Hindawi
Security and Communication Networks
Volume 2017, Article ID 1701243, 21 pages
https://doi.org/10.1155/2017/1701243

Research Article

Hierarchical Group Based Mutual Authentication and Key Agreement for Machine Type Communication in LTE and Future 5G Networks

Probidita Roychoudhury,[1] Basav Roychoudhury,[2] and Dilip Kumar Saikia[1]

[1]*National Institute of Technology Meghalaya, Shillong, Meghalaya, India*
[2]*Indian Institute of Management, Shillong, Meghalaya, India*

Correspondence should be addressed to Probidita Roychoudhury; probidita.phukan@gmail.com

Received 26 July 2016; Revised 1 October 2016; Accepted 12 October 2016; Published 19 January 2017

Academic Editor: Muhammad Khurram Khan

In view of the exponential growth in the volume of wireless data communication among heterogeneous devices ranging from smart phones to tiny sensors across a wide range of applications, 3GPP LTE-A has standardized Machine Type Communication (MTC) which allows communication between entities without any human intervention. The future 5G cellular networks also envisage massive deployment of MTC Devices (MTCDs) which will increase the total number of connected devices hundredfold. This poses a huge challenge to the traditional cellular system processes, especially the traditional Mutual Authentication and Key Agreement (AKA) mechanism currently used in LTE systems, as the signaling load caused by the increasingly large number of devices may have an adverse effect on the regular Human to Human (H2H) traffic. A solution in the literature has been the use of group based architecture which, while addressing the authentication traffic, has their share of issues. This paper introduces Hierarchical Group based Mutual Authentication and Key Agreement (HGMAKA) protocol to address those issues and also enables the small cell heterogeneous architecture in line with 5G networks to support MTC services. The aggregate Message Authentication Code based approach has been shown to be lightweight and significantly efficient in terms of resource usage compared to the existing protocols, while being robust to authentication message failures, and scalable to heterogeneous network architectures.

1. Introduction

The market for cellular data is growing at a tremendous pace with the birth of a new generation of cellular system at almost every decade. This has been largely triggered by the explosive growth in mobile traffic coupled with advancements in wireless communication technologies. As per the Cisco Visual Networking Index 2014–19 [1], machine to machine (M2M) also termed as Machine Type Communication (MTC) connections are expected to grow to 10.5 billion by 2019. M2M or MTC communication refers to communication between entities without any human intervention, intended to support applications like home automation, security and surveillance, healthcare, traffic management, and many others. The communicating entities are termed Machine Type Communication Devices (MTCDs). Standardization organizations, like Third Generation Partnership Project (3GPP), European Telecommunications Standards Organization (ETSI), and so forth, have already come up with standards on MTC architecture and security architecture under 3GPP Long Term Evolution Advanced (LTE-A) [2]. Under the 7th Framework Programme for Research (7FP) [3], the European Union has initiated several programs to promote research in technologies to build wired and wireless communication networks of the future. A key project, Mobile and wireless communication Enablers for Twenty-twenty Information Society (METIS) [4], the flagship program of the European Union on Fifth Generation Cellular Network (5G), has attempted to prepare the groundwork for 5G network before any standardization activities are carried out. The scenarios and use cases analyzed by METIS [5] indicate a massive deployment of MTCDs in future cellular networks. The existing cellular network is ill-equipped to handle the surge in traffic from such massive MTC deployment. The traffic

generated from the authentication procedure, executed for security reasons by each MTCD before they can attach to the network, becomes a serious concern in such scenarios. A sizable number of MTCDs trying to attach to the network simultaneously with each having to undergo the Mutual Authentication and Key Agreement (AKA) procedure result in massive signaling overload at both access network and core network.

The existing literature on MTC authentication has tried to find different approaches to reduce the signaling traffic needed for the AKA procedure by grouping the MTCDs based on different criteria like belonging to the same application, appearing in the same location, and so forth. The schemes exploiting the group based approach use one of the following methods for reducing the AKA signaling load:

(1) First MTCD performs a full AKA with the core network and also authenticates the group by fetching the authentication data for the entire group to the serving network; the rest of the group members authenticates locally at the serving network using the prefetched authentication data.

(2) Each MTCD sends its authentication message to a MTCD selected from among the group to be a group leader who in turn aggregates the same and transmits it to the core network.

The main drawback of the first approach is that while it reduces the signaling load between the home network and the serving network, the load at the access network remains unchanged. The latter approach successfully reduces the load at the access network but a single corrupt authentication message in the aggregate will result in the rejection of the authentication of the entire group. Furthermore, the group leader based approach also includes the complexity and consequent delay of group leader selection and, in the event of group leader failure/exit, reselection of the new incumbent. Hence, a natural corollary to these contributions would be one that addresses these limitations.

In this paper, the authors propose the Hierarchical Group based Mutual Authentication and Key Agreement (HGMAKA) protocol for MTC which will be suitable for both current LTE-A and future 5G networks. The main contributions of this paper are the following:

(1) Introduction of a hierarchical approach, as against the existing first MTCD based or group leader based approach, for implementation of group based AKA between a group of MTCDs and the Core Network.

(2) Adoption of hierarchical small cell architecture, in line with 5G architecture, for reduction in signaling load due to authentication as against macrocell architecture in existing literature.

(3) Introduction of en route integrity verification of authentication messages to avoid batch reauthentication due to failures.

(4) Performance comparison with ten other group based schemes in the literature.

The proposed group based hierarchical protocol uses the support of network elements instead of a group leader, thereby eliminating the additional complexity related to group leader management. In view of resource constrained devices used in most M2M applications, it uses the lightweight symmetric key based aggregate Message Authentication Code (MAC) approach with integrity verification of authentication messages at each level of the hierarchy. This eliminates the chance of group authentication failure at the core network which can otherwise be caused, in the existing group based schemes using aggregate MAC, by even a single corrupt MAC appearing in the aggregate authentication message.

Furthermore, the use of small cells helps in offloading traffic from the macrocells to small cells resulting in further reduction in signaling load at the access network level. This is also in line with the architectures of future cellular networks as proposed by several projects like METIS, iJOIN [6], TROPIC [7], and so forth advocating the use of small cells. Depending on applications, MTCDs may be deployed in low coverage areas like underground parking or remote inaccessible locations where connectivity to cellular networks may not be feasible. The proposed HGMAKA scheme takes into consideration such cases through the use of small cells and capillary network of devices. Being hierarchical in nature, it allows larger group sizes through multilevel aggregation resulting in lesser number of groups for a given number of MTCDs. Moreover, highly mobile MTCDs like those in public vehicles will need frequent handovers resulting in high overhead. The proposed protocol uses mobile femtocells to address this issue.

The rest of the paper is organized as follows: Section 2 provides an overview of MTC under LTE-A, the mutual AKA protocol, and the importance of heterogeneous cellular network architecture; Section 3 spells out the motivation behind the proposed protocol; Section 4 reviews some related works from the literature on group based mutual AKA protocols and highlights the issues therein; Sections 5 and 6 present the proposed system architecture along with the proposed protocol; Section 7 presents the security analysis of the protocol; Section 8 provides performance analysis of the proposed protocol vis-à-vis existing group based protocols; and Section 9 concludes the paper.

2. Machine Type Communication and Small Cell Architecture

2.1. Machine Type Communication. Machine Type Communication (MTC) [8] also known as M2M communication refers to the communication between entities without any human intervention. This communication mainly entails collection of data by the MTCD and transmitting to an MTC Server which processes the data and initiates some action. There can also be scenarios where the MTC Server triggers the MTCD to perform some action. A third scenario may involve two MTCDs communicating among themselves with or without involving any MTC Server. Standardization organizations, like 3GPP, have published specifications to enable optimization of 3G and LTE cellular networks for MTC traffic. The MTCD connects to the LTE core network, also

known as Evolved Packet Core (EPC), via the evolved Node B (eNodeB/eNB) which is the base station in the Radio Access Network (RAN), named Evolved Universal Terrestrial RAN (eUTRAN) in case of LTE. LTE also supports other non-3GPP RAN which can be either trusted or untrusted, depending on the operator's policies. The RAN, and thereby, the base station, can differ in case of trusted/untrusted 3GPP/non-3GPP networks. Before the MTCD can communicate over the LTE network, it has to perform a mutual authentication with the core network. The Mobility Management Entity (MME) represents the network in this authentication procedure. In case of trusted non-3GPP access networks, the Authentication, Authorization, and Accounting (AAA) Server, and for untrusted non-3GPP networks, the evolved Packet Data Gateway (ePDG), performs the authentication with the MTCD. The Home Subscriber Server (HSS) is the central database of all authentication information and assists the MME/AAA/ePDG in the authentication procedure. The Packet Data Gateway (PDG) entity provides connectivity to external packet data networks.

In case of indoor traffic, a smaller and less powerful base station, named Home eNodeB (HeNB) [9], can be used, which improves the coverage in indoor locations with higher data rates. The HeNB is a small, less expensive base station that is located at the customer site and connects to the EPC via the customer's existing broadband access line like DSL or broadband cable. Multiple HeNBs can be managed by an optional element of the EPC called the Home eNodeB Gateway (HeNB-GW). It multiplexes connections from multiple HeNBs into a single stream in order to alleviate the load on the MME. The HeNB-GW is transparent to the MME as well as to the HeNBs. Figure 1 gives the network architecture of LTE.

A Mutual Authentication and Key Agreement procedure is carried out between the MME/AAA/ePGD and the MTCD. The authentication procedure used is the Evolved Packet System-Authentication and Key Agreement (EPS-AKA) [10] for 3GPP and Extensible Authentication Protocol-AKA (EAP-AKA) [11] for non-3GPP networks. On successful completion of the authentication procedure, both parties share a secret session key to be used for confidentiality and integrity protection or for deriving next level keys to be used for the same purpose. Figure 2 shows the steps involved in the EPS-AKA protocol.

2.2. Small Cell Architecture and HetNets. As mentioned in the previous section, the growing number of MTC connection is a concern for the current cellular network which will face tremendous challenges in providing the desired level of user satisfaction to both M2M and Human to Human (H2H) communications. One of the possible approaches to handle hyper densification of cells is to offload traffic into smaller cells served by low powered base stations like micro and pico eNodeBs (eNB), Home eNodeBs (HeNB), Relay Nodes (RN), and even Remote Radio Heads (RRH) (radio elements which have a wired connection to an existing macro eNB). As the deployment of additional macrocells is not practically possible due to paucity of space and prohibitive costs, a network of macro as well as small cells, termed as Heterogeneous Network or HetNet [12], will assist in a long way in handling the massive numbers of users. Small cells bring in the advantages of load balancing between cells, improved coverage leading to improved user satisfaction, and reduced latency and lower power consumption as the base stations are closer to the users. A HetNet in LTE-A can consist of macrocells, microcells, picocells, and femtocell, in the decreasing order of the base station power. While femtocells cater to only indoor traffic, the others mostly cater to outdoor traffic. Another small cell termed mobile femtocell (MFemtocell) [13] combines the concepts of femtocell with mobile relay. It consists of dedicated relay nodes mounted outside public vehicles like buses, trains, and so forth. The users inside the vehicle form a femtocell which uses the mobile relay node to communicate with the macro base station. The base station views the MFemtocell and all its users as a single entity and at the same time, the users are unaware of the existence of the mobile relay node. The deployment of MFemtocell will prove to be beneficial for high mobility scenarios and also improve network coverage.

In addition to the above which operate in the licensed spectrum, the network can also incorporate Capillary Networks of MTCDs where the devices also use some local communication technologies like Bluetooth, ZigBee, WiFi, and so forth in the unlicensed spectrum. The devices in the network communicate with MTC Servers via a MTC Gateway Device that acts as the intermediary between the cellular network and the capillary network of devices. The MTC Gateway Device is equipped with dual communication capabilities like cellular and some local access technology [14].

The small cells have their share of challenges to overcome. These include cost effective deployment and efficient operation and management of small cells by operators, intercell interference management, technologies for backhauling traffic from small cells to core networks, security, and many more. The authors in [15] have listed some of the issues related to use of capillary M2M networks, which includes interference and low data rates for applications with high end-to-end reliability requirement.

Considering the fact that small cells are expected to be an integral component of future mobile cellular networks, the proposed hierarchical group based authentication protocol uses an architecture which brings together a macrocells network with an overlay of small cells and capillary networks.

3. Motivation

While MTC applications span across a wide domain, each with distinct features and requirements, the common feature distinguishing them from the regular H2H communication is the presence of massive number of devices. As these devices have to execute EPS-AKA to connect to the EPC for access authentication, a large number of these trying to connect at the same time will create substantial signaling overload in the access network. In a roaming scenario, where the devices are away from their home network (HN), the serving network (SN) will have to fetch the authentication data from the HN leading to an increased signaling load. Consequently, this necessitates the design of Mutual Authentication and Key

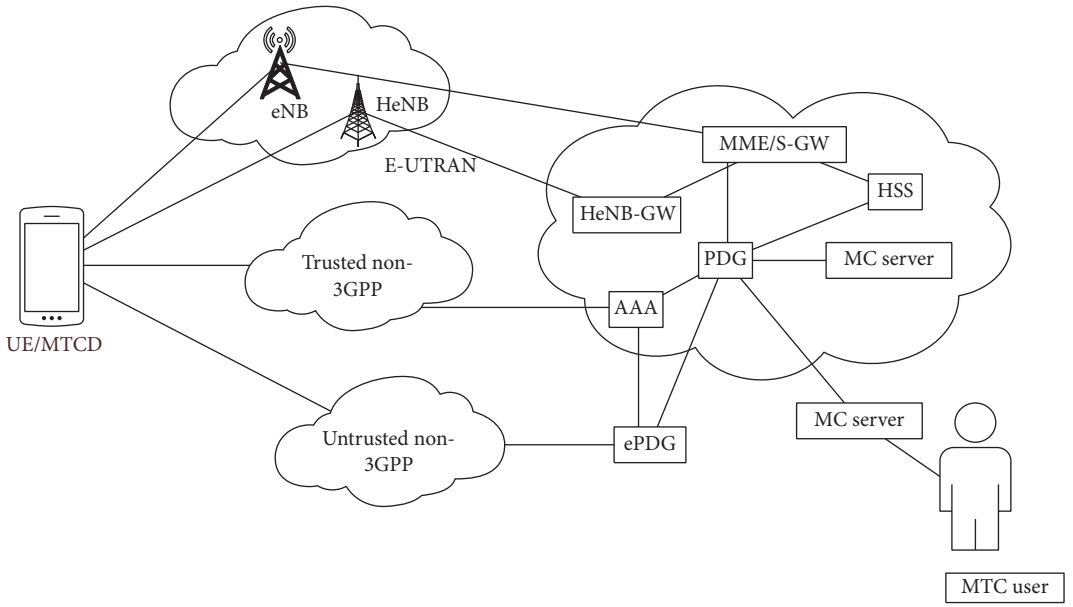

Figure 1: LTE-A network architecture.

