IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 28 June 2022, accepted 25 July 2022, date of publication 1 August 2022, date of current version 4 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3195304

RESEARCH ARTICLE

Group Security Authentication and Key Agreement Protocol Built by Elliptic Curve Diffie Hellman Key Exchange for LTE Military Grade Communication

KARIM H. MOUSSA[1], AHMED H. EL-SAKKA[2], SHAWKY SHAABAN[2], AND HASSAN NADIR KHEIRALLAH[2]

[1]School of Internet of Things, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China
[2]Electronics and Communications Department, Faculty of Engineering, Alexandria University, Alexandria 21544, Egypt

Corresponding author: Ahmed H. El-Sakka (eng-ahmed.elsakka@alexu.edu.eg)

ABSTRACT 4G mobile communication is a global technology. Therefore, it is essential to enforce confidentiality between mobile users and their networks. This paper presents a Group Security Authentication and Key Agreement Protocol Built by Elliptic Curve Diffie Hellman Key Exchange (GSAKA-ECDHKE) to overcome and address the LTE networks Evolved Packet System Authentication and Key Agreement Protocol (EPS-AKA) protocol flaws and vulnerabilities. GSAKA-ECDHKE is presented for 4G mobile military group communications to provide security, confidentiality, and privacy while the users and networks authenticate. By embedding the Group Commander (GC) role in the EPS-AKA protocol to control the member authentication in the group. GSAKA-ECDHKE protocol is based on Elliptic Curve Diffie Hellman Key Exchange (ECDHKE) and hash function to generate and share secret Elliptic Curve (EC) key to encrypt and protect the routing authentication parameters. The Automated Validation of Internet Security Protocols and Applications (AVISPA) tool is used for security analysis and formal verification. AVISPA demonstrated that GSAKA-ECDHKE had overcome various known security attacks such as Man In The Middle (MITM), replay attacks, and Denial of Services (DoS) attacks, satisfying the evaluated security requirements. Additionally, the suggested protocol provides the lowest communication overheads compared to the existing group-based AKA protocols.

INDEX TERMS Authentication, group, 4, military, evolved packet system, LTE, AKA protocol, AVISPA.

I. INTRODUCTION

Mobile technology has been the most significant global event during the past thirty years, dominating all parts of life [1]. Therefore, it has become vital for all individuals and businesses to save time and effort in fulfilling their duties. Each generation of mobile technology has added an appointed feature to its previous generation, for example, increasing the number of users, raising the data rates, and preserving privacy for each user [2]. Security is the base requirement for protecting the subscribers' privacy of any mobile telecommunications system. User privacy provides the network's users with convenient service connectivity, preventing the network from being exploited and securing their information. Authentication of the users for network access and ensuring a bidirectional trust between users and their network are key elements of building such secured systems. Both secure connectivity and user authentication are related to the authentication and access control mechanisms that provide secure network services for all the network subscribers [3].

Security among networks and users became a significant issue for any wireless network to save the daily subscribers' private data, financial transactions, and personal conversations. The authentication of users and networks in 4G mobile is the first step toward establishing security trust. In the presence of different attack types such as Man In The

The associate editor coordinating the review of this manuscript and approving it for publication was Usama Mir.

Middle (MITM) [4], replay attacks, and Denial of Services (DoS) attacks [5], the Third Generation Partnership Project (3GPP) established the Evolved Packet System Authentication and Key Agreement Protocol (EPS-AKA) authentication protocol to serve as the authentication protocol for Long Term Evolution (LTE) networks that were presented for the 4G mobile communication, which demands adaptive bandwidth, low contact latency, greater data rates, and increased capacity and coverage of the networks [6].

To gain access and use the benefits of the network, the user verifies his identity using the EPS-AKA authentication mechanism. By attaching the user's authentication request, which includes pre-authentication information, to the Mobility Management Entity (MME), the MME temporarily transfers the user's identity information to the Home Subscriber Server (HSS), which is responsible for generating the authentication vector and returning it to the MME [7].

Recently, the 4G LTE network was applied to military communication systems providing higher spectral efficiency than the previous mobile generations [1], [6]. Nevertheless, the EPS-AKA mechanism cannot ideally provide full security for military life. Because the EPS-AKA protocol exposes numerous vulnerabilities and threats, the network's and mobile users' privacy and confidentiality may be affected [8]. For example, provide the UE's permanent identification IMSI to the serving MME in plaintext in the attached request message without first proving the MME's integrity. As a result, an attacker who intercepts and reads this communication may quickly identify this UE and violate his privacy. Also, the attacker can obtain the SNID, which contains the Public Land Mobile Network ID, which is a grouping of the MCC and the MNC (MCC+MNC) according to 3GPP guidelines because the serving MME delivers its identification to the HSS in plaintext in the authentication information request message. Consequently, the attacker can easily intercept and read this message, identify this MME, and impersonate the serving network.

Typically, wired links are transmitted in plaintext, allowing an attacker to capture the authentication vectors provided with the serving MME and obtain the value of session keys. As a result, the attacker can compromise the confidentiality of UE's conversations. Furthermore, the transmission between EPC network entities is vulnerable to attack [9]. As a result, an attacker can obtain the shared information and disrupt network privacy and secrecy. According to 3GPP standards, the UE authenticates the HSS with the value AUTN, and the serving MME authenticates the UE with the value RES, whereas the UE does not authenticate the MME and the MME does not authenticate the HSS. Therefore, attackers can reroute traffic from a legitimate network to a bogus network [10].

Any group authentication protocol's primary aims are to ensure secrecy and security among communicative entities, especially for crucial group military communications [11]. Several group AKA methods have been presented to achieve the aims of the mutual AKA between the group members and their network, whether used for group mobile authentication or machine type devices (MTDs) authentication. Privacy preservation and network overhead reduction are the most important aims for this type of communication [12]. This section presents a summary of the available group-AKA protocols. Chen *et al.* presented the G-AKA protocol as the first group AKA protocol[13]. The MME uses previously authenticated members' information to authenticate the remaining members in this protocol. As a result, the AKA procedure for the remaining devices in the group may be streamlined. However, when many members need to access the network simultaneously, the protocol causes signaling congestion [14]. It is also vulnerable to different security threats such as DoS and MITM. Lai *et al.* [15] offered SE-AKA for 3GPP networks, while Jiang *et al.* [16] presented EG-AKA for non-3GPP networks to increase the security of G-AKA. These protocols resist the attacks; however, they have a large computation expense due to asymmetric key operations [17].

The symmetric key-based NOVEL-AKA protocol is suggested by Lai *et al.* to decrease computation overhead [18]. However, it has additional issues, such as network signaling congestion, subject to DoS and redirection assaults. Furthermore, Choi *et al.* [19] introduced the GROUP-AKA protocol, which minimizes the signaling congestion in many members who need to access the network simultaneously while retaining the group key's unlinkability. However, the protocol has privacy preservation problems and is vulnerable to DoS attacks.

Cao *et al.* enhanced the group-based AKA protocols security by presenting a group signature-based GBAAM-AKA protocol [20]. Nevertheless, the system incurs high computation costs and does not guarantee privacy due to asymmetric key operations. Fu *et al.* introduced a PRIVACY-AKA protocol, which protects privacy through asymmetric cryptography [21]. While the protocol is resistant to known assaults, it is computationally expensive and lacks essential forward and backward secrecy.

Lai *et al.* developed the lightweight GLARM-AKA protocol [22] to minimize communication overheads. While the protocol benefits users with limited resources, it does not guarantee group key unlinkability, leaving users' identities vulnerable to impersonation attacks. While the protocol is advantageous for users with little resources, it fails to secure group key unlinkability and is thus vulnerable to identity capturing and impersonation attacks. Li *et al.* [23] developed the GR-AKA protocol to maintain security and privacy by reserving members' IDs via sophisticated and time-consuming Lagrange Component (LC) calculations. Yao *et al.* suggested the group-based secure GBS-AKA protocol to survive assaults and minimize communication overhead [24]. However, it does not safeguard member privacy and is susceptible to impersonation and denial-of-service assaults. Additionally, it violates the group key's unlinkability.

They were considering the security and non-security issues. Parne *et al.* suggested that the security improved group-based SEGB-AKA protocol to increase security [25].

The protocol protects members' privacy and defeats the most known assaults. It keeps the unlinkability in the group key, and anytime a member joins or leaves the group, the group's key will be modified. Furthermore, its computation and transmission overheads are manageable. However, the protocol is vulnerable to a single DoS attack and, contrary to its promises, fails to address the most important problem in communication networks.

The proposed Group Security Authentication and Key Agreement Protocol Built by Elliptic Curve Diffie Hellman Key Exchange (GSAKA-ECDHKE) is presented for military group communication using the Elliptic Curve Diffie Hellman Key Exchange (ECDHKE) [26]. To overcome the weaknesses and threats of the LTE standard authentication protocol EPS-AKA and protects the MUEs privacies in the group.

Earlier group-AKA protocols employed lengthy parameters due to accumulating and concatenating the identifying information of all group members to calculate the authentication parameters. In addition, the time required to recalculate and update the group ID and the secret keys when a member leaves or joins the group is a high burden. These last notes contribute to the increased computational time required to execute the authentication process.

