses and high computational and

Scheme	Messages (Bits)	Total Bits
Proposed Protocol	M1: 592, M2: 320	912
[14]	M1: 848, M2: 1696, M3: 2128, M4: 1104	5776
[16]	M1: 80, M2: 256, M3: 400, M4: 400, M5: 960, M6: 704, M7: 448	3248
[8]	M1: 256, M2: 320, M3: 320, M4: 1152, M5: 576, M6: 448, M7: 384, M8: 384	3840
[4]	M1: 640, M2: 768, M3: 832, M4: 512, M5: 256	3008

Table 3. Comparison of Communication Costs

communication costs (3840 bits) due to intensive use of asymmetric cryptography. Although the scheme of Palit *et al.* [4] shows competitive computational performance and moderate communication overhead (3008 bits), it has been demonstrated to be vulnerable to denial-of-service attacks [19]. In contrast, the proposed protocol achieves key agreement and mutual authentication using the minimum number of message exchanges (2 messages, 912 bits) and acceptable computational costs, while providing superior security features as shown in Table 4, making it highly suitable for secure deployment in 5G environments.

5 Conclusion

The proposed MAKA protocol for 5G networks addresses the security and performance limitations of existing schemes. It is shown that the proposed protocol is resistant to a wide range of attacks and provides key security features. Formal verification with ProVerif and performance comparisons demonstrate its robustness and efficiency, making it suitable for diverse 5G applications. For future research, it is recommended to use various security evaluation tools and well-equipped 5G test laboratories to further verify the security of current MAKA protocols in 5G environments. While the proposed protocol relies on Elliptic Curve Cryptography (ECC), which is vulnerable to quantum computing attacks such as Shor's algorithm, designing post-quantum MAKA protocols resistant to quantum computing threats is proposed as a critical direction for future work. While the proposed protocol demonstrates computational and communication efficiency suitable for 5G environments through theoretical analysis and comparisons, real-world scalability testing with a large number of users and service providers remains an open challenge. Due to resource constraints in this study, we did not conduct simulations or experiments on platforms like NS-3 or real 5G testbeds. As a direction for future work, we recommend evaluating the protocol's performance in simulated large-scale 5G networks using tools such as OMNeT++ or MATLAB to assess latency, throughput, and resource utilization under high-load scenarios involving thousands of users and multiple service providers. Future research should also explore efficient key update and revocation strategies to enhance the protocol's adaptability in large-scale multi-server setups.

References

[1] Y. Liu, L. Huo, and G. Zhou, "TR-AKA: A two-phased, registered authentication and key agreement protocol for 5G mobile networks," *IET Information Security*, vol. 16, no. 3, pp. 193–207, 2022.

[2] Ericsson, "Ericsson Mobility Report," June 2024. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/mobility-report

[3] Y. Xiao and S. Gao, "5GAKA-LCCO: a secure 5G authentication and key agreement protocol with less communication and computation overhead," *Information*, vol. 13, no. 5, p. 257, 2022.

[4] S. K. Palit, M. Chakraborty, and S. Chakraborty, "Performance analysis of 5GMAKA: A lightweight mutual authentication and key agreement scheme for 5G network," *The Journal of Supercomputing*, vol. 79, no. 4, pp. 3902–3935, 2023.

[5] J. Zhang, Y. Li, and X. Wang, "Efficient multi-server authentication protocols for 5G-enabled IoT networks," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 2, pp. 1234–1245, 2023.

[6] Y. Liu, M. Zhao, and X. Li, "Resilient MAKA protocol design for 5G multi-server architecture," *Journal of Network and Computer Applications*, vol. 215, p. 103350, 2023.

[7] I. ul Haq, J. Wang, Y. Zhu, and S. Maqbool, "A survey of authenticated key agreement protocols for multi-server architecture," *Journal of Information Security and Applications*, vol. 55, p. 102639, 2020.

[8] 3GPP TS 33.501, "Security architecture and procedures for 5G system," 2019.