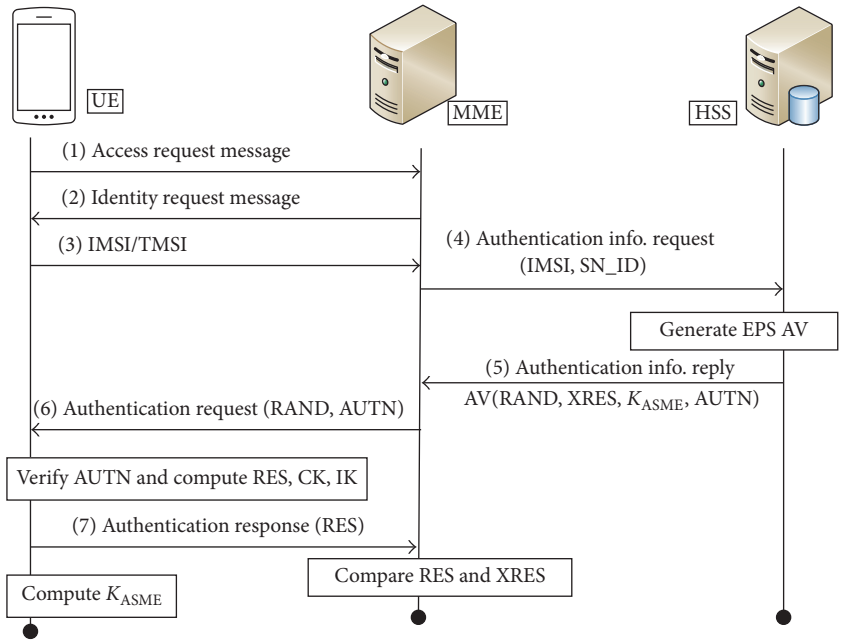

Figure 2: EPS-AKA protocol.

Agreement protocols for MTC aimed at minimizing the signaling overhead.

In addition, the MTCDs are usually low powered devices lacking high computational capabilities. This calls for an authentication protocol of lower computational complexity. Furthermore, an additional issue with cellular MTC is the presence of the devices at cell edges and in indoor spaces like inside underground parking lots, shopping malls, hospitals, and so forth, where network connectivity may be poor. Likewise, some of the devices may not exhibit any form of mobility (e.g., security camera) while others may be highly mobile (e.g., devices in trains or buses). Thus, the authentication procedure needs to take into consideration the following factors:

(i) Massive deployment of MTCDs causing signaling overload

(ii) Deployment environment of the MTCDs (i.e., indoor/outdoor)

(iii) Mobility behavior of the MTCDs (i.e., stationary/mobile)

(iv) Computational and storage capabilities of MTCDs

This paper thus proposes an Authentication and Key Agreement protocol to reduce the signaling load, while addressing the aforementioned factors. The first three factors are accounted for in the deployment architecture while the last factor is handled by the use of symmetric cryptography in the proposed algorithm.

4. Related Works

A number of group based mutual authentication schemes exist in the literature. The schemes can be categorized into three types:

(i) Schemes consist of one device of the group performing a full AKA with the core network and thereafter the authentication records for all members of the group are retrieved from the HN and sent to the SN. For the second group member onwards, the HN remains offline with the authentication being performed locally at the SN with the prefetched data. Some of the schemes following this approach are SE-AKA [16] by Lai et al., MTC-AKA [17] also by Lai et al., DGBAKA [18] by Zhang et al., and GAKA [19] by Chen et al. with some minor variations.

While these schemes successfully reduce the signaling traffic between the home network and the serving network, the load at the access network remains unchanged.

(ii) Schemes use aggregation of messages where a selected group leader from among the MTCDs receives individual messages from all group members, aggregates them into a single message along with an aggregate signature or an aggregate MAC, and forwards the same to the core network. The core network element verifies the aggregate signature or aggregate MAC to authenticate the group. Schemes like Cao et al. in [20, 21] uses the aggregate signature concept and those [22] by Lai et al., [23] by Choi et al., and GLARM [24] by Lai et al. are based on the aggregate MAC concept. While the scheme [20] is based on the concept of aggregate signature proposed by Boneh et al. [25], the scheme [21] uses the Nyberg Rueppel signature scheme [26].

The aggregate MAC approach for group authentication is a lightweight one, significantly reducing the signaling overhead at the access network level as compared to aggregate signature schemes. Most existing schemes use a group leader to aggregate the MACs received from the individual MTCDs in a group. However, this can cause a bottleneck at the group leader and also involves the additional complexity and delay associated with group leader selection and management. Another drawback of the aggregation approach is that a single bad (message, MAC) pair or a single invalid signature in the aggregate can cause the entire authentication to fail resulting in repeating the entire authentication process. An attacker only needs to change a single bit in a message or MAC across the group to ensure a failure. So, the cost of this endeavor from the attacker's point of view is very low, making the system susceptible to Denial of Service attacks.

(iii) Schemes follow a batch processing approach whereby authentication requests from multiple devices are processed at once as a batch, like ABAKA [27] by Huang et al.

These schemes are also hounded by the issue of a single invalid signature requiring the rerun of the entire authentication process.

5. HGMAKA Protocol: System Architecture

The proposed system architecture consists of macrocells, femtocells, mobile femtocells, and also capillary network of devices, thereby forming a HetNet. The architectural elements are classified into three tiers, each consisting of heterogeneous devices performing distinct activities:

(i) The first tier, Tier 1, consists of the data generating/consuming MTCDs, for example, smart meters and medical equipment. They generate data periodically or on being triggered and send the data to some MTC Server. The MTCDs may be static (e.g., surveillance camera) or may be mobile (devices inside train, bus, etc.). They need to be authenticated before they can use the network for sending data to the MTC Servers. Thus, they generate authentication request messages which are sent to the Tier 2 elements.

(ii) The second tier, Tier 2, comprises aggregating elements, those which aggregate the multiple streams of data received from elements in the first tier into a single stream and forward it to the elements in the next tier. Additionally, they verify the MACs of the received messages to verify their integrity and ensure that no tampered messages are included in the aggregate. These elements can be HeNBs, mobile femtocells, MTC Gateway devices, and so forth.

(iii) The third tier, Tier 3, elements provide the last mile connectivity to the core network. They accept the multiple incoming aggregated streams from Tier 2 elements and forward them to the core network either as individual streams or as an aggregate stream as per the requirements. They are also the second level aggregator and verifier. Like the Tier 2 elements, they also verify the integrity of the incoming messages from Tier 2 using MAC before including them in the aggregate. These elements include Home eNodeB Gateway, if HeNBs were the Tier 2 elements, and eNodeB if the corresponding Tier 2 elements were MFemtocells or MTC Gateway devices.

The proposed protocol thus not only performs the aggregation of multiple streams from a lower tier into a single stream at a higher tier, but also eliminates the possibility, due to presence of tampered messages in the aggregate, of authentication failure of the entire group. The Tiers 2 and 3 elements take up the role of aggregation from the group leaders of aggregate based schemes in the literature, thus eliminating

the requirement of group leader and the associated computational overheads. Moreover, the limited communication range of group leader due to the use of local access technology at the group level restricts the group sizes. The use of hierarchical architecture also removes this limitation, thereby allowing larger groups of MTCDs to spread across a macrocell. Figure 3 shows the network architecture while Figure 4 illustrates the advantage of hierarchical architecture in terms of group sizes. Table 1 lists the classification of the architectural elements.

6. HGMAKA Protocol: Algorithm

The aggregation of authentication message at various hierarchies in HGMAKA protocol is followed by a key agreement phase resulting in a unique secret session key being shared among an individual MTCD and the corresponding SN.

The grouping of MTCDs in this protocol may be predetermined or may be done on the fly. For example, a group of MTCDs belonging to the same MTC owner or using the same MTC application may be grouped together by the operator and assigned a group key. Alternately, a group can also be created on the fly based on, say, the current location of the MTCDs, like group of MTCDs inside a vehicle/train/bus and so forth. In this scenario, a group key agreement protocol must be used to generate a group key for all group members with the HSS involved in the process. The group key can also be changed dynamically as changes in membership occur in the group. While there are various group key management protocols available in the literature [28–30] and so forth, the same is beyond the scope of the paper and hence not discussed here. The group key is shared among the MTCDs, HSS, the Tier 1, the Tier 2, and the Tier 3 elements. Notations explain the symbols used in the protocol. The proposed protocol consists of two phases:

(i) *Aggregate generation phase*, where the authentication request messages and MACs, sent by Tier 1 elements, are aggregated at higher levels, first by Tier 2 elements and followed by Tier 3 elements. Integrity verification of the incoming messages is performed at each level prior to the aggregate generation.

(ii) *Group based Mutual Authentication and Key Agreement phase*, where the HSS authenticates all group members simultaneously from the aggregate MAC and the MTCDs authenticate the MME and HSS through a random challenge response protocol terminating with the derivation of unique shared secret keys between the MME and each MTCD.

6.1. Aggregate Generation Phase

(1) Consider a group Gi where the jth MTCD $\mathrm{MTCD}_{Gi\text{-}j}$, identified by its Unique International Mobile Subscriber Identity $\mathrm{IMSI}_{Gi\text{-}j}$, shares a secret key $K_{Gi\text{-}j}$ with the HSS. In this phase, $\mathrm{MTCD}_{Gi\text{-}j}$ generates the authentication request message ($M_{Gi\text{-}j}{}^{1}$) containing the IMSI ($\mathrm{IMSI}_{Gi\text{-}j}$), Group Identifier ($\mathrm{GID}_{Gi}$), random number ($R_{Gi\text{-}j}$), and a MAC ($\mathrm{MAC}_{Gi\text{-}j}$) computed on the same using the secret key $K_{Gi\text{-}j}$ and sends it to the Tier 2 element. Also included in this communication is another MAC ($\mathrm{MAC}_{Gi\text{-}j}{}^{1}$) computed on $M_{Gi\text{-}j}{}^{1}$ and $\mathrm{MAC}_{Gi\text{-}j}$ using the group key GK_{Gi}. While the former is used for authentication of the MTCD by the HSS, the latter is used for verifying the integrity of the message by the Tier 2 element. The detailed steps are listed below:

(a) Each MTCD j of group Gi generates an authentication message $M_{Gi\text{-}j}{}^{1}$ which contains MTCD identity ($\mathrm{IMSI}_{Gi\text{-}i}$), group identifier ($\mathrm{GID}_{Gi}$), a random number $R_{Gi\text{-}i}$ generated by the $\mathrm{MTCD}_{Gi\text{-}j}$, and a MAC, $\mathrm{MAC}_{Gi\text{-}j}$. This MAC, later aggregated by the Tier 2 elements, is used by the HSS to authenticate the MTCD.

$$M_{Gi\text{-}j}{}^{1} = \mathrm{IMSI}_{Gi\text{-}j} \parallel R_{Gi\text{-}j} \parallel \mathrm{GID}_{Gi} \parallel \mathrm{MAC}_{Gi\text{-}j}, \quad (1)$$

where $\mathrm{MAC}_{Gi\text{-}j} = f2(K_{Gi\text{-}j}, \mathrm{IMSI}_{Gi\text{-}j} \parallel R_{Gi\text{-}j} \parallel \mathrm{GID}_{Gi})$.

(b) The MTCD computes a second MAC on $M_{Gi\text{-}j}{}^{1}$ with the group key GK_{Gi}, to be used for integrity verification of these Tier 1 messages at the Tier 2 element,

$$\mathrm{MAC}_{Gi\text{-}j}{}^{1} = f2\left(\mathrm{GK}_{Gi}, M_{Gi\text{-}j}{}^{1}\right). \quad (2)$$

(c) $\mathrm{MTCD}_{Gi\text{-}j}$ sends $M_{Gi\text{-}j}{}^{1} \parallel \mathrm{MAC}_{Gi\text{-}j}{}^{1}$ to the Tier 2 element.