Our proposed protocol has been successful in decreasing the length of the used variables and neglecting the required time to recalculate and update the group ID and the secret keys of the group when a member leaves or joins the group by authenticating each group member individually with the Group Commander (GC) when the Military User Equipment (MUEi) needs to access the network. That leads to reducing network congestion. The GC is a device with special capabilities for military aspects to guarantee the security requirements for MUEi. The GC's role is to transfer and receive the effective variables used in calculating and generating the authentication routing parameters. By controlling the mutual authentication between the protocol entities MME, HSS, and MUEs.

During protocol mechanism execution, the proposed protocol exploits hash functions and an Elliptic Curve (EC) secret key to authenticate entities with considerable security shape mutually. Also, in the proposed GSAKA-ECDHKE, the authentication mechanism will stop if the GC becomes unavailable. Additionally, the proposed protocol achieves the lowest network overheads compared to other protocols. Finally, the proposed protocol addresses the single key critical issue that earlier group-based AKA protocols could not address. The abbreviations for the symbols and lengths used in the procedure are listed in Table.1.

The residuals of this article are arranged as follows: In sections 2 and 3, a brief background on EPS-AKA and ECDHKE is presented. The proposed authentication Protocol is presented in section 4, along with its initializations and a demonstration of the protocol mechanism. Section 5 demonstrates the proposed protocol verification using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool for security verification analysis, investigating the proposed protocol under different types of attacks (DoS, MIMT, Reply, and impersonate MME attacks) and the performance analysis by calculating the protocol's communication overheads and computational complexity. Finally, the conclusion of this work is summarized in section 6.

TABLE 1. Symbols' abbreviations and lengths.

symbols abbreviations	Description	Lengths (bit)
AUTNgc	Authentication GC	64
AUTNhss	Authentication HSS	64
AUTNi	Authentication MUEi	64
BLA	Base Location Area	40
F	authentication BLA	64
GID	Group ID	128
IMSIi	International Mobile Subscriber Identifier	128
K	LTE Key	128
Kecc	Elliptic Curve Key	192
KSI	Key Set Identifier	3
MIDgc	Military Identifierer GC	128
MUENCi	Military User equipment Network Capability	4
NT	Network Type	16
RANDi	Random Number	128
RES	Response	64
RNgc	Random Prime Number GC	192
RNi	Random Prime Number MUEi	192
SNID	Serving Network Identifier	40
XRES	Expected Response	64

II. EPS-AKA PROTOCOL BACKGROUND

The EPS-AKA protocol is the 4G mobile communication network's authentication standard version. The main three sections are the User Equipment (UE), which saves the LTE key K and International Mobile Subscriber Identity (IMSI) as the factory default, the MME, and the HSS, which store the IMSI and LTE key K in the authentication center (AC) and generate a random number (RANDi) and rising sequence number (SQN). As a result, its procedure is as follows:

1) The UE sends an authentication request to MME, and MME responds by sending the UE a request for authentication information. UE responds to MME by transmitting his IMSI, User Network Capability (UNC), which informs MME of the security algorithm available in UE, and Key Set Identifier (KSI) = 7, indicating that UE lacks an authentication vector.
2) MME sends an authentication information request to HSS that includes the UE identification, the serving network identity (SNID), the mobile country code (MCC) and mobile network code (MNC), the network type (NT), and the number of Authentication Vectors (AV) that MME requires.
3) When HSS receives an authentication information request from MME, it utilizes a crypto function to determine the Expected Response (XRES), the Authentication Token (AUTN), the Cipher Key (CK), and the Integrity Key (IK). These technologies are supported,

including LTE K, SQN, and RANDi. As a result, the access network's key derivation function calculates the Access Security Management Entity (KASME) key using the CK, IK, SQN, and SN ID. After that, HSS sends AVi to MME, which contains RANDi, AUTNi, and XRESi.

4) MME obtained AVi from HSS and transferred only RANDi and AUTNi to UE, leaving XRESi in the Evolved Packet Core (EPC) to securely store these two values.
5) The UE then uses RANDi to compute its Response (RES) and AUTNu using the same crypto function HSS used to drive AUTNi and XRESi. Following that, UE compares its calculated AUTNu to the received AUTNi to authenticate HSS if AUTNu = AUTNi, then sends its driven RES to MME, which compares it to the stored XRESi to authenticate UE if RES = XRESi.

III. ELLIPTIC CURVE DIFFIE-HELLMAN

The idea of ECDHKE is to use the Elliptic Curve Cryptography (ECC) to generate a public key (asymmetric key) for two entities that have their private keys and use the Diffie-Hellman key exchange (DHKE) to exchange these two public keys. Each of these entities has an EC public and private key, and the public key will be equal [27]. Using ECC, exchanging these two public keys produces a secret key between the communicative entities.

$$Y_{\text{MUEi public}} = \left(X_{\text{MUEi private}} \times BP\right) \bmod P \quad (1)$$

$$Y_{\text{GC public}} = \left(X_{\text{GC Private}} \times BP\right) \bmod P \quad (2)$$

Furthermore, the shared secret (SEC) key will be as follow

$$Y_{\text{MUEi public}} \times X_{\text{GC Private}} = Y_{\text{GC public}} \times X_{\text{MUEi private}} = \text{SEC} \quad (3)$$

This SEC is the same as one way hash function, and BP is the ECC base point, P is chosen prime number for ECC, $Y_{\text{MUEi puplic}}$ and $X_{\text{MUEi private}}$ are the public and private keys of MUE, respectively, $Y_{\text{GC puplic}}$ and $X_{\text{GC private}}$ are the public and private key of GC. In this proposed protocol, we use ECDHKE, which can be described as follows the MUE chooses a random prime number RNi ($X_{\text{MUEi private}}$), which is the private key of MUE from P prime order of EC generator matrix G and generates its public key using the ECC which is {RNi}Kecc ($Y_{\text{MUEi puplic}}$). Also, the GC chooses a random prime number RNgc ($X_{\text{GC private}}$), which is the private key of GC from n prime order of EC generator matrix G, and generates its public key using the ECC, which is {RNgc}Kecc ($Y_{\text{GC puplic}}$). By exchanging these two public keys using DHKE, each entity has its private key and the public key of the other entity. Each of the MUE and the GC calculate the shared EC secret key SEC (RNi, {RNgc}Kecc), which is equal to SEC (RNgc, {RNi}Kecc as in (3) [28].

IV. THE PROPOSED GSAKA-ECDHKE

The proposed GSAKA-ECDHKE protocol is presented for constructing 4G secure military group communication. The proposed GSAKA-ECDHKE introduces solutions to overcome the standard EPS-AKA flaws and threats in LTE networks by modifying the standard EPS-AKA. The GC is the new entity that has been added to the EPS-AKA construction, and it has high communication capability, storage capacity, and a backup battery. The GC does not know the IMSIs of the group members. Also, each member does not know the GC IMSIgc and its Military ID (MIDgc). The role of GC is to transfer, receive, and control data to and from each MUEi of the group members and control the MUEi authentication. Therefore, if the GC fails during the authentication process, the group's network and MUEs will find out, and the authentication procedure will be terminated.

A. INITIALIZATION AND PREPARATION

In 4G/LTE networks, each MUE has a pre-shared key (K) which is factory default and stored on its universal subscriber identity module/universal integrated circuit card and in the AC in HSS. This key is used in some Key Derivation Functions (KDF) to find the values of the EPS-AKA authentication parameters.

The SQN used in the standard EPS-AKA is dispensed to decrease the length of the authentication parameters due to preregistering all the group members in HSS with the GC, which led to minimizing the bandwidth. Each MUEi of the group has its own identity (IMSIi), which is known and registered in HSS. Also, the GC has its own identities, IMSIgc and MIDgc, which HSS knows. The HSS establishes a Group Information List (GIL) to handle and register the MUEs information. By using the grouping algorithm, the group is constructed based on specific rules of military-grade hierarchy in which the group members belong to the same tasks or jobs, within the same region, or have similar behaviors. The GIL includes all data concerning the group MUEs' and GC identities. HSS provides the group ID (GID) as a hash function between MIDgc and LTE K of the group members, the factory default. The GC device can send the IMSIi related to any MUEi of the group members instead of its IMSIgc to the MME.

B. GROUP INITIALIZATION

HSS, constructing a framework for MUEs, and GC as shown in Fig.1. Constructing the group in the proposed protocol mainly depends on the military-grade hierarchy, which lists the MUEs. The GIL in HSS forms a group of the MUEs involved in the authentication process with the same local communication region (a place where the group members work together). So, the mobility of the group members is restricted, especially during the authentication process. Once the MUE is authenticated, it can move around. All the group members and the GC information are preregistered in GIL as a group. The authentication process starts immediately when group members attach an authentication request to the MME, and group membership can be changed when a new MUEi is added or removed by GIL. Only HSS has the authority to add or remove MUEi without any effect on the

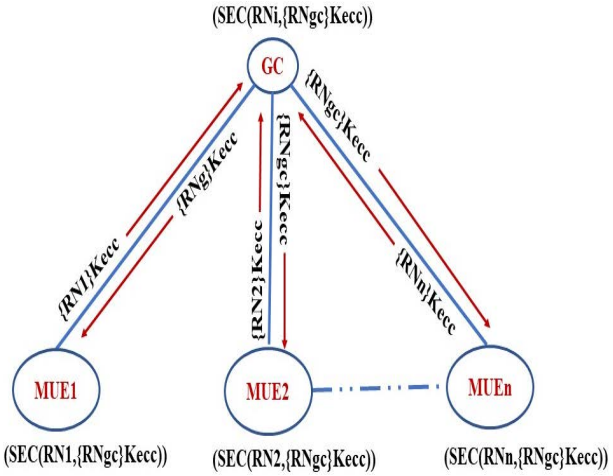


FIGURE 1. Framework construction of MUEs, and GC.

routing authentication parameters and the generated secret keys, which are calculated individually by each group user.