[9] L.-H. Li, L.-C. Lin, and M.-S. Hwang, "A remote password authentication scheme for multi-server architecture using neural networks," *IEEE*

Transactions on Neural Networks, vol. 12, no. 6, pp. 1498–1504, 2001.

[10] D. Yang and B. Yang, “A biometric password-based multi-server authentication scheme with smart card,” in *2010 International Conference on Computer Design and Applications*, vol. 5, pp. V5-554–V5-558, IEEE, 2010.

[11] S. K. H. Islam, “A provably secure ID-based mutual authentication and key agreement scheme for mobile multi-server environment without ESL attack,” *Wireless Personal Communications*, vol. 79, no. 3, pp. 1975–1991, 2014.

[12] R. Amin, S. K. H. Islam, M. S. Obaidat, G. P. Biswas, and K.-F. Hsiao, “An anonymous and robust multi-server authentication protocol using multiple registration servers,” *International Journal of Communication Systems*, vol. 30, no. 18, e3457, 2017.

[13] A. Kumar and H. Om, “An improved and secure multiserver authentication scheme based on biometrics and smartcard,” *Digital Communications and Networks*, vol. 4, no. 1, pp. 27–38, 2018.

[14] T.-Y. Wu, Z. Lee, M. S. Obaidat, S. Kumari, S. Kumar, and C.-M. Chen, “An authenticated key agreement protocol for multi-server architecture in 5G networks,” *IEEE Access*, vol. 8, pp. 28096–28108, 2020.

[15] P. K. Roy and A. Bhattacharya, “A lightweight multi-server authentication and key agreement protocol based on group key (MAKA) for multi-server environment,” *The Journal of Supercomputing*, pp. 1–28, 2022.

[16] Y. Xiao and S. Gao, “5GAKA-LCCO: A secure 5G authentication and key agreement protocol with low computational and communication overhead,” *Information*, vol. 13, no. 5, p. 257, 2022.

[17] M. M. Modiri, M. Salmasizadeh, J. Mohajeri, and B. H. Khalaj, “Two protocols for improving security during the authentication and key agreement procedure in the 3GPP networks,” *Computer Communications*, vol. 215, pp. 11–23, 2023.

[18] W. Wang, H. Huang, F. Xiao, Q. Li, and L. Xue, “An adaptive secure handover authenticated key agreement for multi-server architecture communication applications,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 9830–9839, 2022.

[19] M. Jaberi, H. Mala, and S. M. S. Madani, “Cryptanalysis of two authenticated key agreement protocols in multi-server environments,” *ISeCure: The ISC International Journal of Information Security*, vol. 17, no. 2, pp. 179–187, 2025.

Seyede Marzieh Sadat Madani received her B.Sc. and M.Sc. degrees in computer engineering and information security from the University of Isfahan (UI) in 2020 and 2024, respectively. Her research interests include Security Protocols, Network Security, and Post-Quantum Cryptography.

Hamid Mala received the B.Sc, M.Sc, and Ph.D degrees in Electrical Engineering from Isfahan University of Technology (IUT), in 2003, 2006, and 2011, respectively. He joined the Department of Information Technology Engineering, University of Isfahan (UI), in 2011. He is currently with the Faculty of Computer Engineering, UI, as an associate professor. Hamid is an Executive Board member of the Iranian Society for Cryptology since 2022. His research interests include Design and Cryptanalysis of Block Ciphers, Cryptographic Protocols, and Post-Quantum Cryptography.

Mehrad Jaberi received his B.Sc. in Information Technology Engineering from Isfahan University of Technology (IUT) in 2014. He also received his M.Sc. degree in Information Security from the University of Isfahan (UI) in 2017. Under supervision of Dr.Hamid Mala, Mehrad received his Ph.D degree in Secure Computing from University of Isfahan (UI) in 2025. Mehrad's research interests include Secure Multiparty Computation, Network Security, Cryptographic Protocols and Blockchain.