(2) On receiving the message $M_{Gi\text{-}j}{}^{1} \parallel \mathrm{MAC}_{Gi\text{-}j}{}^{1}$, the Tier 2 element of group Gi first verifies the integrity of the received message from $\mathrm{MAC}_{Gi\text{-}j}{}^{1}$ using the shared group key GK_{Gi}. On successful verification, it proceeds to compute the aggregate MAC ($\mathrm{aggMAC}_{Gi}{}^{k2}$) and aggregate message ($\mathrm{agg}M_{Gi}{}^{k2}$) for all Tier 1 elements under it. Finally, the Tier 2 element computes a MAC, $\mathrm{MAC}_{Gi}{}^{k2}$, on $\mathrm{agg}M_{Gi}{}^{k2} \parallel \mathrm{aggMAC}_{Gi}{}^{k2}$ to be later used for the integrity verification of the message at the Tier 3 element. The steps are listed below:

(a) For each Tier 1 element j of group Gi under the Tier 2 element k, the Tier 2 element verifies $\mathrm{MAC}_{Gi\text{-}j}{}^{1}$ in the received message $M_{Gi\text{-}j}{}^{1} \parallel \mathrm{MAC}_{Gi\text{-}j}{}^{1}$ using (2). Successful verification assures that the message $M_{Gi\text{-}j}{}^{1}$ has not been tampered with.

(b) The Tier 2 element then computes aggregate MAC, $\mathrm{aggMAC}_{Gi}{}^{k2}$, by performing XOR operation on the individual MACs received from the Tier 1 elements

$$\mathrm{aggMAC}_{Gi}{}^{k2} = \mathrm{aggMAC}_{Gi}{}^{k2} \oplus \mathrm{MAC}_{Gi\text{-}j}. \quad (3)$$

(c) Aggregate message $\mathrm{agg}M_{Gi}{}^{k2}$ is compiled by concatenating individual messages received from Tier 1 elements

$$M_{Gi\text{-}j}{}^{1} = \mathrm{remove}\left(M_{Gi\text{-}j}{}^{1}, \mathrm{MAC}_{Gi\text{-}j}\right). \quad (4)$$

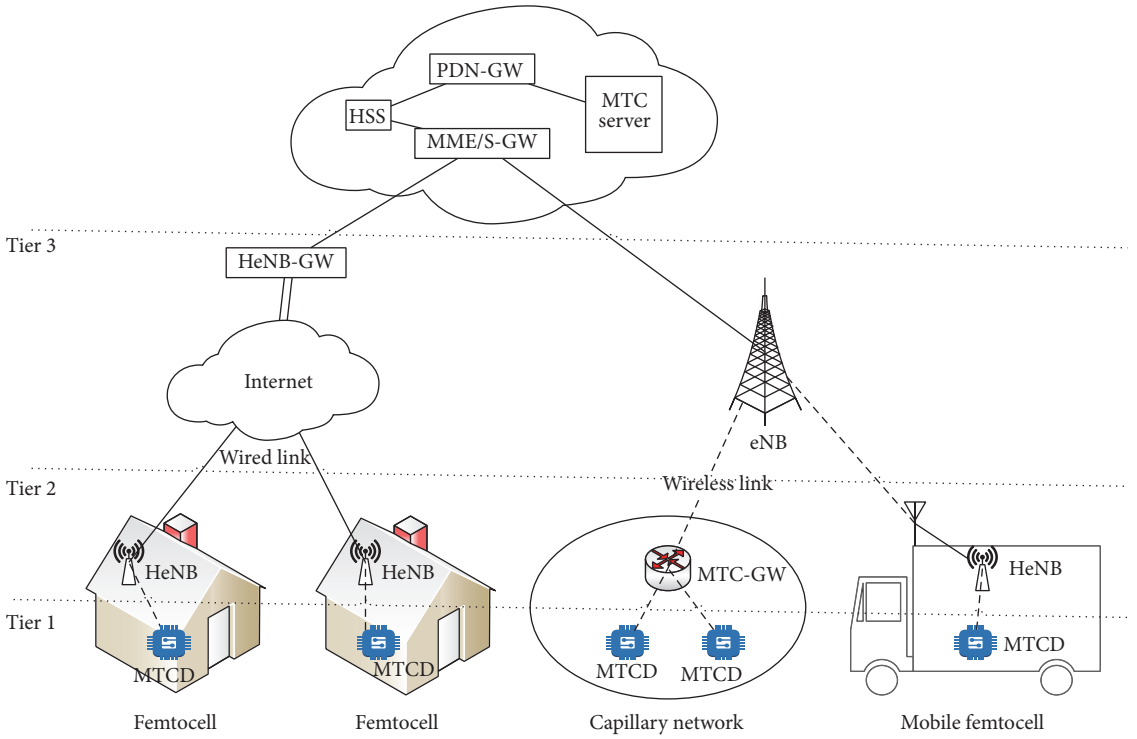

Figure 3: Proposed MTC network architecture.

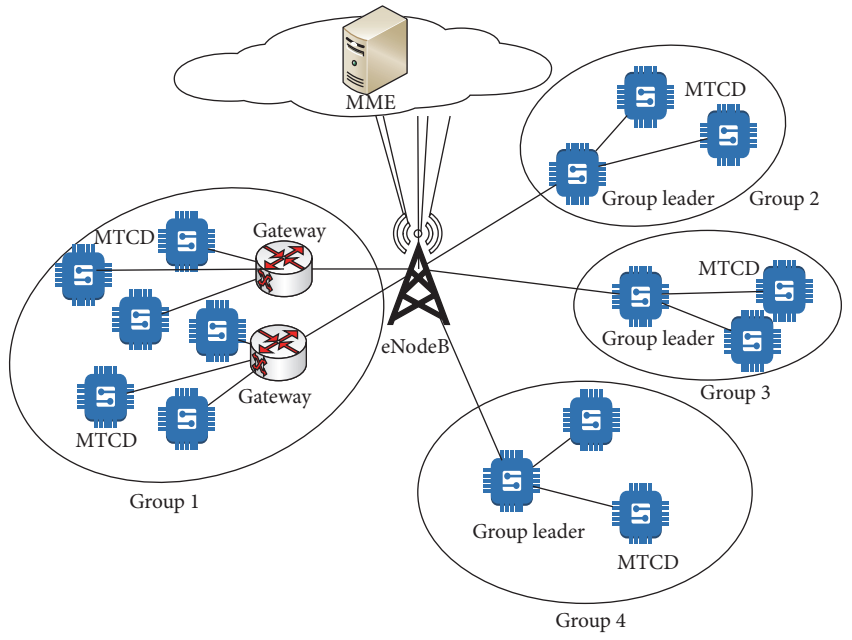

Figure 4: Larger groups enabled by hierarchical architecture.

Table 1: Network elements.

Tier	Description	Devices
Tier 1	The end-device in the communication scenario which generates the authentication requests	MTCD
Tier 2	Level 1 aggregator and integrity verifier of authentication requests from Tier 1	HeNB, MTC-GW, mobile femtocell
Tier 3	Last mile connecting device and Level 2 aggregator and integrity verifier of Tier 2 messages	HeNB-GW, eNodeB

The remove function excluded the MAC field from individual messages.

$$\text{agg}M_{\text{G}i}{}^{k2} = \text{agg}M_{\text{G}i}{}^{k2} \parallel M_{\text{G}i\text{-}j}{}^{1}. \quad (5)$$

(d) $\text{MAC}_{\text{G}i}{}^{k2}$ is then computed for integrity verification of message at the next higher Tier element

$$\text{MAC}_{\text{G}i}{}^{k2} = f2\left(\text{GK}_{\text{G}i}, \text{agg}M_{\text{G}i}{}^{k2} \parallel \text{aggMAC}_{\text{G}i}{}^{k2}\right). \quad (6)$$

(e) Finally, the first level aggregate message $\text{agg}M_{\text{G}i}{}^{k2} \parallel \text{aggMAC}_{\text{G}i}{}^{k2} \parallel \text{MAC}_{\text{G}i}{}^{k2}$ is sent to Tier 3 element.

(3) The Tier 3 element performs integrity verification of the message received from Tier 2 and then the second level aggregation is as follows:

(a) For each Tier 2 element of group Gi under the Tier 3 element, the Tier 3 element verifies $\text{MAC}_{\text{G}i}{}^{k2}$ in the received aggregate message $\text{agg}M_{\text{G}i}{}^{k2} \parallel \text{aggMAC}_{\text{G}i}{}^{k2} \parallel \text{MAC}_{\text{G}i}{}^{k2}$ using (6).

(b) On successful verification, an aggregate MAC, $\text{aggMAC}_{\text{G}i}$, is computed from the first level aggregate MACs, to be used by the HSS for authenticating the entire group Gi

$$\text{aggMAC}_{\text{G}i} = \text{aggMAC}_{\text{G}i} \oplus \text{aggMAC}_{\text{G}i}{}^{k2}. \quad (7)$$

(c) The aggregate message $\text{agg}M_{\text{G}i}$ is compiled by concatenating the level one aggregate messages of group Gi:

$$\begin{aligned} \text{agg}M_{\text{G}i\text{-}j}{}^{k2} &= \text{remove}\left(\text{agg}M_{\text{G}i\text{-}j}{}^{k2}, \text{aggMAC}_{\text{G}i}{}^{k2}\right) \\ \text{agg}M_{\text{G}i} &= \text{agg}M_{\text{G}i} \parallel \text{agg}M_{\text{G}i}{}^{k2}. \end{aligned} \quad (8)$$

(d) The second level aggregate message $\text{agg}M_{\text{G}i} \parallel \text{aggMAC}_{\text{G}i}$ is then sent to MME.

6.2. Group Authentication and Key Agreement Phase

(1) On receiving the aggregate authentication request and aggregate MAC, that is, $\text{agg}M_{\text{G}i} \parallel \text{aggMAC}_{\text{G}i}$ from the Tier 3 element, the MME forwards this, along with its Serving Network Identity (SN_ID), that is, $\text{agg}M_{\text{G}i} \parallel \text{aggMAC}_{\text{G}i} \parallel \text{SN\_ID}$, to the HSS.

(2) The HSS receives this aggregate message authentication request for group Gi from the MME and authenticates the entire group from the aggregate MAC:

(a) The HSS verifies the aggregate MAC, $\text{aggMAC}_{\text{G}i}$ (see (7)).

(b) The HSS uses a random number R_{HSS} to compute a Temporary Group Key, $\text{TGK}_{\text{G}i}$,

$$\text{TGK}_{\text{G}i} = f1\left(\text{GK}_{\text{G}i}, R_{\text{HSS}} \parallel \text{SN\_ID}\right) \quad (9)$$

and generates an authentication token AUTH_{HSS}, to be used by the MTCD for authenticating the HSS, as

$$\text{MAC}^{\text{HSS}}{}_{\text{G}i} = f2\left(\text{TGK}_{\text{G}i}, R_{\text{HSS}} \parallel \text{GID}_{\text{G}i}\right) \quad (10)$$

$$\text{AUTH}_{\text{HSS}} = R_{\text{HSS}} \parallel \text{MAC}^{\text{HSS}}{}_{\text{G}i}. \quad (11)$$

(c) Next, the HSS generates individual session keys ($K^{\text{ASME}}{}_{\text{G}i\text{-}j}$) for each MTCD in the group as well as the expected response XRES to the random challenge that the HSS sends to the MTCDs for authentication purposes via MME.

For each n member j of group Gi
For $j = 1$ to n
$\text{AK}_{\text{G}i\text{-}j} = f5(K_{\text{G}i\text{-}j}, R_{\text{HSS}})$
$\text{CK}_{\text{G}i\text{-}j} = f3(K_{\text{G}i\text{-}j}, R_{\text{HSS}})$
$\text{IK}_{\text{G}i\text{-}j} = f4(K_{\text{G}i\text{-}j}, R_{\text{HSS}})$
$K^{\text{ASME}}{}_{\text{G}i\text{-}j} = \text{KDF}(\text{SQN} \oplus \text{AK}_{\text{G}i\text{-}j}, \text{CK}_{\text{G}i\text{-}j}, \text{IK}_{\text{G}i\text{-}j}, \text{SN\_ID})$
$\text{XRES}_{\text{G}i\text{-}j} = f6(K_{\text{G}i\text{-}j}, R_{\text{HSS}})$
End For

The HSS compiles the authentication information together with the computed K_{ASME} and XRES for all members of the group in the form of a table called Group Key Index (GKI). The GKI for group Gi is constructed as seen in Table 2.

(d) The HSS sends the GKI and AUTH_{HSS} (from (11)) to the MME.

(2) The MME receives these from the HSS and saves the GKI. It forwards AUTH_{HSS} to each MTCD of group $\text{GID}_{\text{G}i}$.

(3) At each MTCD, on receiving the authentication challenge, one has the following:

(a) The MTCDs authenticate the HSS from this challenge by verifying $\text{MAC}^{\text{HSS}}{}_{\text{G}i}$ after computing $\text{TGK}_{\text{G}i}$ (using (9), (10)).

(b) Each MTCD j in the group also computes a response ($\text{RES}_{\text{G}i\text{-}j}$) to the challenge given by the HSS

$$\text{RES}_{\text{G}i\text{-}j} = f6\left(K_{\text{G}i\text{-}j}, R_{\text{HSS}}\right). \quad (12)$$

The group members send their response to the Tier 2 element who aggregates the response in the same way as for MAC. The aggregate RES is sent to the Tier 3 element that again aggregates the response and sends it as an aggregate response $\text{aggRES}_{\text{G}i}$ for the group. The procedure is the same as aggregate MAC generation.

(4) The MME receives the aggregate response ($\text{aggRES}_{\text{G}i}$) and performs the authentication as follows:

(a) The MME verifies if $\text{aggRES}_{\text{G}i} == \text{XRES}_{\text{G}i\text{-}1}$ xor $\text{XRES}_{\text{G}i\text{-}2}$ xor .....

Table 2: Group Key Index.