Therefore, updating and recalculating the routing authentication parameters is not required whenever a user is added or removed from a group, in contrast to the majority of related works, which rely on aggregating and integrating the information of all group members in order to establish secret keys and generate a group ID using the agreed encryption algorithm and hash function.

C. CALCULATE AND SHARE THE SECRET EC KEY

The authentication process starts when the MUEi asks to join the network. The MUEi sends its public key to the GC, which calculates it by selecting RNi, the random prime number for MUEi, and uses ECC algorism to calculate it by:

$$MUEi_{puplic\ key} = (RNi \times BP)\ mod\ P. \tag{4}$$

Once the GC of the group received the MUEi public key, immediately send his public key to the MUEi:

$$GC_{puplic\ key} = (RNgc \times BP)\ mod\ P. \tag{5}$$

It was computed by selecting RNgc, the random prime number for GC, and applying the ECC algorithm to it. Consequently, after the MUEi and the GC have exchanged their public key together, each entity of MUEi and the GC calculates the secret key SEC_{key} That will be used later to encrypt and protect the authentication parameters.

$$SEC_{key} = MUEi_{puplic\ key} \times RNgc = GC_{puplic\ key} \times RNi \tag{6}$$

The generated SEC_{key} has a feature like the hash function that is one-way encryption and will use to protect the HSS provided parameters RANDi, MIDgc that concerned MUEi and the group (step 7).

D. SESSION KEY AGREEMENT

The proposed GSAKA protocol conveys the authentication parameters using two secret session keys. The first one is

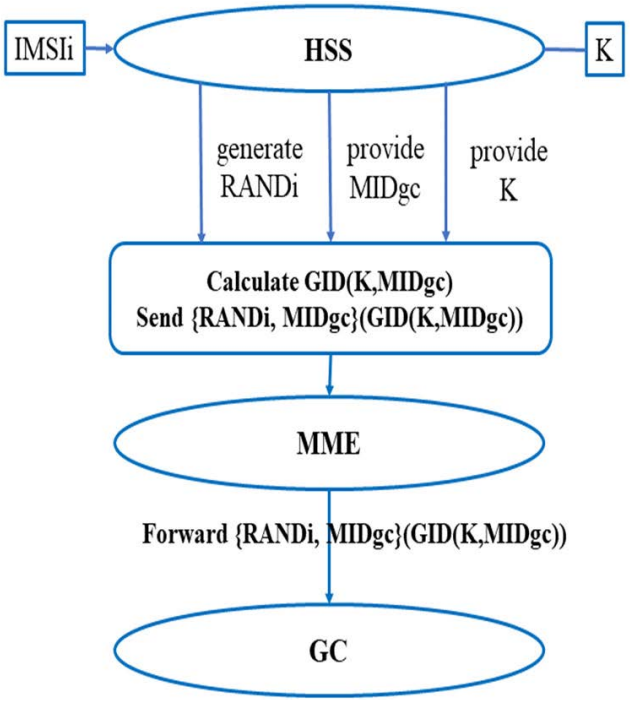


FIGURE 2. Secret session keys between the HSS and MME and between the MME and GC.

(GID (K, MIDgc)) produced by the HSS and shared between HSS, MME, and between the MME, GC as shown in Fig. 2 to encrypt the given HSS data about the GC (MIDgc) and the MUEi (RANDi) that is used to compute the remainder of the authentication hash function. Once the MME and the GC have received the created HSS message, they have accepted the session key generated by the HSS. As a result, the GC transmits the authentication parameters to the MUEi device to proceed to the next stage using the established session key that was initially generated by the MUEi and the GC of the group (SEC (RNi, {RNg}Kecc)).

The agreement on the session key between the HSS, MME, and GC, as well as the session key between the GC and the MUEi device, provides mutual authentication among the proposed protocol entities and encrypts the traffic moving from and to the core network to protect the distributed information provided by the HSS.

E. THE PROPOSED PROTOCOL MECHANISM

The steps of the GSAKA-ECDHKE protocol will start once the MUEi attaches his pre-authentication request to MME. Immediately, the MME requests the MUEi for his authentication information. Then the protocol will be as follow.

1) Step 1: MUE –> GL: {IMSIi, K, RNi} Kecc
   After MME requests for MUEi identification, the MUEi sends his IMSIi, K after encrypted by ECC key kecc and its public key that is generated by selecting a random private prime number (RNi), which is the private EC key and used to generate its public EC key {RNi}kecc to the GC to produce the shared secret key between them.

2) Step 2: GC –> MUE: {RNg}Kecc
Once the GC receives the MUEi identifiers IMSIi and the public key {RNi}kecc, it sends its' public EC key {RNgc}kecc that is generated by its selection to a random private prime number (RNgc) to use later in the protocol mechanism to establish the MUEi-GC secret EC key.
3) step 3: GC –> MME: F (IMSIi, K), MUENCi, KSI
In the time which the GC send their public key {RNgc}kecc to the MUEi, it also sends the MUEi identity IMSIi hashed with its pre-shared LTE K to MME to cover the value of MUEi identities away from any attacker F (IMSIi, K) because the hash function cannot retrieve and send the Military User Equipment Network Capability (MUENCi) which is the available algorithms In MUEi for security combining with KSI which is set to equal seven as in the original EPS-AKA which specify that the MUEi has no authentication key.
4) step 4: MME –> HSS: F (IMSIi, BLA), SNID, NT, N
The MME recognized the received MUEi identity, then it forward IMSIi and Base Location Area of the serving MME (BLA) hashed together F (IMSIi, BLA) to the HSS, combining with it the SNID, which denotes the MUEi access network, Network Type (NT) that is the network accessed by MUEi Evolved Universal Terrestrial Radio Access Network (E-UTRAN) and the requested authentication vectors N.
5) step 5: HSS –> MME: {RANDi, MIDgc} (GID (K, MIDgc)), AUTNhss (IMSI, RANDi, MIDgc), XRES (RANDi, MIDgc), F (IMSIi, BLA)
Using its KDF with its crypto function calculates the Authentication token of the HSS (AUTNhss), a one-way hash function of IMSI, RANDi, and MIDgc parameters, and use to authenticate MUEi and HSS. Later the HSS generates RANDi Upon acceptance of the Authentication data message sent by the MME. Moreover, calculating the Expected Response (XRES) is a one-way hash function to RANDi, MIDgc, and authenticates MUEi and MME. To protect the communication between HSS and MME against a forge attack, the HSS sends the RANDi, MIDgc encrypted by the secret hash function (GID (K, MIDgc)). In this proposal, the Group ID (GID) is the GID hash function generated upon acceptance of the authentication request from MME and is not generated until receiving the IMSIi of the group member who attached his authentication request. Once the HSS obtains IMSIi, it calculates the secret member GID using the registered MIDgc hashed with K. In addition, HSS sends both AUTNhss (IMSI, RANDi MIDgc,) and XRES (RANDi, MIDgc,) to MME.
6) Step 6: MME –> GC: {RANDi, MIDgc} (GID (K, MIDgc)), AUTNhss (IMSI, RANDi, MIDgc)
From the received authentication information, MME forward the encrypted concatenated value of RANDi and MIDgc by GID hash function {RANDi, MIDgc} (GID (K, MIDgc)) to GC and combines with it the AUTNhss (IMSI, RANDi, MIDgc) and keep the value of the XRES in MME stored to use it later to authenticate MUEi and MME.
7) step7: GC –> MUEi: {RANDi, MIDgc} (SEC (RNi, {RNg}Kecc)), AUTNgc (IMSIi, RANDi, MIDgc)
In this step, after receiving the message from MME, the GC keep the AUTNhss and send the encrypted concatenated value of RANDi and MIDgc by the shared secret EC key that is generated between it and MUEi {RANDi, MIDgc} (SEC (RNi, {RNgc}Kecc)) and AUTNgc (IMSI, RANDi, MIDgc) to MUEi. The goal of this step is to authenticate the MUEi and the GC.
8) step 8: MUEi –> GC: AUTNi (IMSIi, RANDi, MIDgc)
Once the MUEi receives RANDi and MIDgc, it will calculate its authentication parameters AUTNi (IMSIi, RANDi, MIDgc) to verify it with the receipted AUTNgc (IMSIi, RANDi, MIDgc) from GC and authenticate each other (GC-MUEi).
9) step 9: GC –> MUEi: AUTNhss (IMSIi, RANDi, MIDgc) The GC will send the kept value AUTNhss (IMSIi, RANDi, MIDgc) to the MUEi after he has authenticated it. The MUEi uses this value later to authenticate HSS.
10) step 10: MUEi –> MME: AUTNi (IMSIi, RANDi, MIDgc), RES (RANDi, MIDgc)
Finally, upon accepting the received message from GC, the MUEi will calculate RES (RANDi, MIDgc) and send its authentication parameters AUTNi to verify it with the receipted from GC AUTNhss (IMSI, RANDi, MIDgc) to authenticate HSS-MUEi. Moreover, send RES (RANDi, MIDgc) to MME to verify it with XRES (RANDi, MIDgc) for MME-MUEi authentication.

