GBAAM: Group-based Access Authentication for MTC in LTE Networks

Jin Cao, Maode Ma, Hui Li

1 State Key Laboratory of Integrated Service Network, Xidian University, China
2 State Key Laboratory of Information Security, Institute of Information Engineering, CAS, Beijing, China
3 School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore

ABSTRACT

Machine Type Communication (MTC), as one of the most important communication approaches in the future mobile communication, has drawn more and more attention. To meet the requirements of low power consumption of devices and mass device transmission is the key issue to achieve MTC applications security in the Long Term Evolution (LTE) networks. When a large number of MTC devices simultaneously connect to the network, each MTC device needs to implement an independent access authentication procedure in the current third Generation Partnership Project (3GPP) standard, which will cause a severe signaling congestion in the LTE network. In this paper, we propose a group-based access authentication scheme, by which a good deal of MTC devices can be simultaneously authenticated by the network and establish an independent session key with the network respectively. Our scheme cannot only greatly reduce the signal transmission for mass of devices to the network and thus avoid the signaling overload over the LTE network, but also achieve robust security including Key Forward/Backward Secrecy (KFS/KBS) and non-repudiation verification. The experimental results and formal verification by using the TLA+ and TLC show that the proposed scheme is secure against various malicious attacks. Copyright © 2015 John Wiley & Sons, Ltd.

KEYWORDS

group-based access authentication; MTC; LTE; TLA+ and TLC; aggregate signature

Correspondence
E-mail: caoj897@gmail.com

Received . . .

1. INTRODUCTION

MTC, also named as Machine to Machine (M2M) communication, is viewed as one of next sophisticated techniques for future wireless communications, which has attracted much attention of standardization organizations such as 3GPP. Different from the traditional human to human (H2H) communications designed in the current wireless networks, MTC is a special type of data communications between entities that can exchange and share data without any requirement on any form of human intervention. It is mainly used for automatically collecting and delivering measurement of data such as real-time monitoring data from remote areas to the network by wire, radio or other mechanisms. Due to its characteristics of low power, low cost, and no human intervention, the development of MTC has quickly become the driving force of the market for a large number of real-time network applications, such as public safety, environmental monitoring, and so on.

With the emergence of new applications, MTC needs to be defined on the existing wireless platforms to meet the demands of higher data rate, better coverage and scalable expansion. Owing to the higher throughput and lower transmission latency, 3GPP LTE in comparison with other networks, will play a central role in real-time mobile MTC scenarios [1, 2].

Different from the common user equipments (UEs), MTC devices (MTCDs) bring some new requirements including lower power consumption and mass device transmission, and the number of MTCDs may increase quickly. It is estimated that the number of MTCDs will be 1000 times larger than the number of the common UEs. The analysis on the MTC market predicts, by 2014, there
A dynamic group member mechanism is addressed. Furthermore, the GBAAM scheme can resist the several existing attacks and achieve robust security protection including KFS/KBS and non-repudiation verification.

A dynamic group member mechanism is addressed to achieve the dynamic nature of MTC group for MTC without a complex group key management process in LTE networks. A formal verification is conducted by employing Temporal Logic Actions (TLA+) and its model checker Temporal Logic Checker (TLC) to demonstrate security functionality of our proposal against various malicious attacks.

1.1. Overview of existing congestion avoidance proposals

Until now, only a few schemes for congestion avoidance have been proposed [9–19].

Device Group Algorithm (DGA): A new MTCD grouping algorithm for congestion avoidance has been designed in [9]. By the scheme, multiple MTCDs can construct a managed group by using various technologies such as Bluetooth, and choose a group leader. Then the group members can transfer and receive data to/from the MTC server via the group leader. The scheme can largely reduce the network traffic load by using a grouping scheme. However, it is mainly aimed at avoidance of congestion in the communications without consideration of security between the MTC devices and the MTC server. Thus it does not have any function to support an access authentication procedure for multiple devices to register to the network.

GAKA: A group-based authentication and key agreement (GAKA) method for a group of UEs roaming from the same home network (HN) to a serving network (SN) has been presented in [10]. By the scheme, multiple UEs, which belong to the same HN, can form a group of UEs. When the first UE in a group moves to the SN, the SN obtains authentication information for the UE and other members from the concerned HN by performing full authentication. Thus, when other group members come into the SN, the SN can authenticate each of them locally without the involvement of the HN.