Group identifier	Group member IMSI	K_{ASME}	XRES
GID_{Gi}	$IMSI_{Gi\text{-}1}$	$K^{ASME}_{Gi\text{-}1}$	$XRES_{Gi\text{-}1}$
	$IMSI_{Gi\text{-}2}$	$K^{ASME}_{Gi\text{-}2}$	$XRES_{Gi\text{-}2}$
	⋮	⋮	⋮
	$IMSI_{Gi\text{-}n}$	$K^{ASME}_{Gi\text{-}n}$	$XRES_{Gi\text{-}n}$

(b) If the verification is successful, the MTCDs in the group are authenticated. The MTCDs generate their individual K_{ASME}s as

$$K^{ASME}_{Gi\text{-}j} = \mathrm{KDF}\left(\mathrm{SQN} \oplus \mathrm{AK}_{Gi\text{-}j}, \mathrm{CK}_{Gi\text{-}j}, \mathrm{IK}_{Gi\text{-}j}, \mathrm{SN\_ID}\right). \quad (13)$$

Thus, at the end of the Authentication and Key Agreement phase, a shared secret key K^{ASME}_{Gi} is shared between each MTCD and the MME. Figure 5 shows a working example of the aggregate generation phase and Figure 6 shows the message sequence in the group Authentication and Key Agreement phase.

7. Security Analysis

To illustrate that the proposed protocol is secure, the authors have followed a three-pronged approach to security analysis. First, an informal security analysis of the protocol has been performed to demonstrate the resilience of the protocol against different protocol attacks. Second, the automated formal verification tool AVISPA has been used to verify if the security goals of mutual authentication and secure key agreement have been met. And third, a provable security approach has been used to formally prove that no feasible polynomial time adversary exists which can break the security of the scheme.

7.1. Mutual Authentication. The proposed protocol is based on the aggregate MAC concept. An aggregate MAC ($aggMAC_{Gi}$) is generated at two levels, Level 1 by Tier 2 elements and Level 2 by Tier 3, from the individual MACs sent by the MTCDs in the group. This is sent to the MME which forwards it to the HSS. The HSS verifies this aggregate MAC and generates a random challenge and forwards it to the MME in the form of an authentication token ($AUTH_{HSS}$) along with the expected response (XRES) for each MTCD in the group. The MME broadcasts the same to all MTCDs. All MTCDs authenticate the HSS from $AUTH_{HSS}$. The MTCD computes their individual response to the random challenge and sends it via the Tier 2 and Tier 3 elements after going through two levels of aggregation to arrive at the aggregate response ($aggRES_{Gi}$). The MME then completes the mutual authentication by comparing the received aggregate response to the precomputed responses stored at the MME. Thus, the MME and MTCDs perform a mutual authentication.

7.2. Secured Key Agreement. Once mutual authentication is successful, both parties, that is, MME and MTCD, have a session key shared between them. The key ($K^{ASME}_{Gi\text{-}j}$) is generated at both ends after mutual authentication and is never transmitted over any channel. Also, the respective keys are generated at both ends as KDF(SQN $\oplus$ $AK_{Gi\text{-}j}$, $CK_{Gi\text{-}j}$, $IK_{Gi\text{-}j}$, SN_ID), where $AK_{Gi\text{-}j}$, $CK_{Gi\text{-}j}$, and $IK_{Gi\text{-}j}$ are computed from R_{HSS} and the shared secret key $K_{Gi\text{-}j}$ only after mutual authentication is successful.

7.3. Man-in-the-Middle Attack. Despite the fact that the entire communication, both within the capillary network/femtocell and between the MTC-GW and the network, occurs in plaintext, an intruder cannot determine the session keys (TGK_{Gi} and $K^{ASME}_{Gi\text{-}j}$) as these require the knowledge of long term secret keys shared between the MTCD and the HSS as well as the random numbers and sequence numbers.

7.4. Replay Attack. Replay attacks can be thwarted by the use of the random numbers generated by the MTCDs and HSS in the authentication procedure. Even if an attacker captures the messages, the same random number cannot be used for another round of authentication as fresh numbers are required for a fresh round of authentication.

7.5. Key Compromise Impersonation (KCI). A key agreement protocol is KCI-resilient if no adversary can masquerade as an honest user and establish a session key with another user whose long term key has been compromised by the adversary. The proposed protocol is KCI-resilient as proved formally in the next section.

7.6. Forward Secrecy. Forward secrecy implies that if the long term key of the entity is compromised then the secrecy of the earlier session keys generated is not affected. In case of HGMAKA, the session key $K^{ASME}_{Gi\text{-}j}$ is computed as a function of the long term secret key $K_{Gi\text{-}j}$ as well as R_{HSS} and SQN, both of which changes for each session. Thus, even if $K_{Gi\text{-}j}$ is compromised, it will be difficult for the attacker to correlate both the correct R_{HSS} and SQN to be used for generating a past session key. Hence, the protocol has forward key secrecy.

7.7. Formal Verification Using AVISPA. The main goal of the proposed protocol is to provide mutual authentication between the MTCD and the MME/HSS and secured key agreement between the entities. The formal verification of the proposed protocol is tested on *Automated Validation of Internet Security Protocols and Applications* (AVISPA) [31]. AVISPA is a tool for automatic security analysis of protocols represented in *High Level Protocol Specification Language* (HLPSL) and evaluated using automatic deduction techniques. We have modeled three roles, MTCD, Gateway, and MME, using HLPSL. As the MME and HSS have a secured channel between them, so we have integrated the roles of MME and HSS into a single role. The On-the-Fly Model Checker has been used to test our protocol. The goals of the

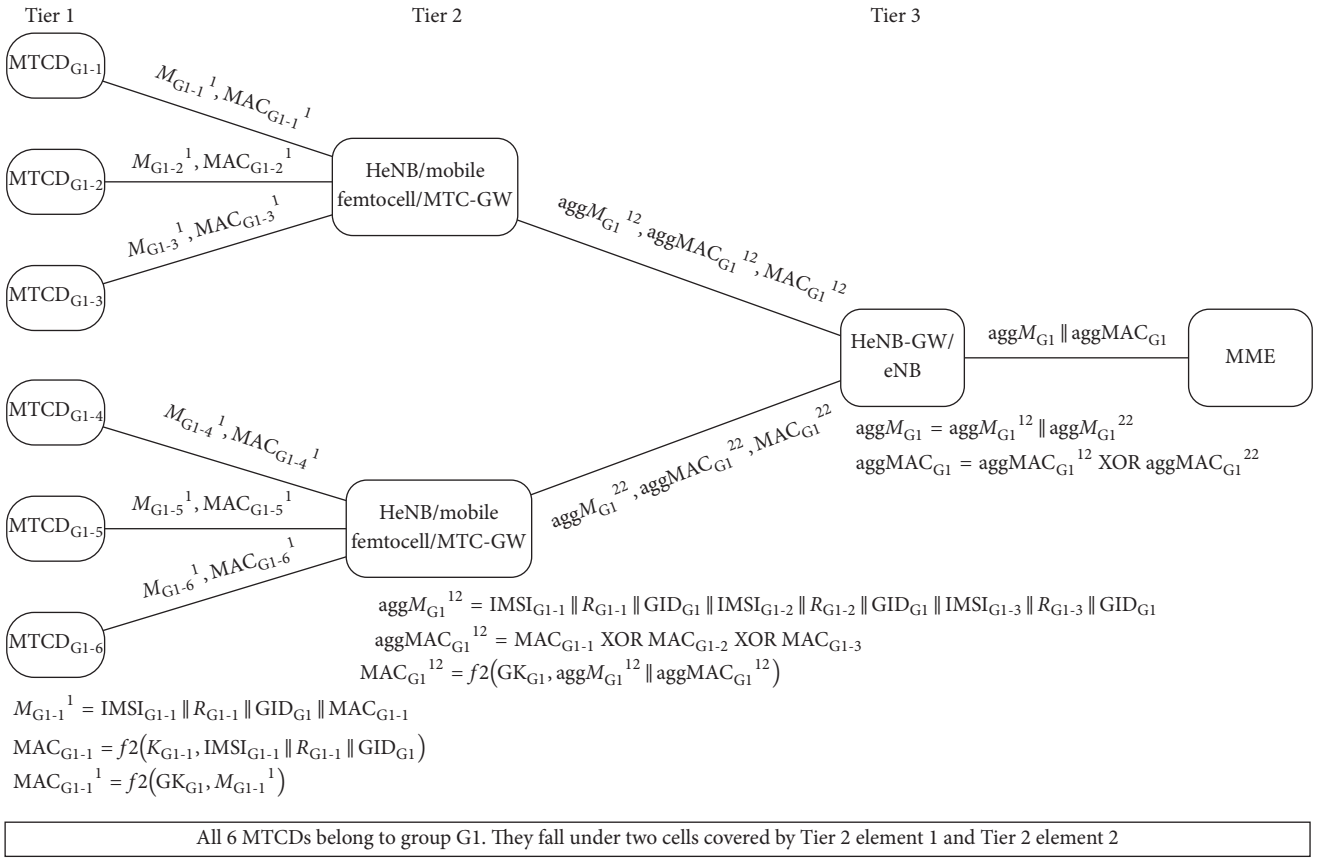

Figure 5: Example working of proposed protocol (aggregate generation).

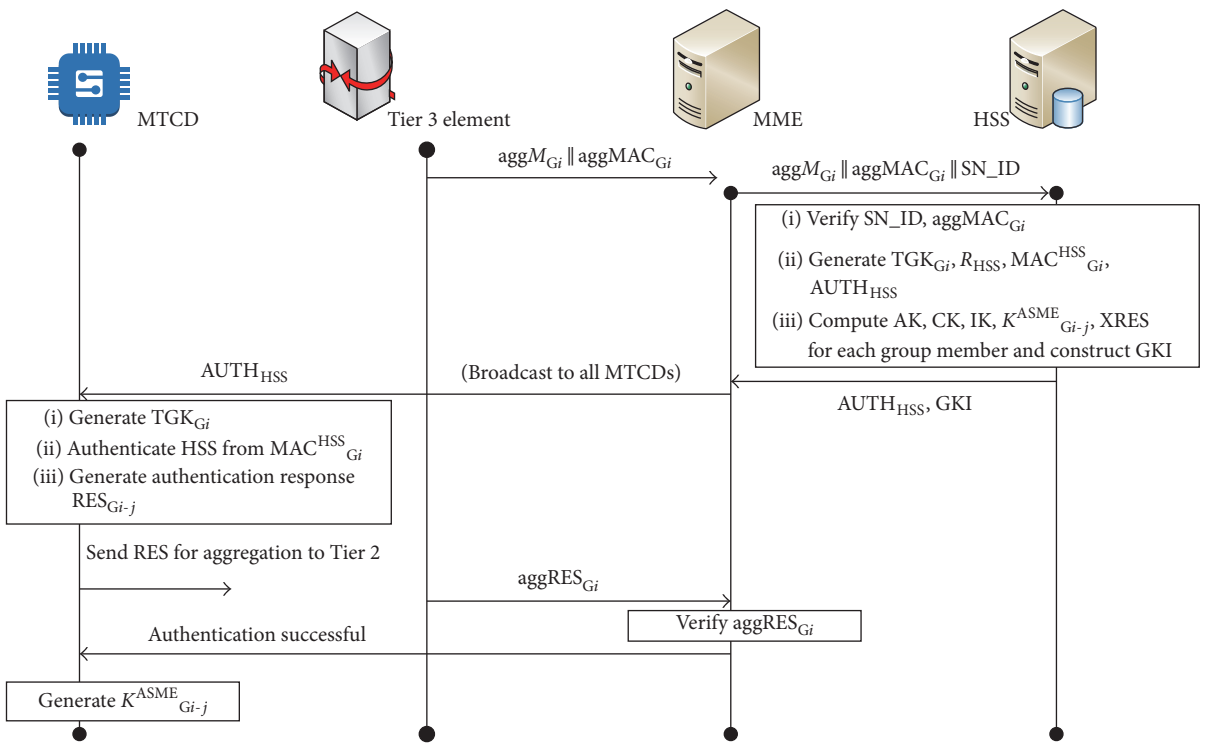

Figure 6: Message exchanges in group Authentication and Key Agreement phase.

```
goal
        secrecy_of kasme
        secrecy_of gtkg1
        authentication_on mtcd_mme_rg11
        authentication_on mtcd_mme_rhss
end goal
```

Algorithm 1: Goal specification of proposed protocol.

protocol are given in Algorithm 1 and the output is given in Algorithm 2. The HLPSL roles for MTCD and MME are given in Appendix A as Algorithms 3 and 4.

7.8. Formal Protocol Analysis. The foundation of the security proof of the proposed protocol lies in Shoup's formal security model [32] for secure Mutual Authentication and Key Agreement using the simulation based technique. In [33] Zhang suitably modified Shoup's model to fit the mobile communication scenario. Further, Huang et al. in [34, 35] also used Zhang's model to present a provably secure AKA protocol for UMTS. The authors have based their security proof on Zhang's model. In Zhang's simulation based model, the proof of security is arrived at by comparing the performance of an adversary in a real protocol execution (real world) and an ideal scenario (ideal world) which is assumed to be secure by definition. The protocol is said to be provably secure if it is not possible to distinguish between the actions of the attacker in the real world and the ideal world.

As per Shoup's model, the MTC scenario involves two types of communication channels: a secure (assumed) channel between network entities and an insecure wireless channel between the MTCD entities and the network entities. The latter is under the full control of the adversary who can read, modify, and replay the messages transmitted over the channel. The adversary can set up and initiate multiple sessions between MTCD and network and also acquire session keys and apply it on some mathematical function. The activities of the adversary are captured under two scenarios: ideal world and real world. The security of the proposed protocol is proved by comparing the actions of the adversary in the two worlds and proving that the adversary cannot do any more harm in the real world than it would do in the ideal world. Appendix B contains the descriptions of the ideal and real world as well as certain preliminary definitions.