V. AVISPA TOOL SECURITY VERIFICATION

The GSAKA-ECDHKE protocol was built in the HLPSL [29] language and rigorously tested for security using the AVISPA tool [30]. The fundamental purpose of the protocol is to deliver mutual authentication between the MUEi and the network entities GC, MME, and HSS. Furthermore, the proposed protocol should be able to maintain the confidentiality of the shared secret EC key for each MUE (SEC (RNi, {RNg}Kecc)) and the immediate group key (GID (K, MIDgc)) during the authentication process. Fig.3 depicts the protocol's mechanism. The protocol's four core entities are MUEs, GC, MME, and HSS. The roles of these parties are described in HLPSL terminology in Fig 4,5,6,7 respectively. It is assumed that the channel between any two entities is insecure, and an attacker has control over the channel between any two Entities. The security analysis and verification results in the AVISPA tool simulation in Fig. 8 employing the OFMC and CL-AtSe backends are displayed in Fig.9 and Fig.10, respectively. The findings show that the GSAKA-ECDHKE protocol can achieve the goals while resisting all the attacks (such as a replay, MitM, and redirection attacks) that prohibit the protocol from attaining these goals.

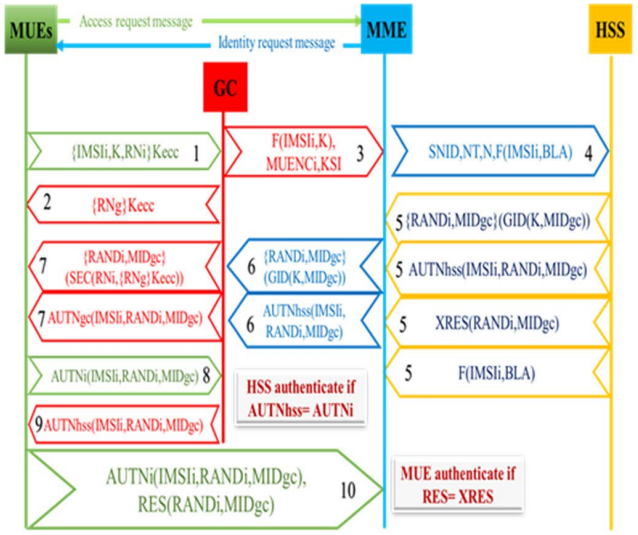


FIGURE 3. Proposed GSAKA-ECDHKE protocol mechanism.

A. SECURITY ANALYSIS

This section discusses the proposed GSAKA-ECDHKE protocol's security properties in terms of mutual authentication between protocol entities, a key agreement between them, the protection of shared secret EC keys, the group's privacy preservation, and MUEs represented in MIDgc that HSS provided it RANDi, MIDgc (GID (K, MIDgc)) via the protocol's mechanism. Additionally, after this section, we demonstrate why the suggested protocol is immune to all known assaults and capable of resolving the single key problem.

This section discusses the proposed GSAKA-ECDHKE protocol's security properties in terms of mutual authentication between protocol entities, a key agreement between them, the protection of shared secret EC keys, the group's privacy preservation, and MUEs represented in MIDgc that HSS provided it RANDi, MIDgc (GID (K, MIDgc)) via the protocol's mechanism. Additionally, after this section, we demonstrate why the suggested protocol is immune to all known assaults and capable of resolving the single key problem.

1) Any authentication protocol's basic security requirements are mutual to authenticate any entity sending or receiving a message to/from another entity. In the proposed, there are four mutual authentications.
   - The MME authenticates the GC when it sends the RANDi, MIDgc values encrypted by the hash function value GID (K, MIDgc), noting that HSS provided MIDgc during the protocol mechanism. The AVISPA has proved that the GC can recognize this value {RANDi, MIDgc} (GID (K, MIDgc)) and authenticate MME. If GC does not know the value of MIDgc and (GID (MID, MIDgc)), it will not accept the message from MME and cannot authenticate by MME.
   - The GC authenticates MUEi. The role of the GC is to control sending the MUEs authentication information to MME and forward the authentication parameters from MME to the MUEi. So, by using the ECKE, the GC and MUE exchanged their public keys and generated a secret EC key between them. As mentioned above, SEC (RNi, {RNgc}Kecc) or SEC (RNgc, {RNi}Kecc) are equal. The GC sends the {RANDi, MIDgc} SEC (RNi, {RNgc}Kecc) and AUTNgc (IMSIi, RANDi, MIDgc) to MUEi then the MUEi calculates its AUTNi (IMSIi, RANDi, MIDgc) and verifying it with the accepted AUTNgc to authenticate each other. Later the GC sends the AUTNhss (IMSIi, RANDi, MIDgc). AVISPA proves that GC can authenticate MUEi.
   - The MUEs authenticate HSS after receiving {RANDi, MIDgc} (SEC (RNi, {RNgc}Kecc)) and the authentication value AUTNhss (IMSIi, RANDi, MIDgc) that is calculated in HSS and forwarded by GC. The MUEi calculated its AUTNi (IMSIi, RANDi, MIDgc) value and verified it with AUTNhss.
   - Finally, the MME authenticates MUEs upon receiving the RES (RANDi, MIDgc) calculated by MUEi, and sent to MME and comparing it with the stored value XRES (RANDi, MIDgc), which it had generated in the HSS.
2) So, an adversary cannot generate the shared secret EC key without knowing the private key of each MUEi and GC, which is not sent out of these two entities. Also, the adversary cannot calculate the GID without knowing the value of MIDgc, which HSS provided it hashed to avoid exposing it to the adversary.
3) By encrypting MUEi identifiers (IMSIi) once with the asymmetric key IMSIi Kecc and hashing them once with the LTE key K F (IMSIi, K) [21], in contrast to the GSL-AKA protocol, which employs temporary identifiers via one-way hash functions, thereby increasing the computational time and complexity of the protocol mechanism. The suggested protocol protects privacy by utilizing encryption processes with a hash function for privacy preservation.
4) In the proposed protocol, to prevent the network signaling congestion, each MUEi selects its unique RNi and generates its public key {RNi} Kecc and

Exchanges it with the GC to generate the secret key SEC (RNgc, {RNi} Kecc) between them.

1) To decrease signaling congestion and communication overhead, the GC does not broadcast its public key to all group members but instead sends it to the MUEi, who attaches his authentication request to reduce the needed bandwidth from the start of the protocol mechanism. Following that, the GC and the MUEi exchanged their secret EC key to safeguard the authentication parameters they sent to each other. Furthermore, the HSS and MME must authenticate themselves to the MUE and the GC using the hashed GID (K, MIDgc). As a result,

```
role role_MUEi(
MUEi,GC,MME,HSS                          :agent
IMSIi,MIDgc,MUENCi                       :text
K,Kecc                                   :public_key
SND,RCV                                  :channel(dy))
played_by MUEi
def=
local
State                                    :nat
AUTNgc,SEC,AUTNhss,RES,AUTNi             :hash_func
RNi,RNg,RANDi                            :text
init
State := 0
transition
1. State  =0 /\ RCV(start) =|>
   State':=1 /\ RNi':=new()
             /\ SND({IMSIi.K.RNi'}_Kecc)
2. State  =1 /\ RCV({RNg'}_Kecc) =|> State':=2
7. State  =2 /\ RCV({RANDi'.MIDgc}_SEC(RNi.{RNg}_Kecc)
               .AUTNgc(IMSIi.RANDi'.MIDgc)) =|>
   State':=3 /\ secret(RANDi',sec_1,{MUEi,MME})
             /\ SND(AUTNi(IMSIi.RANDi'.MIDgc))
9. State  =3 /\ RCV(AUTNhss(IMSIi.RANDi'.MIDgc)) =|>
   State':=4 /\ secret(RANDi',sec_1,{MUEi,MME})
             /\ SND(AUTNi(IMSIi.RANDi'.MIDgc)
               .RES(RANDi.MIDgc))
end role
```

FIGURE 4. MUEi role in HLPSL.

```
role role_GC(
MUEi,GC,MME,HSS                          :agent
MIDgc                                    :text
K,Kecc                                   :public_key
SND,RCV                                  :channel(dy))
played_by GC
def=
local
State                                    :nat
KSI,MUENCi,RNi,RNg,IMSIi,RANDi   :text
F,GID,AUTNgc,SEC,AUTNi,AUTNhss           :hash_func
init
State := 0
transition
1. State  =0 /\ RCV({IMSIi'.K'.RNi'}_Kecc) =|>
   State':=1 /\ RNg':=new()
             /\ SND({RNg'}_Kecc)
             /\ KSI':=new() /\ MUENCi':=new()
             /\ SND(F(IMSIi'.K').MUENCi'.KSI')
6. State  =1 /\ RCV({RANDi'.MIDgc}_GID(K.MIDgc)
               .AUTNhss(IMSIi.RANDi'.MIDgc)) =|>
   State':=2 /\ secret(RANDi',sec_1,{MUEi,MME})
             /\ secret(GID,sec_2,{MME,GC})
             /\ SND({RANDi'.MIDgc}_SEC(RNi.{RNg}_Kecc)
               .AUTNgc(IMSIi.RANDi'.MIDgc))
8. State  =2 /\ RCV(AUTNi(IMSIi.RANDi'.MIDgc)) =|>
   State':=3 /\ secret(RANDi',sec_1,{MUEi,MME})
             /\ SND(AUTNhss(IMSIi.RANDi'.MIDgc))
end role
```

FIGURE 5. GC role in HLPSL.

```
role role_MME(
MUEi,GC,MME,HSS                          :agent
IMSIi,MIDgc,SNID,NT,BLA                  :text
K                                        :public_key
SND,RCV                                  :channel(dy))
played_by MME
def=
local
State                                    :nat
KSI,MUENCi,N,RANDi                       :text
F,AUTNhss,GID,,RES,AUTNi                 :hash_func
init
State := 0
transition
3. State  =0 /\ RCV(F(IMSIi.K).MUENCi'.KSI') =|>
   State':=1 /\ N':=new()
             /\ SND(SNID.NT.N'.F(IMSIi.BLA))
5. State  =1 /\ RCV({RANDi'.MIDgc}_GID(K.MIDgc)
               .AUTNhss(IMSIi.RANDi'.MIDgc)
               .F(IMSIi.BLA))  =|>
   State':=2 /\ secret(RANDi',sec_1,{MUEi,MME})
             /\ secret(GID,sec_2,{MME,GC})
             /\ SND({RANDi'.MIDgc}_GID(K.MIDgc)
               .AUTNhss(IMSIi.RANDi'.MIDgc))
10. State =2 /\ RCV(AUTNi(IMSIi.RANDi'.MIDgc)
               .RES(RANDi'.MIDgc)) =|>
   State':=3 /\ secret(RANDi',sec_1,{MUEi,MME})
end role
```

FIGURE 6. Role of the MME.

the proposed protocol keeps the network immune from experiencing signaling congestion.

2) Keep the session keys' ability to be unlinked. After successfully executing the proposed protocol, the session keys AUTNhss, AUTNi, AUTNgc, XRES, and RES between each MUEi and the network are changed using the newly generated random numbers RANDi and the HSS-supplied MIDgc. When one of these session keys is leaked, the adversary cannot connect it to the adversary's previous and subsequent session keys.
3) Additionally, to maintain the GID's unlinkability, whenever another MUEi wants to enter or leave the group, the GID remains unaffected because the GID is based on the LTE key K of the MUEi attached to the authentication request and the MIDgc produced by HSS. As a result, there is no way to connect the present group key to the previous or subsequent group keys. Additionally, only HSS has the authority to add or remove MUEi.
4) A most symmetric key, AKA protocol security, is entirely dependent on pre-shared secret keys, such as LTE key K, and if these keys are stolen, all other secret information may be retrieved, allowing an opponent to authenticate themselves to the network.

```
role role_HSS(
MUEi,GC,MME,HSS                          :agent
IMSIi,NT,SNID,MIDgc,BLA                  :text
K                                        :public_key
SND,RCV                                  :channel(dy))
played_by HSS
def=
local
State                                    :nat
N,RANDi                                  :text
F,GID,XRES,AUTNhss:hash_func
init
State := 0
transition
4. State  =0 /\ RCV(SNID.NT.N'.F(IMSIi.BLA)) =|>
   State':=1 /\ RANDi':=new()
             /\ secret(RANDi',sec_1,{MUEi,MME})
             /\ secret(GID,sec_2,{MME,GC})
             /\ SND({RANDi'.MIDgc}_GID(K.MIDgc)
               .AUTNhss(IMSIi.RANDi'.MIDgc)
               .XRES(RANDi'.MIDgc).F(IMSIi.BLA))
end role
```

FIGURE 7. HSS role in HLPSL.

As a result, to address the single most critical issue, this section describes a strategy for preserving pre-shared keys and resolving the problem of single key exposure. This strategy is comprised of two recommendations. Pre-shared secret keys should never be used explicitly as key generators. Second, an attacker who obtains these pre-shared keys will never be able to discover any session keys and authenticate to the network. Thus, the most critical property of the proposed protocol is that it can overcome the single key problem, which no other AKA protocol has been able to do.

- The proposed protocol uses the LTE key K as the key generator for the hash function and uses a combination of asymmetric and hash functions throughout the protocol mechanism, such as the shared secret EC key between the MUEi and the GC, which is an asymmetric key SEC (RNi, {RNgc}Kecc) and the hash function like GID, AUTNhss, AUTNi, AUTNgc, XRES, and RES.
- During the protocol process, hash functions are utilized to secure the BLA of the base station to prevent MME from impersonating F (IMSIi, BLA). Also, construct the shared secret EC key between the MUEi and the GC SEC (RNi, {RNgc}Kecc) and GID (K, MIDgc)

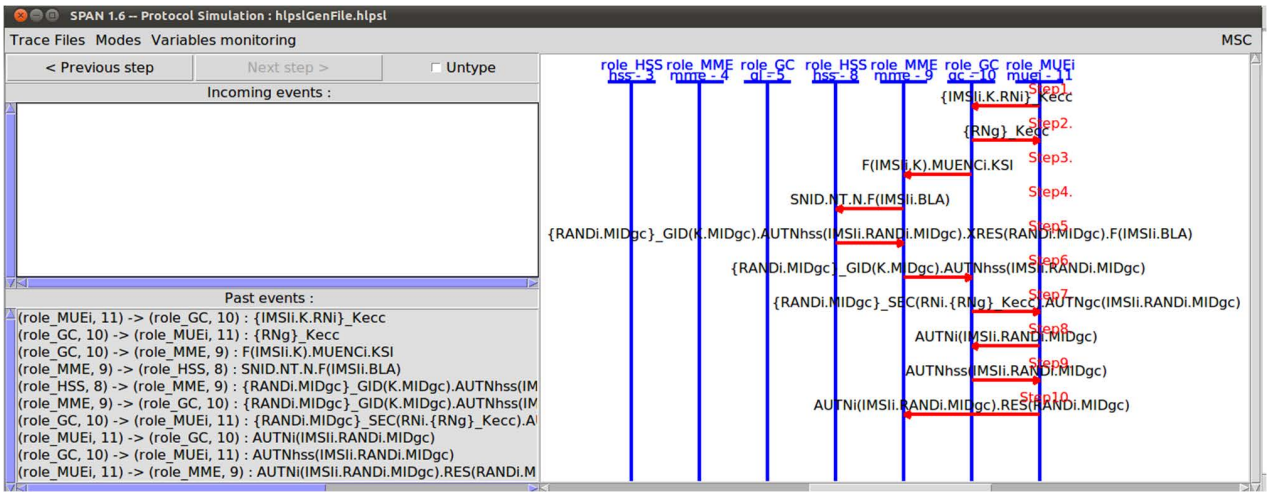


FIGURE 8. AVISPA simulation.

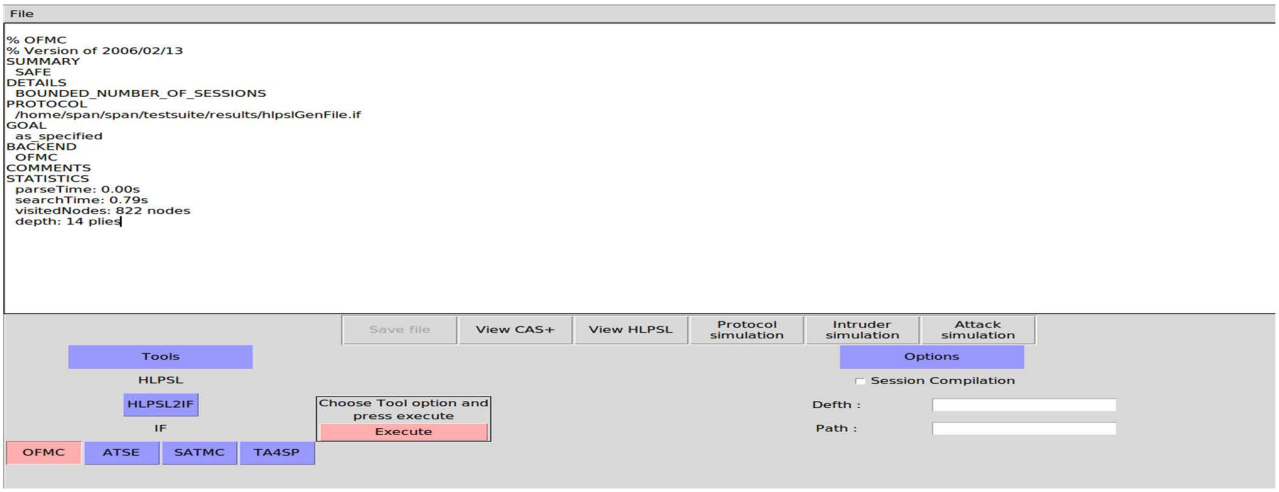


FIGURE 9. OFMC output goal.