Security Proof. Each entity in the proposed protocol possesses a random number generator which produces random numbers, $R_{\mathrm{G}i\text{-}j}$ (for all i and j) and R_{HSS}, for the entity instances. Let all random numbers be selected randomly in A and $E1$ is the event that T_A is collision-free; we get

$$\Pr\left(\overline{E1}\right) \leq \frac{n_i^2}{2}\left(2^{-|R_{\mathrm{G}i\text{-}j}|} + 2^{-|R_{\mathrm{HSS}}|}\right), \tag{14}$$

where n_i is the count of instances created by A and $|R_{\mathrm{G}i\text{-}j}|$ and $|R_{\mathrm{HSS}}|$ are lengths of the respective random numbers and are polynomial in k, and then $\Pr(\overline{E1})$ is negligible.

Lemma 1. *For real world adversary A with collision-free (assumed) transcript T_A and independent function family $f2$ assumed to be collision resistant in T_A, the probability of the event $\overline{E2}$ ($E2$ is the event that T_A is authentic) is given by*

$$\Pr\left(\overline{E2}\right) \leq n_i\left(2 \times \mathrm{Adv}_{f2}^{\mathrm{mac}}(p, n)\right), \tag{15}$$

where $p = O(t)$ and $n = O(n_i)$ and t is the execution time of A.

Proof. For T_A to be nonauthentic, we must have at best one instance that has accepted based on a stimulus sent by a noncompatible instance. To prove this we show that upper bound on the probability of T_A being nonauthentic is given by (15) and the proof proceeds as follows.

Case 1. Say, the network entity instance $I_{i'j'}$ accepted on receipt of aggregate message $\mathrm{agg}M_{\mathrm{G}i} \parallel \mathrm{aggMAC}_{\mathrm{G}i}$. We try to prove that the stimulus was sent by a compatible MTCD entity. If $I_{i'j'}$ has accepted, that means that the MAC verification has been successful. However, the identity $\mathrm{IMSI}_{\mathrm{G}i\text{-}j}$ of each MTCD entity instance in the group is used in the computation of MAC. Since the IMSI contains the HN identity embedded in it, if $I_{i'j'}$ accepts then it could only be on stimulus sent by a compatible MTCD instance. Let F be the adversary for Message Authentication Code $f2$ with oracle access to $f2_K$ where K is a randomly selected key. Let $\mathrm{PID}_{i'j'} = \mathrm{IMSI}_{i'j'}$ of user U and U might be initialized by F. F starts its execution by picking keys for all users other than U and proceeds as in the real world. If F needs access to the MAC function $f2_K$ under the key of U, the same is provided by the corresponding oracles. Further, the evaluation of functions $f1$, $f3$, $f4$, $f5$, $f6$, and KDF with U's key is realized through returning a constant or random number by F. Say, $I_{i'j'}$ accepts at some point and F outputs $\mathrm{aggMAC}_{\mathrm{G}i}$ and the message $\mathrm{agg}M_{\mathrm{G}i}$ and stops else F stops with a null string as output.

Let $\mathrm{Succ}(F, f2)$ be the event that F outputs aggMAC and a message which has not been given to the oracle $f2_K$ as a query. Let the event that $I_{i'j'}$ has accepted on stimulus by noncompatible instance be represented by $E_{i'j'}$. If $E_{i'j'} = 1$ then the adversary has been successful in forging the MAC for the given message. Thus,

$$\Pr\left(E_{i'j'} = 1\right) \leq \Pr\left(\mathrm{Succ}\left(F, f2\right) = 1\right) \tag{16}$$

$$\text{hence, } \Pr\left(E_{i'j'} = 1\right) \leq \mathrm{Adv}_{f2}^{\mathrm{mac}}(p, n), \tag{17}$$

where $p = O(t)$ and $n = O(n_i)$.

Case 2. Say, MTCD instance I_{ij} accepted on stimulus $\mathrm{AUTH}_{\mathrm{HSS}}$. Let the event that I_{ij} accepted on stimulus from a nonnetwork entity be denoted by E_{ij} and let the event that I_{ij} accepted on stimulus from a noncompatible network entity $I_{i'j'}$ be denoted by E'_{ij}. If E'_{ij} has occurred then $I_{i'j'}$ must have received $\mathrm{agg}M_{\mathrm{G}i} \parallel \mathrm{aggMAC}_{\mathrm{G}i}$ before sending $\mathrm{AUTH}_{\mathrm{HSS}}$. As T_A is collision-free $R_{\mathrm{G}i\text{-}j}$ could be generated only by I_{ij} which implies that the adversary has successfully forged one or more MACs in aggMAC. Thus,

$$\Pr\left(E'_{ij} = 1\right) \leq \mathrm{Adv}_{f2}^{\mathrm{mac}}(p, n). \tag{18}$$

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/Proposed.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.21s
  visitedNodes: 8 nodes
  depth: 3 plies
```

Algorithm 2: Results from OFMC.

```
role mtcd(MTCD,GW,MME: agent,
          Kg1_1,Gkg1: symmetric_key,
          IMSIg1_1, GIDg1:text,
          HMAC: function,
          F1,F2,F3,F4,F5,F6,KDF:function,
          SQN:text,
          SND,RCV,SNDG,RCVG:channel(dy))
played_by MTCD def=
local State:nat,
  Rg1_1:text,
  MACg1_1:text,
  GMACg1_1:text,
  Authhss,Rmme,Rhss,Macmme:text,
  SNID:text
init State:=0
transition
(1)  State=0 ∧ RCV(start)=|>
      State':=3 ∧ Rg1_1':=new() ∧ MACg1_1':= {HMAC(IMSIg1_1.Rg1_1')}_Gkg1 ∧ GMACg1_1':=
{HMAC(IMSIg1_1.Rg1_1'.MACg1_1'.GIDg1)}_Gkg1 ∧ SNDG(IMSIg1_1.Rg1_1'.MACg1_1'.GIDg1.GMACg1_1') ∧
witness(MME,MTCD,mme_mtcd_rg11,Rg1_1')
(2)  State=3 ∧
RCV(Rhss'.({HMAC(Rhss'.GIDg1)}_({F1(Rhss'.SNID)}_Gkg1)).Rmme'.({F2(Rmme'.(Rhss'.({HMAC(Rhss'.GIDg1)}_
({F1(Rhss'.SNID)}_Gkg1))))}_({F1(Rhss'.SNID)}_Gkg1))) =|>
      State':=6 ∧ SNDG({F6(Rhss'.Rg1_1)}_Kg1_1) ∧ request(MTCD,MME,mtcd_mme_rhss,Rhss') ∧
request(MTCD,MME,mtcd_mme_rmme,Rmme')
end role
```

Algorithm 3: Role of MTCD in HLPSL.

If E_{ij} has occurred then the adversary was successful in forging the MAC for the message R_{HSS}. An execution of A results in an adversary F' for $f2$ such that the event of I_{ij} accepting results in F' ends with output $\mathrm{MAC}^{\mathrm{HSS}}{}_{Gi}$ and R_{HSS}. Thus in (18), we have $\Pr(E_{ij} = 1) \leq \Pr(\mathrm{Succ}(F', f2) = 1)$. Hence,

$$\Pr\left(E_{ij} = 1\right) \leq \mathrm{Adv}_{f2}^{\mathrm{mac}}\left(p, n\right). \tag{19}$$

Therefore, the upper bound on the probability that MTCD instance I_{ij} accepted on a stimulus from a network instance noncompatible with it is given by

$$\Pr\left(E_{ij} = 1\right) + \Pr\left(E'_{ij} = 1\right) \leq 2 \times \mathrm{Adv}_{f2}^{\mathrm{mac}}\left(p, n\right). \tag{20}$$

Case 3. Let the network instance that accepted on stimulus aggRES_{Gi} be denoted by $I_{i''j''}$ and $I_{i''j''}$ had earlier sent R_{HSS}. If aggRES_{Gi} was not sent by a MTCD instance then the