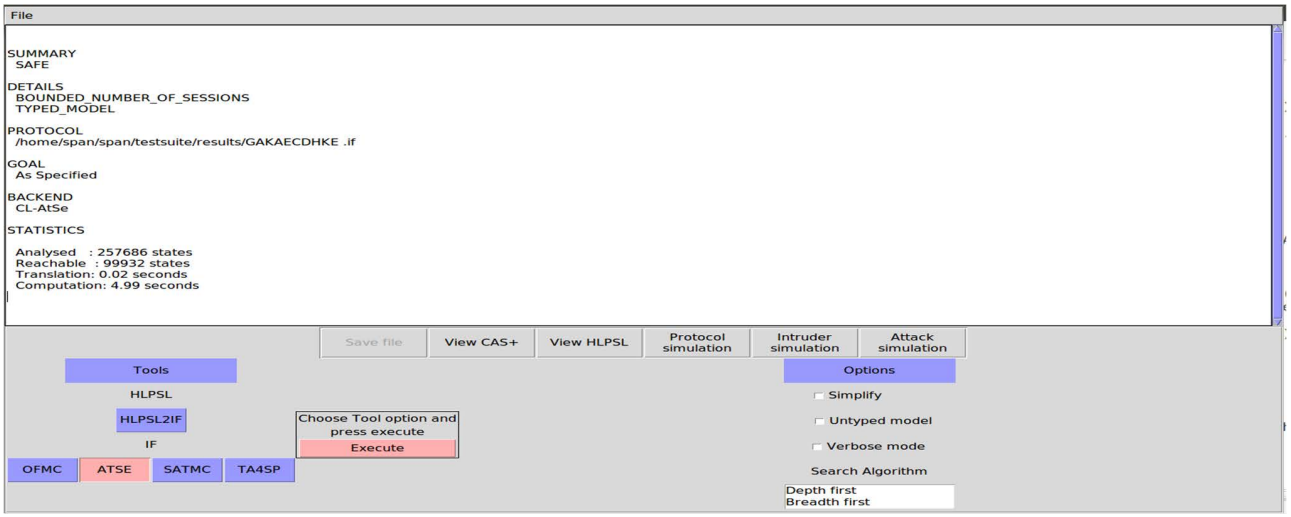


FIGURE 10. CL-AtSe output goal.

if the attacker earns the shared secret key SEC (RNi, {RNgc}Kecc) or GID (K, MIDgc) via eavesdropping on the channel, they would not be intelligent to gain the hash functions' inputs or to vulnerable secret data and keys. There are numerous private data in the proposed protocol, such as the IMSI and K of each MUEi,

MIDgc, and the RANDi, the output of HSS, which is used solely as one input of the authentication hash functions AUTNhss, AUTNi, AUTNgc, XRES, and RES and is never revealed in any means. Thus, the GSAKA-ECDHKE protocol can resolve the problem of single key exposure.

B. ANTI-ATTACKING CAPABILITY

Any authentication protocol needs to be immune to the various attack types to ensure the secrecy and confidentiality of its users, enabling the users to send and receive their information safely. Therefore, it is mandatory to discuss the capability of the proposed protocol against various attacks.

1) DOS ATTACK RESISTANCE

There are two forms of DoS attacks boosting network resources to use bandwidth and keep the network busy and halting network services. They all have the same goal to prevent legal MUEs from associating and reaping advantages from their network. The proposed protocol can prevent DOS attacks. In GAKA-ECDHKE, HSS generates RANDi immediately after receiving MUEi identifying information such as IMSIi, which are preregistered in the GIL. The authentication mechanism step 5 checks if the received IMSIi is in the GIL registered or not to determine if the request was issued from a genuine MUEi. As a result, HSS will reject communications from malicious DOS and conserve its resources to serve genuine MUEi. GSAKA-ECDHKE can evaluate the legitimacy of incoming requests early in the authentication process, and the interaction will be terminated if the DOS is not detected as registered in the GIL. An improvement over the procedures HSS supplies the MIDgc after accepting and confirming the MUEi identity IMSIi. Consequently, generating RANDi and pro-vide MIDgc encrypting it using the GID hash function {RANDi, MIDgc} (GID (K, MIDgc)).

In order to evaluate the resistance of the proposed protocol, AVISPA simulates a basic DOS attack. DOS is a new role formed with knowledge of the protocol's public keys and functionalities. When DOS makes an authentication request to the MME, the request is routed to HSS for verification. In step 3, the communication will be terminated. HSS discards DOS notifications because it can detect bogus MUEi quickly and easily without wasting home network server resources. Because DOS IMSI is not registered in the GIL, their request will be rejected. Therefore, the proposed protocol is immune enough to DoS attacks.

2) REPLAY ATTACK RESISTANCE

The proposed protocol embeds random numbers RANDi in the authentication parameters of each MUEi AUTNi and RES, AUTNgc for GC, and the HSS and MME AUTNhss authentication parameters, respectively, to resist replay attacks [31]. As a result, these random integers prevent these authentication settings from replaying and reusing. Replay attacks are conducted by resending a previously authorized communication from a prior session. Two sessions with identical input parameters are represented in parallel in the environment role to identify potential replay attacks. If the GSAKA-ECDHKE protocol is open to replay attacks, an adversary can intercept messages transmitted by a valid entity in one session and delay or replay them in another without being discovered by other entities. In AVISPA, this assault has been modeled individually and in conjunction with other attacks. Both options result in a SAFE condition. Thousands of scenarios have been studied, as evidenced by the AVISPA findings, and no attack has been discovered. As a result, it is possible to conclude that the GSAKA-ECDHKE protocol is immune to replay attacks.

3) IMPERSONATE MME ATTACK

The GC selected the group members in the proposed protocol based on certain rules related to the same tasks and location. As a result, each MME must provide its BLA to HSS, which it has acknowledged. In a redirection attack, an attacker sets up a fake base station to imitate a legitimate MME and get access to MUEi-protected data. The BLA of the connected base station is contained in the authentication parameter F (IMSI, BLA) in the proposed protocol to avoid redirection attacks. When the HSS computes F (IMSI, BLA) using the BLA given by MME and discovers that F (IMSI, BLA) sent by MME is not equal to F (IMSI, BLA), it detects the attack occurs and denies the authentication request.

A session with an intruder assuming the role of MME is simulated in AVISPA to test the GSAKA-ECDHKE protocol under the forge MME attack. In this scenario, the intruder assumes the role of MME and may read all messages sent to MME and communications from another genuine session. Following the testing, AVISPA was completed, and the output was SAFE. As a result, it is possible to conclude that the GSAKA-ECDHKE protocol is immune to MME fake station attacks.

4) MITM ATTACK RESISTANCE

The authentication parameters of each MUEi AUTNi and RES and the authentication parameters of the HSS and MME (AUTNhss and XRES) and AUTNgc for GC are formed in the proposed protocol utilizing new secret generated data such as RANDi, MIDgc. An adversary can never create a MITM attack and produce these authentication parameters to authenticate itself instead of a valid MUEi to the network if they are unknown during the pre-authentication procedures. The protocol then defends against MITM attacks.

The AVISPA program employs the Dolev-Yao intruder model, implying that the intruder can monitor all network information flow and truncate, add, and redirect messages. Because all communications transmitted in GSAKA-ECDHKE are encrypted, an intruder cannot decipher the information.

The integrity protecting component, resulting from the hash function, is present in each message. Even little changes to the source content will result in large changes to the hash output. As a result, the receiver can determine whether the

TABLE 2. Comparative security features analysis.

Security Feature (SF)									
Protocol name	SF1	SF2	SF3	SF4	SF5	SF6	SF7	SF8	SF9
Our proposed GSAKA-AKA	Asymmetric	YES	YES	YES	YES	YES	YES	YES	YES
G-AKA[13]	Symmetric	YES	YES	NO	NO	NO	YES	NO	NO
SE-AKA[15]	Hybrid	YES	NO	YES	NO	YES	YES	YES	NO
EG-AKA[16]	Hybrid	YES	NO	YES	NO	NO	YES	YES	NO
NOVEL-AKA[18]	Symmetric	YES	YES	YES	NO	NO	YES	YES	NO
GBAAM-AKA[20]	Asymmetric	YES	NO	NO	NO	YES	YES	YES	YES
GROUP-AKA[19]	Symmetric	YES	NO	NO	NO	NO	YES	YES	YES
GLARM-AKA[22]	Symmetric	YES	YES	YES	NO	YES	NO	YES	YES
PRIVACY-AKA[21]	Hybrid	YES	NO	NO	NO	YES	YES	YES	YES
GR-AKA[23]	Asymmetric	YES	NO	YES	NO	YES	YES	YES	YES
GBS-AKA[24]	Symmetric	YES	YES	NO	NO	NO	NO	YES	YES
SEGB-AKA[25]	Symmetric	YES	YES	YES	YES	YES	YES	YES	YES
GSL-AKA[33]	None (Hash)	YES	YES	YES	YES	YES	YES	YES	YES

SF1: Type of cryptosystem; SF2: Basic security requirements; SF3: Follow the 3GPP standard; SF4: Privacy preservation; SF5: Solve the single key problem; SF6: DoS attack resistance; SF7: Impersonate MME attack; SF8: MITM attack resistance; SF9: Avoid network signaling congestion

message was altered during transmission. The timestamp is another key component that protects messages from intruders. Even though the intruder can reroute and replay every message in the network, the receiver can always establish the message's freshness based on this component. Therefore, a MITM attack [32]can be avoided. In other words, the GSAKA-ECDHKE protocol is immune to data alteration and eavesdropping attacks.

The comparative security features analysis between the proposed protocol and the previous group-based AKA protocols is shown in Table 2. It is observed that the proposed protocol follows the asymmetric key cryptosystem methodology, and it can achieve all the security requirements. The protocol avoids all the identified attacks in the communication network and maintains key forward and backward security. In addition, the protocol preserves the privacy of MUEs and avoids a single key problem. Hence, the GSAKA-ECDHKE protocol is superior to all the existing group-based AKA protocols.

C. PERFORMANCE ANALYSIS

In this section, we compare the communication overhead of our proposed GSAKA-ECDHKE protocol to that of existing group-based AKA protocols and demonstrate that the proposed protocol has the lowest overhead. To properly assess the communication overhead, it is necessary to note that the HSS does not provide the MIDgc to any MUEi with an authentication request attached. However, the HSS provided the MIDgc for the MUEi, which is preregistered as a group member in the GIL concerning this MIDgc. Although the overhead associated with creating groups, joining or leaving groups, and distributing group keys is negligible in the proposed protocol, we ignored them when computing the total overhead because each MUEi group member is authenticated individually with the GC.

1) COMMUNICATION OVERHEAD

The protocol's communication overhead is the total number of bits transmitted via the protocol mechanism. According to Fig.2 and Table.1, the proposed protocol's communication overhead can be calculated as follows:

1) Total Bits (TB) in step 1:
   {IMSIi, K, RNi} Kecc = 128+128+192=448
2) TB in step 2:
   {RNg}Kecc = 192
3) TB in step 3:
   F (IMSIi, K), MUENCi, KSI = 64+4+3=71
4) TB in step 4:
   F (IMSIi, BLA), SNID, NT= 64+40+16=120
5) TB in step 5:
   {RANDi, MIDgc} (GID (K, MIDgc)), AUTNhss (IMSI, RANDi, MIDgc), XRES (RANDi, MIDgc), F (IMSIi, BLA) = 256+64+64+64=448
6) TB in step 6:
   {RANDi, MIDgc} (GID (K, MIDgc)), AUTNhss (IMSI, RANDi, MIDgc)
   = 256+64=320
7) TB in step 7: {RANDi, MIDgc} (SEC (RNi, {RNg}Kecc)), AUTNgc (IMSIi, RANDi, MIDgc) = 256+64=320
8) TB in step 8:
   AUTNi (IMSIi, RANDi, MIDgc) = 64
9) TB in step 9:

TABLE 3. Communication overheads.

Protocol name	Number of massages	Total number of bits
Our proposed GSAKA-ECDHKE	10	2175
G-AKA[13]	8	2768
SE-AKA[15]	8	4536
EG-AKA[16]	12	4928
NOVEL-AKA[18]	8	3504
GBAAM-AKA[20]	10	5251
GROUP-AKA[19]	9	4480
GLARM-AKA[22]	8	3960
PRIVACY-AKA[21]	8	4320
GR-AKA[23]	7	2280
GBS-AKA[24]	6	2992
SEGB-AKA [25]	8	3512
GSL-AKA[33]	8	2504

TABLE 4. Computation complexity.

Protocol name	Total CC
Our proposed GSAKA-ECDHKE	$(5T_{hash}+3T_{enc})$ u $+(4T_{hash}+T_{enc})$ g.
G-AKA[13]	$(7\ T_{hash})$ u $+(2T_{hash})$ g.
SE-AKA[15]	$(4T_{mul}+\ 6T_{hash})$ u $+(2T_{hash})$ g $+\ T_{hash}$
EG-AKA[16]	$(4T_{mul}+5T_{hash}+3T_{enc})$ u $+(T_{hash}+T_{enc})$ g.
NOVEL-AKA[18]	$(6T_{hash})$ u $+(2T_{hash})$ g.
GBAAM-AKA[20]	$(7T_{mul}+4T_{hash}+3T_{mtp})$ u $+(T_{mul}\ 2T_{pair}+T_{hash})$ g.
GROUP-AKA[19]	$(2T_{mod}+4T_{hash})$ u $+(8T_{hash}+T_{enc})$ g.
GLARM-AKA[22]	$(7T_{hash})$ u $+(4T_{hash})$ g.
PRIVACY-AKA[21]	$(8T_{hash}+3T_{mul})$ u $+(6T_{hash}+T_{mul})$ g.
GR-AKA[23]	$(T_{LC}+2T_{mul}+T_{hash})$ u $+(5T_{hash}+T_{LC}+2T_{mul})$ g.
GBS-AKA[24]	$(5T_{hash})$ u $+(2T_{hash})$ g.
SEGB-AKA[25]	$(7T_{hash}+4T_{enc})$ u $+(4T_{hash})$ g.
GSL-AKA[33]	$(4T_{hash})$ u $+(8T_{hash})$ g.

AUTNhss (IMSIi, RANDi, MIDgc) = 64

10) TB in step 10:
AUTNi (IMSIi, RANDi, MIDgc), RES (RANDi, MIDgc) = 64+64=128

The proposed protocol's total communication overhead equals the sum of the overheads calculated previously and is equal to =2175 bits for each group. Like our calculation method, the communication overhead of other group-based AKA protocols is calculated in Table.3 illustrates a comparative analysis of the communication overhead of existing group-based AKA protocols in the case of a single group.

It is observed that the proposed protocol achieves the lowest communication overhead compared to all other AKA protocols.

2) COMPUTATIONAL COMPLEXITY

The total Computational Complexity (CC) generated by each protocol is the computation time of the cryptographic functions used in the protocol. Where the hash operation time (Thash) = 0.067 ms and encryption time (Tenc) = 0.161 ms referring to [19], [23]. So, the CC for our proposed equal:

1) at the MUEi devices is equal to:
(3Thash+1Tenc) u + (2Thash+2Tenc) g
2) at the network is:
(3Thash) u + (1Thash+1Tecn) g.
3) Thus, the total CC of GSAKA-AKA is equal to:
(5Thash+3Tenc) u + (4Thash+1Tenc) g.

The comparative analysis of the CC of the proposed GSAKA-ECDHKE protocol with the existing group-based AKA protocols is presented in Table. 4. Where u is the number of MUE and g is the number of the groups.

VI. CONCLUSION

This study introduced a reliable military authentication protocol. The GSAKA-ECDHKE protocol authenticates MUEs in LTE networks by activating multi-mutual authentication among protocol entities (MME, GC), (GC, MUEs), (MUEs, MME), and (MUEs, HSS). Recently, group-AKA protocols have employed long parameters because they concatenated member identifiers. When a member quits or joins, they must recalculate the group ID and secret keys. This increases processing time and complexity. Our proposed protocol decreases the variable length and ignores the time needed to recalculate group ID and secret keys. By individually authenticating the group members with the GC, the performance analysis revealed that the proposed protocol has suitable communication overheads for group communication to re-authenticate the network entities. So, for this reason, the GSAKA-ECDHKE is robust for group re-authentication in the case of handover group authentication. Our proposal is proven to overcome the weaknesses and threats of the LTE standard authentication protocol utilizing the two generated secret keys. The protocol met all security goals and stopped known attacks, like DOS, MIMT, Reply, and impersonated MME attacks. It also helped protect the privacy of MUEs and fixed the one major problem that previous group-based AKA protocols could not fix, which was discussed in the security analysis.

REFERENCES

[1] L. J. Vora, "Evolution of mobile generation technology: 1G to 5G and review of upcoming wireless technology 5G," *Int. J. Mod. Trends Eng. Res.*, vol. 2, no. 10, pp. 281–290, 2015.

[2] X. Yan and M. Ma, "A privacy-preserving handover authentication protocol for a group of MTC devices in 5G networks," *Comput. Secur.*, vol. 116, May 2022, Art. no. 102601.

[3] Z. A. Zukarnain, A. Muneer, and M. K. A. Aziz, "Authentication securing methods for mobile identity: Issues, solutions and challenges," *Symmetry*, vol. 14, no. 4, p. 821, Apr. 2022.

[4] A. Mallik, "Man-in-the-middle-attack: Understanding in simple words," *J. Pendidikan Teknol. Inf.*, vol. 2, no. 2, pp. 109–134, 2019.

[5] H. Zhang, P. Cheng, L. Shi, and J. Chen, "Optimal DoS attack scheduling in wireless networked control system," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 3, pp. 843–852, May 2016.

[6] *Digital Cellular Telecommunications System (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; 3G Security; Network Domain Security (NDS)*, document TS 33.204, Version 10.0.0, Release 10, 3GPP, Transaction Capabilities Application Part (TCAP), 2011.