```
role mme(MTCD,GW,MME: agent,
         Kg1_1, Gkg1: symmetric_key,
         IMSIg1_1, GIDg1:text,
         HMAC: function,
         F1,F2,F3,F4,F5,F6,KDF:function,
         SQN:text,
         SND,RCV:channel(dy))
played_by MME def=
local
         State:nat,
         SNID,Aggmacg1,Aggres,Rhss, Macg1:text,
         GTKg1,Kasmeg1_1:symmetric_key,
         Authhss:text,
         Akg1_1,Ckg1_1,Ikg1_1,Xresg1_1, Rmme,Rg1_1,Macmme:text,
init
         State:=3
transition
(1)      State=3 ∧ RCV(IMSIg1_1.Rg1_1'.GIDg1.(xor(Aggmacg1,{HMAC(IMSIg1_1.Rg1_1')}_Gkg1)).SNID) =|>
         State':=5 ∧ Rhss':=new() ∧ GTKg1':={F1(Rhss'.SNID)}_Gkg1 ∧ secret(GTKg1',gtkg1,{MME,MTCD}) ∧
Macg1':={HMAC(Rhss'.GIDg1)}_GTKg1' ∧ Authhss':= Rhss'.Macg1' ∧ Akg1_1':= {F5(Rhss')}_Kg1_1 ∧ Ckg1_1':=
{F3(Rhss')}_Kg1_1 ∧ Ikg1_1':= {F4(Rhss')}_Kg1_1 ∧ Kasmeg1_1':=KDF(xor(SQN,Akg1_1'),Ckg1_1',Ikg1_1',SNID) ∧
secret(Kasmeg1_1',kasme,{MTCD,MME}) ∧ Xresg1_1':= {F6(Rhss'.Rg1_1')}_Kg1_1 ∧ Rmme':= new() ∧ Macmme':=
{HMAC(Rmme'.Authhss')}_GTKg1' ∧ SND(Authhss'.Rmme'.Macmme') ∧ witness(MME,MTCD,mtcd_mme_rmme,Rmme) ∧
witness(MME,MTCD,mtcd_mme_rhss,Rhss)
(2)      State=5 ∧ RCV(xor(Aggres,{F6(Rhss.Rg1_1')}_Kg1_1)) =|>
         State':=6 ∧ request(MTCD,MME,mme_mtcd_rg11,Rg1_1')
end role
```

Algorithm 4: Role of MME in HLPSL.

adversary was successful in forging $\text{MAC}^{\text{HSS}}{}_{Gi}$ and it can be proved similar to (17) that the upper bound on this event is $\text{Adv}^{\text{mac}}_{f2}(p,n)$. Further, say, aggRES_{Gi} was output by noncompatible MTCD instance I_{i1j1}. In this case I_{i1j1} received AUTH_{HSS} (which contains R_{HSS}) prior to generating the stimulus. With T_A being collision-free, it is not possible for a network entity, other than $I_{i''j''}$, to output AUTH_{HSS} pointing to an adversary having forged $\text{MAC}^{\text{HSS}}{}_{Gi}$. Similar to (19) we have an upper bound on this event as $\text{Adv}^{\text{mac}}_{f2}(p,n)$. Thus, the upper bound on the probability that the stimulus on $I_{i''j''}$ was from a noncompatible MTCD instance is $2 \times \text{Adv}^{\text{mac}}_{f2}(p,n)$.

We can therefore conclude that the probability of the event $\overline{E2}$ is

$$\Pr\left(\overline{E2}\right) \leq n_i \left(2 \times \text{Adv}^{\text{mac}}_{f2}\left(p,n\right)\right). \tag{21}$$

□

Lemma 2. *Let $f1$, $f3$, $f4$, $f5$, $f6$, and KDF, represented by H, be a family of pseudorandom functions and H is independent of $f2$ and collision resistant in the real world adversary A's authentic and collision-free transcript T_A. The algorithm D that distinguishes between the transcript of the real world adversary A and the ideal world adversary A^* has*

$$\text{Adv}^{\text{dist}}_{T_A,T_{A^*}}(D) \leq n_{\text{MTCD}} \text{Adv}^{\text{prf}}_{H}\left(p,n\right), \tag{22}$$

where A initializes n_{MTCD} MTCD entities and n_i instances. $p = O(t)$ and $n = O(n_i)$.

Proof. A simulator with a real world adversary A creates an ideal world adversary A^* and converts the transcript in the real world (T_A) into a transcript in the ideal world (T_{A^*}) so that the two are almost identical. The conversion is carried out in the following manner:

(i) An implementation record in T_A is copied to T_{A^*} through an implementation operation.

(ii) A (start session, i, j) record in T_A is connection assignment with the ring master replacing the session key $K^{\text{ASME}}{}_{ij}$ with idealized random session key K_{ij}.

(iii) An (abort session, i, j) is copied to T_{A^*} with T_{A^*} executing (abort session, i, j) operation.

(iv) In case of an application action in T_A, all necessary evaluation is made by the ring master using the session key of the ideal world.

The main difference between T_A and T_{A^*} lies in the application records. We prove that the assignments made by A^* are legal and T_A and T_{A^*} are indistinguishable.

Case 1. Let us assume that an MTCD instance I_{ab} receives AUTH_{HSS} and accepts. This message cannot be sent by a noncompatible network entity as T_A is authentic. Let the stimulus on I_{ab} be sent by compatible MTCD entity instance $I_{a'b'}$. The connection assignment (create, a', b') is made by adversary A^*. This connection assignment has not been

made earlier as AUTH_{HSS} contains R_{HSS} which the MTCD can verify as not being repeated. Thus, the ring master can substitute the session key $K^{\text{ASME}}{}_{ab}$ with an idealized (random) session key K_{ab}.

Case 2. Say $I_{p'q'}$, a network instance, receives $\text{agg}M_{\text{G}i} \parallel \text{aggMAC}_{\text{G}i}$. I_{pq} is an MTCD instance whose individual MAC is included in $\text{aggMAC}_{\text{G}i}$ and $I_{p'q'}$ accepts on this stimulus. The connection assignment made by A^* is (create, p, q). The random session key K_{pq} is assigned by the ring master instead of $K^{\text{ASME}}{}_{p'q'}$. Since $f2$ is collision resistant in T_A, hence the received message could be the stimulus only on $I_{p'q'}$. Thus, the connection assignment (create, p, q) has not be made earlier. This can be proved similarly for all MTCD instances included in the group MAC construction.

Case 3. Let $I_{x'y'}$ be the network instance that receives $\text{aggRES}_{\text{G}i}$ from a group Gi and let I_{xy} be a MTCD instance of an MTCD belonging to this group. Assume $I_{x'y'}$ has accepted on this stimulus. Since $f2$ is collision resistant and T_A is collision-free, as in Case 2, we can prove that I_{xy} has accepted on receiving stimulus provided by $I_{x'y'}$. Thus, the adversary A^* has made a valid connection assignment (connect, x, y) with the ring master setting the session key $K^{\text{ASME}}{}_{x'y'}$ to K_{xy}.

Each start session record in T_{A^*} has a connection assignment as seen from the analysis. The main difference between T_A and T_{A^*} now lies only in the application records. For a single MTCD entity initialized by A and algorithm D that distinguishes between T_A and T_{A^*}, an adversary D' for H is such that $\text{Adv}^{\text{dist}}_{T_A,T_{A^*}}(D) = \text{Adv}^{\text{prf}}_H(D')$. Hence, $\text{Adv}^{\text{dist}}_{T_A,T_{A^*}}(D) \leq \text{Adv}^{\text{prf}}_H(p, n)$, where $p = O(t)$ and $n = O(n_i)$.

For n_{MTCD} MTCD entities with keys $K_{\text{G}i\text{-}1}, K_{\text{G}i\text{-}2}, \ldots, K_{\text{G}i\text{-}n\text{MTCD}}$, D and D' have access to oracles $G_{K_{\text{G}i\text{-}1}}, G_{K_{\text{G}i\text{-}2}}, \ldots, G_{K_{\text{G}i\text{-}n\text{MTCD}'}}$. Thus,

$$\text{Adv}^{\text{dist}}_{T_A,T_{A^*}}(D) \leq n_{\text{MTCD}}\text{Adv}^{\text{prf}}_H(p, n). \tag{23}$$

H can be replaced by any of $f1$, $f3$, $f4$, $f5$, $f6$, or KDF. □

Theorem 3. *Assume f1, f3, f4, f5, f6, and KDF, represented by H, are pseudorandom function families and f2 is a secure MAC and all functions are independent of each other. Then, HGMAKA is a secure AKA protocol.*

Proof. For real world adversary A with transcript T_A, the probability that $f2$ is collision resistant is negligible as $f2$ is a secure Message Authentication Code. From Lemma 2, we have a distinguishing algorithm D for ideal world adversary A^* such that

$$\begin{aligned}&\left|\Pr\left(D\left(T_A\right) = 1 \mid E1 \wedge E2\right)\right.\\&\left.\quad - \Pr\left(D\left(T_{A^*}\right) = 1 \mid E1 \wedge E2\right)\right|\\&\quad \leq n_{\text{MTCD}}\text{Adv}^{\text{prf}}_H(p, n).\end{aligned} \tag{24}$$

Therefore,

$$\begin{aligned}\text{Adv}^{\text{dist}}_{T_A,T_{A^*}}(D) &= \left|\Pr\left(D\left(T_A\right) = 1\right) - \Pr\left(D\left(T_{A^*}\right)\right.\right.\\&\quad \left.\left. = 1\right)\right| = \left|\left(\Pr\left(D\left(T_A\right) \mid E1 \wedge E2\right)\right.\right.\\&\quad - \left.\Pr\left(D\left(T_{A^*}\right) = 1 \mid E1 \wedge E2\right)\right)\Pr\left(E1 \wedge E2\right)\\&\quad + \left(\Pr\left(D\left(T_A\right) \mid \overline{E1} \vee \overline{E2}\right)\right.\\&\quad - \left.\left.\Pr\left(D\left(T_{A^*}\right) = 1 \mid \overline{E1} \vee \overline{E2}\right)\right)\Pr\left(\overline{E1} \vee \overline{E2}\right)\right|\\&\leq \left|\Pr\left(D\left(T_A\right) = 1 \mid E1 \wedge E2\right) - \Pr\left(D\left(T_{A^*}\right)\right.\right.\\&\quad \left.\left. = 1 \mid E1 \wedge E2\right)\right| + \Pr\left(\overline{E2}\right) + \Pr\left(\overline{E1}\right)\\&\leq n_{\text{MTCD}}\text{Adv}^{\text{prf}}_H(p, n) + \Pr\left(\overline{E2}\right) + \Pr\left(\overline{E1}\right).\end{aligned} \tag{25}$$

Again,

$$\begin{aligned}\Pr\left(\overline{E2}\right) &= \Pr\left(\overline{E2} \mid E1\right)\Pr\left(E1\right)\\&\quad + \Pr\left(\overline{E2} \mid \overline{E1}\right)\Pr\left(\overline{E1}\right)\\&\leq \Pr\left(\overline{E2} \mid E2\right) + \Pr\left(\overline{E1}\right).\end{aligned} \tag{26}$$

Hence,

$$\begin{aligned}&\text{Adv}^{\text{dist}}_{T_A,T_{A^*}}(D)\\&\quad \leq n_{\text{MTCD}}\text{Adv}^{\text{prf}}_H(p, n) + \Pr\left(\overline{E2} \mid E1\right) + 2\Pr\left(\overline{E1}\right).\end{aligned} \tag{27}$$

The probabilities $\Pr(\overline{E1})$ and $\Pr(\overline{E2} \mid E1)$ are negligible in k (from (14) and Lemma 1); hence $\text{Adv}^{\text{dist}}_{T_A,T_{A^*}}(D)$ is also negligible. Hence it is proved that HGMAKA is a secure AKA protocol. □

Theorem 4. *HGMAKA is KCI-resilient.*

Proof. We consider the case when the long term secret key of a MTCD has been compromised by an adversary. Let I denote a MTCD entity instance who tries to execute the Authentication and Key Agreement protocol with a compatible network entity instance I'. I' accepts with session key $K^{\text{ASME}}{}_{\text{G}i\text{-}j}$ on the stimulus $\text{agg}M_{\text{G}i} \parallel \text{aggMAC}_{\text{G}i}$. Now, the adversary corrupts $\text{MTCD}_{\text{G}i\text{-}j}$ and I' responds with message AUTH_{HSS}. This is intercepted by the adversary and replaced by message $\text{AUTH}_{\text{HSS}}{}'$. Next, I accepts and generates session key $K^{\text{ASME}}{}_{\text{G}i\text{-}j}{}'$ which is the same as the one generated by the adversary and different from $K^{\text{ASME}}{}_{\text{G}i\text{-}j}$ generated by I'. In the ideal world, the session key generated by I' was prior to the adversary corrupting the MTCD entity instance. Hence, the only connection assignment possible is *create*. Further, in case of I since it was corrupted by the adversary, the only possible connection assignment is *compromise*. However, I cannot be compromised without invalidating the rules of the ideal world as $\text{PID}_I = \text{ID}_{I'}$. Also, *connect* is not possible without $K^{\text{ASME}}{}_{\text{G}i\text{-}j} = K^{\text{ASME}}{}_{\text{G}i\text{-}j}{}'$. Thus, the real world transcript will

Table 3: Probability of authentication failure and success.

Group size	100	200	300	400	500	600	700	800	900	1000
Number of groups	100	50	33	25	20	16	14	12	11	10
$\Pr(A)$	0.6358	0.8687	0.9532	0.9835	0.9942	0.9980	0.9993	0.9998	0.9999	0.9999
$1 - \Pr(A)$	0.3642	0.1313	0.0468	0.0165	0.0058	0.0020	0.0007	0.0002	$7.63E{-}05$	$2.51E{-}05$

be different from the ideal world transcript causing the simulation to be impossible. Therefore, we can conclude that HGMAKA is KCI-resilient. □

8. Performance Analysis

A comparative analysis of the proposed HGMAKA protocol vis-à-vis ten others discussed in Section 4 was performed with respect to three different metrics: (a) number of signaling messages exchanged in executing the protocol; (b) communication cost, that is, the amount of data (in number of bits) transferred in executing the protocols; and (c) computational complexity, that is, the time (in milliseconds) taken for executing the cryptographic operations involved in the protocols, by both the MTCD and the network.

An example scenario is given in the annexure. A total population, n, of 10000 MTCDs was considered which may be divided into groups of varying sizes. It is assumed that 1% of this population can generate corrupt authentication requests, represented by n_c; that is, 100 corrupt messages can exist. The presence of even a single corrupt message can cause authentication failure resulting in a reauthentication requirement for the entire group. For a group size of n_b, the probability that exactly i invalid requests out of n_c are present in the group n_b follows the hypergeometric distribution as

$$\Pr\{X = i\} = \frac{\binom{n-n_c}{n_b-i}\binom{n_c}{i}}{\binom{n}{n_b}} \quad i = 0, 1, 2, \ldots, 100. \tag{28}$$

If A represents the event that reverification of the entire group is required, then probability of A is

$$\Pr(A) = \Pr\{i = 1\} + \Pr\{i = 2\} + \cdots + \Pr\{i = 100\}. \tag{29}$$

This computation has been applied for different group sizes that is n_b and the corresponding probabilities are shown in Table 3.

Thus for a group size of 100, there are 0.6358 probabilities that group reauthentication will be required and 0.3642 that group reauthentication will not be required. As the number of groups decreases resulting in increase in the number of MTCDs in each individual group, the probability of reauthentication (due to failure) increases.

These probabilities are taken into account in the performance analysis for the proposed protocol vis-à-vis the existing protocols. Each of the aforementioned metrics was computed for different protocols using the following formula: Quantity of metric required for g groups for successful authentication $= y \cdot \Pr(A) \cdot t + x \cdot (1 - \Pr(A)) \cdot t$, where t is the number of authentication runs, x is the quantity of metric required for authentication of g groups, and y is the quantity of metric required for reauthentication of g groups.

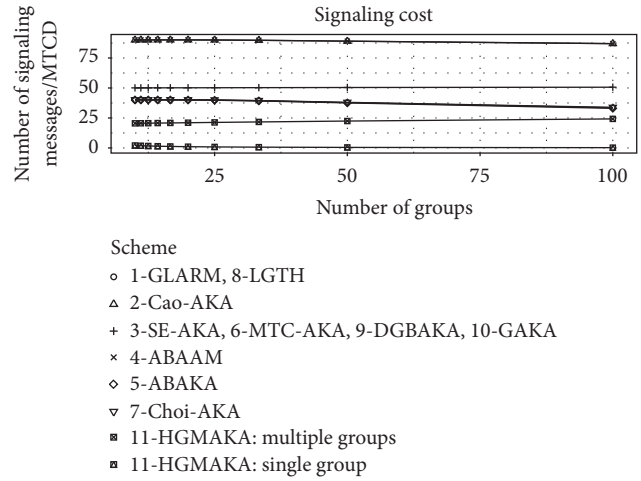

Figure 7: Comparison of signaling messages per MTCD with increasing number of groups.

A total of $t = 10$ runs of the authentication protocol have been considered. For each metric, comparison of the various protocols has been performed considering the probabilities mentioned in Table 3.