[7] G. M. Koien, "Mutual entity authentication for LTE," in *Proc. 7th Int. Wireless Commun. Mobile Comput. Conf.*, Jul. 2011, pp. 689–694.

[8] M. Abdeljebbar and R. El Kouch, "Security improvements of EPS-AKA protocol," *Int. J. Netw. Secur.*, vol. 20, no. 4, pp. 636–644, 2018.

[9] S. Chourasia and K. M. Sivalingam, "SDN based evolved packet core architecture for efficient user mobility support," in *Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft)*, Apr. 2015, pp. 1–5.

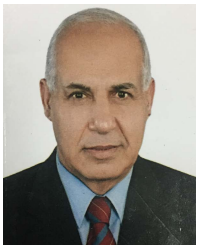
[10] V. T. H. Ahn and M. Ma, "A secure authentication protocol with performance enhancements for 4G LTE/LTE—A wireless networks," in *Proc. 3rd Int. Electron. Commun. Conf. (IECC)*, Jul. 2021, pp. 28–36.

[11] Z. Shang, M. Ma, and X. Li, "A secure group-oriented device-to-device authentication protocol for 5G wireless networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7021–7032, Nov. 2020.
[12] Y. Xia, R. Qi, S. Ji, J. Shen, T. Miao, and H. Wang, "PUF-assisted lightweight group authentication and key agreement protocol in smart home," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–15, Mar. 2022.
[13] Y. Chen, J. Wang, K. Chi, and C. Tseng, "Group-based authentication and key agreement," *Wireless Pers. Commun.*, vol. 62, no. 4, pp. 965–979, Feb. 2012.
[14] P. K. Panda and S. Chattopadhyay, "An enhanced mutual authentication and security protocol for IoT and cloud server," *Inf. Secur. J., A Global Perspective*, vol. 31, no. 2, pp. 144–156, Mar. 2022.
[15] C. Lai, H. Li, R. Lu, and X. S. Shen, "SE-AKA: A secure and efficient group authentication and key agreement protocol for LTE networks," *Comput. Netw.*, vol. 57, no. 17, pp. 3492–3510, 2013.
[16] R. Jiang, C. Lai, J. Luo, X. Wang, and H. Wang, "EAP-based group authentication and key agreement protocol for machine-type communications," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 11, Nov. 2013, Art. no. 304601.
[17] S. Chandra, S. Paira, S. S. Alam, and G. Sanyal, "A comparative survey of symmetric and asymmetric key cryptography," in *Proc. Int. Conf. Electron., Commun. Comput. Eng. (ICECCE)*, Nov. 2014, pp. 83–93.
[18] C. Lai, H. Li, X. Li, and J. Cao, "A novel group access authentication and key agreement protocol for machine-type communication," *Trans. Emerg. Telecommun. Technol.*, vol. 26, no. 3, pp. 414–431, Mar. 2015.
[19] D. Choi, H.-K. Choi, and S.-Y. Lee, "A group-based security protocol for machine-type communications in LTE-advanced," *Wireless Netw.*, vol. 21, no. 2, pp. 405–419, 2015.
[20] J. Cao, M. Ma, and H. Li, "GBAAM: Group-based access authentication for MTC in LTE networks," *Secur. Commun. Netw.*, vol. 8, no. 17, pp. 3282–3299, 2015.
[21] A. Fu, J. Song, S. Li, G. Zhang, and Y. Zhang, "A privacy-preserving group authentication protocol for machine-type communication in LTE/LTE—A networks," *Secur. Commun. Netw.*, vol. 9, no. 13, pp. 2002–2014, 2016.
[22] C. Lai, R. Lu, D. Zheng, H. Li, and X. Shen, "GLARM: Group-based lightweight authentication scheme for resource-constrained machine to machine communications," *Comput. Netw.*, vol. 99, pp. 66–81, Apr. 2016.
[23] J. Li, M. Wen, and T. Zhang, "Group-based authentication and key agreement with dynamic policy updating for MTC in LTE—A networks," *IEEE Internet Things J.*, vol. 3, no. 3, pp. 408–417, Jun. 2016.
[24] J. Yao, T. Wang, M. Chen, L. Wang, and G. Chen, "GBS-AKA: Group-based secure authentication and key agreement for M2M in 4G network," in *Proc. Int. Conf. Cloud Comput. Res. Innov. (ICCCRI)*, May 2016, pp. 42–48.
[25] N. S. Chaudhari, "SEGB: Security enhanced group based AKA protocol for M2M communication in an IoT enabled LTE/LTE—A network," *IEEE Access*, vol. 6, pp. 3668–3684, 2018.
[26] R. R. Ahirwal and M. Ahke, "Elliptic curve Diffie–Hellman key exchange algorithm for securing hypertext information on wide area network," *Int. J. Comput. Sci. Inf. Technol.*, vol. 4, no. 2, pp. 363–368, 2013.
[27] I. Setiadi, A. I. Kistijantoro, and A. Miyaji, "Elliptic curve cryptography: Algorithms and implementation analysis over coordinate systems," in *Proc. 2nd Int. Conf. Adv. Informat., Concepts, Theory Appl. (ICAICTA)*, Aug. 2015, pp. 1–6.
[28] S. K. Sharon, "The elliptic curve cryptography cofactor Diffie–Hellman (ECC CDH) primitive validation system (ECC_CDHVS)," NIST Inf. Technol. Lab., Gaithersburg, AR, USA, Tech. Rep. SP 800-56A, 2011.
[29] A. Muñoz, A. Maña, and D. Serrano, "Model cheking ambient intelligence with avispa," in *Ambient Intelligence Perspectives*. IOS Press, 2009, pp. 182–193.
[30] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. H. Drielsma, P. C. Héam, O. Kouchnarenko, J. Mantovani, and S. Mödersheim, "The AVISPA tool for the automated validation of internet security protocols and applications," in *Proc. Int. Conf. Comput. Aided Verification*, 2005, pp. 281–285.
[31] R. Pries, W. Yu, X. Fu, and W. Zhao, "A new replay attack against anonymous communication networks," in *Proc. IEEE Int. Conf. Commun.*, 2008, pp. 1578–1582.
[32] M. A. Al-shareeda, M. Anbar, S. Manickam, and I. H. Hasbullah, "Review of prevention schemes for man-in-the-middle (MITM) attack in vehicular ad hoc networks," *Int. J. Eng. Manage. Res.*, vol. 10, no. 3, pp. 153–158, Jun. 2020.
[33] M. M. Modiri, J. Mohajeri, and M. Salmasizadeh, "A novel group-based secure lightweight authentication and key agreement protocol formachine-type communication," *Scientia Iranica*, Feb. 2021.

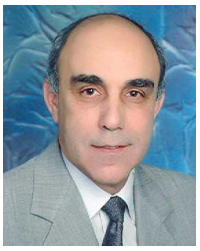
KARIM H. MOUSSA received the B.Sc., M.Sc., and Ph.D. degrees from Alexandria University, Alexandria, Egypt, in 2006, 2011, and 2016, respectively.

He joined the School of Internet of Things, Xi'an Jiaotong-Liverpool University, China, in 2022, where he is currently an Assistant Professor of computer, communications, and electronics. His research interests include digital signal processing, multimedia, data security, communication networks, mobile communication systems, multiuser MIMO systems, massive MIMO, industrial wireless data acquisition systems, and engineering optimization.

AHMED H. EL-SAKKA received the B.Sc. and M.Sc. degrees from Alexandria University, Alexandria, Egypt, in 2004 and 2015, respectively. He joined as the Teaching Assistant Staff with the Electronics and Communications Department, Air Defense College, Alexandria University, in 2008, where he is currently an Assistant Staff teaching digital communication systems and electronics. His research interests include digital communication systems, 4G, 5G mobile, data security, communication networks, and FPGA programming.



SHAWKY SHAABAN received the B.Sc., M.Sc., and Ph.D. degrees from Alexandria University, Alexandria, Egypt, in 1974, 1982, and 1990, respectively. He joined as the Teaching Assistant Staff with the Electronics and Communications Department, Faculty of Engineering, Alexandria University, in 1975, where he is currently an Assistant Professor of digital communications, computers, communications systems, and electronics. His research interests include mobile communications, satellite communication, radar systems, and visual studio programming.



HASSAN NADIR KHEIRALLAH received the B.Sc. degree in electrical engineering from Alexandria University, in 1972, and the M.Sc. and Ph.D. degrees from Carleton University, Canada, in 1974 and 1980, respectively. He was the former Dean with the Faculty of Engineering, Beirut University, from 1997 to 1999. He is currently a Professor of microwave engineering with the Faculty of Engineering, Alexandria University. He was the President of Alexandria University (2006–2009), the French University of Egypt (2012–2018), and Senghor University (2010–2016). He was a member of the Board of Trustees of Bibliotheca Alexandria and the Egypt-Japan University for Science and Technology (E-JUST). He was the Chairperson of the project management at the Ministry of Higher Education, Egypt. He is currently a member of the Management Board of EMUNI University and a Councilor for the President of Alexandria University for international relations. He received the Scientific Research Prize and Taha Hussein Prize from Alexandria University, in 1991 and 2008, respectively, and the "Chevalier de l'ordre National du Merite Francais" from the Republic of France.