8.1. Number of Signaling Messages. The number of message exchanges that takes place for t rounds of authentication for different number of groups across existing group based protocols vis-à-vis proposed HGMAKA protocol was considered. Only group based protocols are considered as the advantages of group based as against nongroup based schemes have already been shown in existing literature. As the HGMAKA protocol allows group sizes to be large due to the hierarchical nature of the aggregation, two variants of this were compared: one with a single large group with the entire population of 10000 MTCDs as members and the other consisting of multiple smaller groups.

The HGMAKA protocol requires the lowest number of signaling messages as compared to the existing protocols as seen in Figure 7. Furthermore, the reduction in signaling messages with increasing number of groups is far more significant in the single group variation as compared to the multiple group version, the reason being that even though the probability of reauthentication increases with larger group size, the proposed protocol does not require the entire group to be reauthenticated in case of failure.

8.2. Communication Cost. The communication costs of the protocols are computed as the sum of the sizes of the individual messages that are exchanged for a full round of AKA.

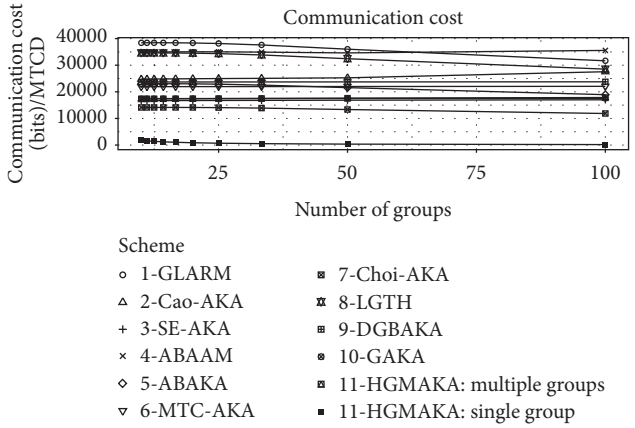

Figure 8: Comparison of communication cost (in bits) per MTCD with increasing number of groups.

The parameters used are placed in the annexure for reference; for example, the size of the messages exchanged between Tier 1 and Tier 2 elements, Tier 2 and Tier 3 elements, and Tier 3 and core networks in HGMAKA is added to arrive at its communication cost. The plot in Figure 8 compares the communication costs.

HGMAKA protocol, with a single group, requires the least communication cost which is significantly lower as compared to others. When considered in multiple smaller group, it still exhibits a low communication cost, with only Choi-AKA and SE-AKA performing a little better.

8.3. Computational Cost. For calculation of the computational complexity of the protocols, only significant cryptographic operations like hash, point multiplication, map-to-point, pairing, and so forth were considered. A table has been included in the annexure that lists the execution times of these operations using Crypto++ library on a Celeron 1.1 GHz processor as MTCD and Dual Core 2.6 GHz processor as a network element (MME/HSS) as reported in [16].

The plots in Figure 9 show that the computation cost exhibited by HGMAKA protocol is low, sharing the same with only three other protocols: MTC-AKA, Choi-AKA, and GAKA.

While the main motivation was to reduce the signaling load on the network, HGMAKA is seen to be light even in terms of communication and computation costs.

9. Conclusion

This paper presents a hierarchical group based mutual authentication scheme, HGMAKA, for Machine Type Communication over LTE network. The proposed lightweight symmetric key based HGMAKA protocol introduces an architectural model that utilizes a heterogeneous network of femtocells, mobile femtocell, and capillary network of MTCDs in line with the future 5G networks. The HGMAKA protocol contributes by reducing the overall signaling load on the access network eNBs and offloads the same to the smaller cells. It also significantly reduces the number of signaling

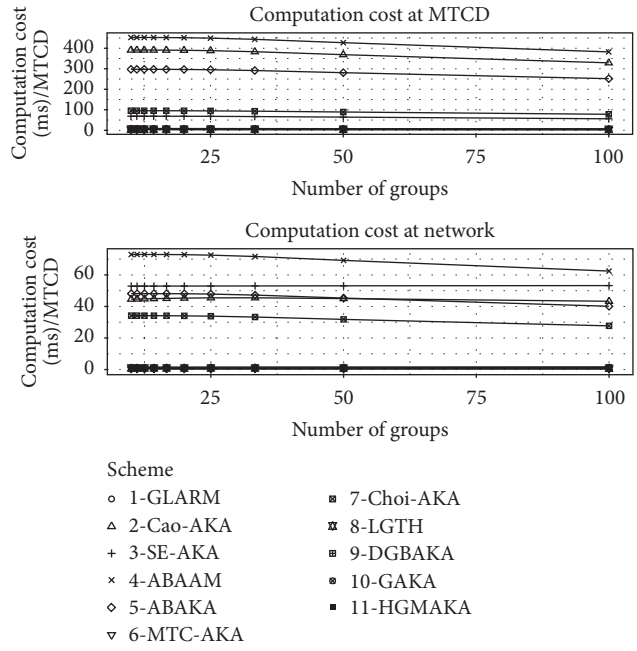

Figure 9: Comparison of computation cost per MTCD at the MTCD and at the network with increasing number of groups.

message exchanges and the size of the exchanged messages by using a single aggregate MAC in place of individual MACs for each MTCD. The scheme also addresses the important issue with aggregate MACs: single corrupt MAC-message pair invalidating the aggregate MAC, through the use of hierarchical en route filtering of MACs. HGMAKA has been shown to perform significantly better compared to other group based protocols. Moreover, this also releases the overhead of group leader management, which includes selection of group leader, its failure, and so forth, which plagues many of the group based protocols.

Appendix

A. Additional Information

A.1. Example Scenario Used for Evaluation of Performance Parameters of Various Schemes. An area where M2M communication is currently used is the Smart Metering Applications. These applications deal with monitoring and management of utilities like electricity, gas, or water. Some of the activities include obtaining meter reading, self-configuration of smart meters, monitoring usage and detecting outages, leakage, and taking necessary actions, like closing-down valves, circuit breakers, and so forth. From the HGMAKA architecture perspective, one can look at this application from the following viewpoints:

(i) First, the smart meters can be considered as MTCDs (Tier 1) which can communicate with a gateway device (Tier 2) which further communicates with the eNodeB (Tier 3) for communication with the core network.

(ii) Second, the household appliances, devices, and so forth can be considered to be MTCDs (Tier 1) forming a capillary network with the smart meters acting as the gateway device (Tier 2). The gateway device can further communicate with the eNodeB (Tier 3) for the last mile connectivity to core network.

In both cases, the final aggregator is the eNodeB. This allows group sizes to be very large, often spanning across a very large geographical area. This is in contrast with other aggregation schemes where the aggregator is a group leader selected from across the MTCDs. In the latter scenario, all MTCDs in a group must be colocated within a specific range of the group leader thus limiting the group size. Consequently, the larger group will result in lower signaling overload on the access network as opposed to multiple smaller groups formed from the same number of MTCDs.

A housing complex with 10 multistoried apartment blocks or towers is considered as test case. Each block consists of 100 households, each equipped with a smart meter. That gives approximately 1000 smart meters in the entire complex. Each apartment block has a gateway device (a total of 10 in the complex) installed by the operator which communicates with the smart meters, aggregates the data/authentication signals received, and sends a single message/signal to the eNodeB. Next, we assume that there are 10 such housing complexes located in the nearby area, all under the coverage of the same macro base station, increasing the number of smart meters to 10000. If all houses in the locality are availing the services of the same utility provider, then all 10000 smart meters can be said to belong to a single group.

Whenever the utility provider wants to communicate with the smart meters, all 10000 smart meters will try to connect to the core network to send their usage, billing, or any other information thus overloading the eNodeB.

As per the proposed hierarchical architecture, which allows large groups, all 10000 smart meters form a group, communicating with the gateways which aggregates and forwards the messages/signals to the eNodeB, which further proceeds with the last level aggregation, and ultimately a single message/signal is forwarded to the MME.

In schemes, where aggregation is performed by group leader, the size of groups will be small, say all 100 smart meters in an apartment block forming a single group. One of the smart meters in each block is selected as a group leader and aggregates the messages/signals for all 100 meters. Thus, for each housing complex consists of 10 groups and for 10 such complexes we will have 100 groups. The group leader sends a single message/signal per group to the eNodeB which forwards them (as received without any aggregation) to the MME.

A.2. Formal Verification Using AVISPA. See Algorithms 3 and 4.

A.3. Performance Comparison. See Tables 4 and 5.

Table 4: Parameters for computing communication costs.

Field	Size (bits)
$K_{Gi\text{-}j}$	128
R	128
SQN	48
AMF	16
MAC	64
RES/XRES	128
CK	128
IK	128
AK	48
IMSI	64
LAI	40
GK	128
GID	64
K_{ASME}	256
ID_{MME}	24
ECDSA public key	160
Timestamp	32
Digital signature	320
Temporary ID	128
Pseudo-ID	320
Security value	128
LMK	256

Table 5: Parameters for computing computation costs.

	Operation	Time (millisec)
At MTCD	Digital signature	4.77
	Hashing	0.0356
	Point multiplication	1.537
	Map-to-point	1.537
	Exponentiation	1.698
	Pairing	38.376
	Decryption	4.77
	Encryption	0.18
At network	Modulus	1.698
	Hashing	0.0121
	Point multiplication	0.475
	Map-to-point	0.475
	Exponentiation	0.525
	Pairing	16.322
	Digital signature	4.77

B. Formal Security Analysis Definitions

B.1. Ideal World. The ideal world consists of entities (denoted by M_i): MTCD or network (can be either home network or serving network, referred to simply as network). MTCD entities communicate with network entities in a two-party setting. Each M_i ($i = 1, 2, \ldots$) can participate in multiple instances denoted by I_{ij} ($j = 1, 2, \ldots$). An adversary plays a game with its opponent a ring master who generates random numbers. The adversary can create and connect entity

instances and can query the ring master. The ring master also provides session keys to the entity instances. The adversary can perform operations which are recorded in a transcript. These operations are as follows:

(i) *(Initialize entity, i, role$_i$, ID$_i$)*: Adversary assigns an identity ID$_i$ (a unique arbitrary bit-string) to the ith entity with the role as 0 for a MTCD entity and 1 for network entity.

(ii) (*Initialize entity instance, i, j, PID$_{ij}$*): It initializes an entity instance for previously initialized entities and assigns a partner identity for this instance. In case of group based architecture, for a network entity instance, the PID$_{ij}$ can be a list corresponding to the IDs of all MTCDs in the group.

(iii) (*Abort session, i, j*): It aborts a previously initialized entity instance.

(iv) (*Start session, i, j, connection_assignment [, key]*): It specifies how a session key K_{ij} for an entity instance I_{ij} is generated. The possible connection assignments are as follows:

(1) *(Create, i', j')*: A random bit-string K_{ij} is generated by the ring master. This is valid only if

(a) $I_{i'j'}$ is an instance compatible with I_{ij} and initialized earlier that is PID$_{ij}$ = ID$_{i'}$ and PID$_{i'j'}$ = ID$_i$ and role$_i$ ≠ role$_j$,

(b) *(create, i', j')* had not been made earlier either on I_{ij} or any other instance.

We can say that I_{ij} is isolated for $I_{i'j'}$ once the *start session* operation completes.

(2) *(Connect, i', j')*: The key K_{ij} is set to $K_{i'j'}$ by the ring master. This assignment is legal only if $I_{i'j'}$ is isolated for I_{ij}.

(3) *Compromise*: *The* ring master sets K_{ij} to *key*.

(v) *(Application, f)*: $f(R, \{K_{ij}\})$ where R is a random input and K_{ij} is a session key. The adversary receives the output of this function from the ring master.

(vi) *(Implementation, comment)*: Inserting comments to the transcript by the adversary.

The activities of the adversary A are logged in the transcript T_A.

B.2. Real World. In the real world, the adversary has full control over the channel. The real world adversary works towards defeating the Mutual Authentication and Key Agreement goals. The adversary starts by initializing the entities using the *initialize entity* operation followed by the *initialize entity instance* operations. In addition to the operations available in the ideal world, the entities in the real world can perform protocol specific operations which can modify their internal state. A network entity N_i with identity ID$_i$ can start an initialization routine whereby it interacts with other network entities with which it has service agreements. The ring master provides a list of networks to N_i with which it has service agreements. N_i sends agreement messages to each network in this list to which the other party responds with an approval message. Communication between the two networks is modeled with the ring master providing a secure communication with the help of mailboxes at both sides.

Each MTCD entity has to be registered with some network entity through the execution of a registration routine. The identity of an MTCD is assumed to be prefixed with the identity of the home network to which it is registered (e.g., IMSI). During the registration routine of MTCD U_i with previously initialized network $N_{i'}$, the ring master assigns arbitrary bit-string K_i to both U_i and $N_{i'}$ which is stored in the variable LTS$_i$ by U_i and (K_i, U_i) in variable LTS$_{i'}$ by $N_{i'}$.

Additionally, a group registration routine is carried out between a group of MTCDs and the network. Here, a list of IDs of MTCD entities forming a group is assigned a group identity GID$_i$ and an arbitrary bit-string as group key GK$_i$ by the ring master. The identity and the key are sent to all MTCDs and $N_{i'}$. The MTCD U_i stores (GID$_i$, GK$_i$) in variable GS$_i$ and $N_{i'}$ stores the same in GS$_{i'}$.

An entity instance is implemented in the form of a state machine having access to ID$_i$, role$_i$, LTS$_i$, and GS$_i$. A state change occurs on receiving some message of the form *(deliver message, i, j, type, InMsg)* from an adversary. I_{ij} responds by reporting *OutMsg* along with status to the adversary. The status can be *continue* (if I_{ij} can receive a next message), *accept* (I_{ij} ends with session key SK$_{ij}$), or *reject* (I_{ij} ends without session key). The transcript T_A records *(implementation, deliver message, i, j, type, InMsg, OutMsg, status)* and (*start session, i, j*) if status is *accept* and *(abort session, i, j)* if status is *reject*.

Also, the operation (*application*, f) can be executed by the adversary using the actual session key {SK$_{ij}$}.

B.3. Some Preliminaries

B.3.1. Negligible Function. A real valued function $\epsilon(k)$ is negligible (in k, k is nonnegative) if for every $c > 0$ there exists $k_c > 0$ such that $\epsilon(k) < k^{-c}$ for all $k > k_c$.

B.3.2. Function Family. A function family is a map $F : \mathcal{K} \times D \to R$, where $\mathcal{K}$ is the set of keys, D is the domain of F, and R is the range of F. The set of keys and R are finite sets. The function F takes two inputs $K \in \mathcal{K}$ and X as input and returns a point Y in R. $F(K, X) = Y$.

For a key $K \in \mathcal{K}$, the mapping $F_K : D \to R$ is defined by $F_K(X) = F(K, X) = Y$. F_K is said to be an instance of the function family F.

B.3.3. Random Function. Let $R = \{0, 1\}^n$ be a finite set and let *Fn* be an oracle which implements a random function. If an adversary queries (x), where x is an input parameter, it receives a random point from R. If $Fn(x)$ is queried multiple times, the response remains unchanged.

B.3.4. Pseudorandom Function Family. It is a family of functions with the property that the input-output behavior of

a random instance of the family is computationally indistinguishable from a truly random function. For a family of functions $F : \{0,1\}^k \times \{0,1\}^l \to \{0,1\}^L$ and $F_K : D \to R$ with $K \in \mathcal{K}$. In the real world, F_K is an instance of F and in the ideal world F_K is a truly random function.

B.3.5. Distinguishing Advantage. The advantage of a probabilistic polynomial time algorithm D in distinguishing between two families of random variables is $X = \{X_k\}_{k\geq 0}$ and $Y = \{Y_k\}_{k\geq 0}$, where X_k and Y_k take values from a finite set S_k. The output of D is 0 or 1 depending on whether it can distinguish between X and Y. The distinguishing advantage of D is

$$\mathrm{Adv}^{\mathrm{dist}}_{X_k}(D) = \left|\Pr\left(D\left(X_k\right) = 1\right) - \Pr\left(D\left(Y_k\right) = 1\right)\right|. \quad \text{(B.1)}$$

B.3.6. Prf-Advantage. Let $F : \mathcal{K} \times D \to R$ be a family of functions and let $G : D \to R$ be another family of functions from $\{0,1\}^l$ to $\{0,1\}^L$ and let A be a probabilistic polynomial time oracle algorithm. The prf-advantage of A is

$$\mathrm{Adv}^{\mathrm{prf}}_F(A) = \left|\Pr\left(A\left(F\right) = 1\right) - \Pr\left(A\left(G\right) = 1\right)\right|. \quad \text{(B.2)}$$

An insecurity function associated with F is given by

$$\mathrm{Adv}^{\mathrm{prf}}_F(t,q) = \max_{A\in\mathcal{A}(t,q)}\mathrm{Adv}^{\mathrm{prf}}_F(A), \quad \text{(B.3)}$$

where $\mathcal{A}(t,q)$ is the set of adversaries that make a maximum of q oracle queries and have running time of t. If, for every probabilistic polynomial time adversary A, $\mathrm{Adv}^{\mathrm{prf}}_F(A)$ is negligible in k, then we say that F is a pseudorandom family.

B.3.7. MAC Advantage. A Message Authentication Code is a family of functions : $\{0,1\}^k \times \mathrm{Dom}(F) \to \{0,1\}^L$, where $\mathrm{Dom}(F)$ is the domain of F and $\mathrm{Dom}(F) = \{0,1\}^{\leq l}$. For $K \in \{0,1\}^k$ and $M \in \{0,1\}^{\leq l}$ $\sigma = F(K,M)$ is the MAC of M. An adversary, A, is a probabilistic polynomial time algorithm that has oracle access for computing MAC for random key J. The mac advantage of A is the probability that A outputs (σ, M) such that $\sigma = F(K,M)$ and M was not a query to the oracle by A. The insecurity function of F is

$$\mathrm{Adv}^{\mathrm{mac}}_F(t,q) = \max_{A\in\mathcal{A}(t,q)}\mathrm{Adv}^{\mathrm{mac}}_F(A). \quad \text{(B.4)}$$

We say that F is a secure message authentication code if, for every polynomially bounded adversary A, $\mathrm{Adv}^{\mathrm{mac}}_F(A)$ is negligible in k.

B.4. Definitions

B.4.1. Stimulus. For a real world entity instance I_{ij}, a message received by I_{ij} causing it to change its status to accept is called a stimulus on I_{ij}.

B.4.2. Authentic Transcript. For every real world adversary A with transcript T_A, T_A is said to be authentic if, for an instance I_{ij}, the stimulus to accept comes only from another compatible instance.

B.4.3. Collision-Free Transcript. For every real world adversary A with transcript T_A, T_A is said to be collision-free if every entity and its instances generate unique nonrepeated random numbers.

B.4.4. Collision Resistant Transcript. For every real world adversary A with transcript T_A, if for function family F is used by entity and entity instance to compute tags $\sigma_1, \sigma_2, \ldots, \sigma_n$ and $\sigma_i \neq \sigma_j$, $i \neq j$, we say that F is collision resistant in T_A.

Notations

$\mathrm{MTCD}_{\mathrm{G}i\text{-}j}$:	MTCD j belonging to group Gi
$\mathrm{IMSI}_{\mathrm{G}i\text{-}j}$:	Unique International Mobile Subscriber Identity Number for $\mathrm{MTCD}_{\mathrm{G}i\text{-}j}$
$\mathrm{GID}_{\mathrm{G}i}$:	Group identifier of group Gi
$\mathrm{GK}_{\mathrm{G}i}$:	Group key of group i shared by all members of the group with the HSS. The group key is also shared between the Tier 2 and Tier 3 elements catering to the group
$K_{\mathrm{G}i\text{-}j}$:	Secret key shared between the $\mathrm{MTCD}_{\mathrm{G}i\text{-}j}$ with the HSS
${M_{\mathrm{G}i\text{-}j}}^1$:	Authentication request message generated by $\mathrm{MTCD}_{\mathrm{G}i\text{-}j}$
$R_{\mathrm{G}i\text{-}j}$:	Random number generated by $\mathrm{MTCD}_{\mathrm{G}i\text{-}j}$
$\mathrm{MAC}_{\mathrm{G}i\text{-}j}$:	Message authentication code sent by $\mathrm{MTCD}_{\mathrm{G}i\text{-}j}$ for authentication by MME
${\mathrm{MAC}_{\mathrm{G}i\text{-}j}}^1$:	Message authentication code sent by $\mathrm{MTCD}_{\mathrm{G}i\text{-}j}$ for verification by Tier 2 element
R_{HSS}:	Random number generated by HSS
${\mathrm{aggMAC}_{\mathrm{G}i}}^{k2}$:	Aggregate MAC generated by kth Tier 2 element
${\mathrm{agg}M_{\mathrm{G}i}}^{k2}$:	Aggregate message generated by kth Tier 2 element
${\mathrm{MAC}_{\mathrm{G}i}}^{k2}$:	MAC generated by kth Tier 2 element of group Gi for verification by Tier 3 element
$\mathrm{aggMAC}_{\mathrm{G}i}$:	Final aggregate MAC generated by Tier 3 element
$\mathrm{agg}M_{\mathrm{G}i}$:	Final aggregate message generated by Tier 3 element
SN_ID:	Serving Network Identity
$\mathrm{TGK}_{\mathrm{G}i}$:	Temporary Group Key for group Gi, used as a group session key
${\mathrm{MAC}_{\mathrm{G}i}}^{\mathrm{HSS}}$:	MAC generated by HSS for group Gi
$\mathrm{AUTH}_{\mathrm{HSS}}$:	Authentication token generated by HSS
CK:	Ciphering key
IK:	Integrity key
AK:	Anonymity key
SQN:	Sequence number
${K^{\mathrm{ASME}}}_{\mathrm{G}i\text{-}j}$:	Session key between MME and MTCD i
$\mathrm{XRES}_{\mathrm{G}i\text{-}j}$:	Expected response to the challenge sent by MME to $\mathrm{MTCD}_{\mathrm{G}i\text{-}j}$
$\mathrm{RES}_{\mathrm{G}i\text{-}j}$:	Response generated by $\mathrm{MTCD}_{\mathrm{G}i\text{-}j}$ in response to challenge by MME
GKI:	Group Key Index
$\mathrm{aggRES}_{\mathrm{G}i}$:	Aggregate response by Tier 3 element to MME

$f1$, $f3$, $f4$, $f5$, $f6$: One-way hash functions
KDF: Key derivation function
$f2$: Message authentication code generating function.

Competing Interests

The authors declare that they have no competing interests.

References

[1] "Cisco Visual Networking Index," http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html.

[2] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Security aspects of Machine-Type Communications (Rel 12), 3GPP TR 33.868 V0.10.0, Septemper 2012.

[3] "FP7: Seventh Framework Programme," https://ec.europa.eu/research/fp7/index_en.cfm.

[4] H. Droste, G. Zimmermann, M. Stamatelatos et al., "The METIS 5G architecture: a summary of METIS work on 5G architectures," in *Proceedings of the 81st IEEE Vehicular Technology Conference (VTC '15)*, pp. 1–5, May 2015.

[5] A. Osseiran, F. Boccardi, V. Braun et al., "Scenarios for 5G mobile and wireless communications: the vision of the METIS project," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 26–35, 2014.

[6] "iJOIN," http://www.ict-ijoin.eu/.

[7] "TROPIC," http://www.ict-tropic.eu/.

[8] 3rd Generation Partnership Project and Technical Specification Group Services and System Aspects, "Service requirements for Machine-Type Communications (MTC) (Rel 12)," 3GPP TS 22.368 V12.0.0, 2012.

[9] 3rd Generation Partnership Project and Technical Specification Group Services and System Aspects, "Service requirements for Home Node B (HNB) and Home eNode B (HeNB) (Rel 12)," 3GPP TS 22.220 V12.0.0, 2012.

[10] 3rd Generation Partnership Project, "Technical Specification Group Service and System Aspects; 3GPP System Architecture Evolution (SAE)," Security architecture (Rel 12) 3GPP TS 33.401 V12.13.0, 2012.

[11] J. Arkko and H. Haverinen, "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)," IETF RFC 4187, January 2006.

[12] M. Condoluci, M. Dohler, G. Araniti, A. Molinaro, and K. Zheng, "Toward 5G densenets: architectural advances for effective machine-type communications over femtocells," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 134–141, 2015.

[13] F. Haider, C.-X. Wang, H. Haas et al., "Spectral efficiency analysis of mobile Femtocell based cellular systems," in *Proceedings of the IEEE 13th International Conference on Communication Technology (ICCT '11)*, pp. 347–351, IEEE, Jinan, China, September 2011.

[14] 3rd Generation Partnership Project and Technical Specification Group Services and System Aspects, "Study on enhancements for Machine-Type Communications (MTC) (Release 12)," V12.0.0, 3GPP TR 22.888, 2012.

[15] V. B. Mišić, J. Mišić, X. Lin, and D. Nerandzic, "Capillary machine-to-machine communications: the road ahead," in *Ad-Hoc, Mobile, and Wireless Networks*, X.-Y. Li, S. Papavassiliou, and S. Ruehrup, Eds., vol. 7363 of *Lecture Notes in Computer Science*, pp. 413–423, Springer, Berlin, Germany, 2012.

[16] C. Lai, H. Li, R. Lu, and X. Shen, "SE-AKA: a secure and efficient group authentication and key agreement protocol for LTE networks," *Computer Networks*, vol. 57, no. 17, pp. 3492–3510, 2013.

[17] C. Lai, H. Li, X. Li, and J. Cao, "A novel group access authentication and key agreement protocol for machine-type communication," *Transactions on Emerging Telecommunications Technologies*, vol. 26, no. 3, pp. 414–431, 2015.

[18] Y. Zhang, J. Chen, H. Li, W. Zhang, J. Cao, and C. Lai, "Dynamic group based authentication protocol for machine type communications," in *Proceedings of the 4th International Conference on Intelligent Networking and Collaborative Systems (INCoS '12)*, pp. 334–341, IEEE, Bucharest, Romania, September 2012.

[19] Y.-W. Chen, J.-T. Wang, K.-H. Chi, and C.-C. Tseng, "Group-based authentication and key agreement," *Wireless Personal Communications*, vol. 62, no. 4, pp. 965–979, 2012.

[20] J. Cao, M. Ma, and H. Li, "A group-based authentication and key agreement for MTC in LTE networks," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '12)*, pp. 1017–1022, Anaheim, Calif, USA, December 2012.

[21] J. Cao, M. Ma, and H. Li, "Access authentication of mass device connections for MTC in LTE networks," *The Smart Computing Review*, vol. 4, no. 4, pp. 262–277, 2014.

[22] C. Lai, H. Li, R. Lu, R. Jiang, and X. Shen, "LGTH: a lightweight group authentication protocol for machine-type communication in LTE networks," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '13)*, pp. 832–837, December 2013.

[23] D. Choi, S. Hong, and H.-K. Choi, "A group-based security protocol for machine type communications in LTE-advanced," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS '14)*, pp. 161–162, IEEE, Ontario, Canada, May 2014.

[24] C. Lai, R. Lu, D. Zheng, H. Li, and X. Sherman, "GLARM: group-based lightweight authentication scheme for resource-constrained machine to machine communications," *Computer Networks*, vol. 99, pp. 66–81, 2016.

[25] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, *Aggregate and Verifiably Encrypted Signatures from Bilinear Maps*, Springer, Berlin, Germany, 2003.

[26] D. Naccache, M. Just, B. Preneel et al., "Nyberg–rueppel signature scheme," in *Encyclopedia of Cryptography and Security*, p. 879, Springer, Boston, Mass, USA, 2011.

[27] J.-L. Huang, L.-Y. Yeh, and H.-Y. Chien, "ABAKA: an anonymous batch authenticated and key agreement scheme for value-added services in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 1, pp. 248–262, 2011.

[28] E. Klaoudatou, E. Konstantinou, G. Kambourakis, and S. Gritzalis, "A survey on cluster-based group key agreement protocols for WSNs," *IEEE Communications Surveys and Tutorials*, vol. 13, no. 3, pp. 429–442, 2011.

[29] T. Rams and P. Pacyna, "A survey of group key distribution schemes with self-healing property," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 2, pp. 820–842, 2013.

[30] R. Roman, C. Alcaraz, J. Lopez, and N. Sklavos, "Key management systems for sensor networks in the context of the Internet of Things," *Computers and Electrical Engineering*, vol. 37, no. 2, pp. 147–159, 2011.

[31] AVISPA: Automated Validation of Internet Security Protocols and Applications, http://www.avispa-project.org/.

[32] V. Shoup, "On formal models for secure key exchange," Tech. Rep., 1999.

[33] M. Zhang and Y. Fang, "Security analysis and enhancements of 3GPP authentication and key agreement protocol," *IEEE Transactions on Wireless Communications*, vol. 4, no. 2, pp. 734–742, 2005.

[34] Y.-L. Huang, C.-Y. Shen, and S. W. Shieh, "S-AKA: a provable and secure authentication key agreement protocol for UMTS networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 9, pp. 4509–4519, 2011.

[35] Y.-L. Huang, C. Y. Shen, S. Shieh, H.-J. Wang, and C.-C. Lin, "Provable secure AKA scheme with reliable key delegation in UMTS," in *Proceedings of the 3rd IEEE International Conference on Secure Software Integration Reliability Improvement (SSIRI '09)*, pp. 243–252, Shanghai, China, July 2009.