Subsequently, a lot of group-based access authentication schemes [11–16] have been proposed taking the advantage of the scheme in [10].

DGBAKA: In the Dynamic Group Based Authentication and Key Agreement (DGBAKA) scheme [11, 12], each MTCD pre-shares a group key with home environment and the other MTCDs in the same group. The group key will be used for authenticating with serving network locally. In addition, the group key generation and group key update methods have been proposed in the DGBAKA scheme, which can achieve the dynamic nature of MTC group in practical applications.
MTC-AKA: In the MTC-AKA scheme [13], the same group shares the same authentication data including a group temporary key (GTK) and other necessary authentication information. When the first MTCD attaches to the SN, the first MTC device in a group completes a full authentication procedure and obtains a GTK on behalf of other MTCDs in the same group. Then the remaining MTCDs in the same group use the GTK to complete the mutual authentication and key agreement with the HN locally.

SE-AKA: The secure and efficient group (SE)-AKA scheme [14] employs the asymmetric key cryptosystem and Elliptic Curve Diffie-Hellman (ECDH) to provide robust security protection including privacy-preservation and KBS/KFS. The access authentication process of the SE-AKA scheme is similar to the MTC-AKA scheme, which also adopts the generation of the GTK to simplify the whole authentication process.

EG-AKA: By adopting the concept of the SE-AKA scheme, the EAP-based Group (EG)-AKA scheme [15] facilitates a group of MTCDs access to LTE core network through non-3GPP access network.

LGTH: In the Lightweight Group Authentication Protocol (LGTH) [16], a large number of MTCDs construct a MTC group and pre-share a group key. And, a group leader can be selected based on the communication capability in the register phase. Then, the MTC group can simultaneously achieve the authentication and key agreement with the HSS by adopting the aggregate message authentication codes.

OBBA: An opportunistic batch bundle authentication scheme (OBBA) has been proposed in [17] to achieve efficient bundle authentication for delay tolerant networks (DTNs). The scheme utilizes the batch verification techniques to largely reduce the computational cost of the intermediate routers of DTNs.

ABAKA: An anonymous batch authenticated and key agreement (ABAKA) scheme for Vehicle to Vehicle (V2V) communication in vehicular Ad Hoc networks (VANETs) has been proposed in [18]. V2V communication is a special type of M2M communication. Since the service provider (SP) can simultaneously authenticate multiple of vehicles and establish different session keys with different vehicles, the scheme alleviates the burden of the SP.

Group Key Management (GKM): A pyramidal security model for large-scale group-oriented computing in mobile ad hoc networks (MANETs) has been proposed to safeguard the multisecurity-level information sharing in one cooperation domain [19]. By the scheme, an efficient group key management solution called integrated tree key graph is designed to utilize only one tree key graph to manage all the involved multicast groups in the proposed pyramidal security model and thus solve the issue of managing frequent key changes for a set of multicast groups.

1.2. Analysis of existing congestion avoidance proposals

These schemes in [10–15] can reduce the communication cost between the HN and the SN. However, they still cannot avoid the network congestion over the SN nodes when a mass of devices come into the SN simultaneously. In addition, these schemes require an establishment of group message management list and involve complex group key management mechanisms, which increase the complexity of the entire system. Furthermore, there are no group key generation method and dynamic group member management mechanisms in the schemes [10,13–16], which do not support dynamic group, and thus be impracticable for MTC in LTE networks.

Adopting the concept of batch verification techniques, the schemes in [17,18] can largely reduce the computational overhead of the network. However, there is still a network congestion problem in the schemes in [17,18] because multiple request messages need to be sent to the network. In addition, Wang and Zhang [20] have pointed out that the scheme in [18] can not resist the conspiracy attack.

Since this scheme [19] is only to design an efficient group key management mechanism for a dynamic multicast environment in MANETs, which can be applied to guarantee the security of information-sharing among group members, and it is unable to achieve the access authentication between group members and the network for in LTE networks.

In conclusion, we summarize kinds of features supported by the congestion avoidance proposals as shown in Table 1. The grouping approach in [9] fits in with mass device transmission for MTC in the LTE networks. However, it cannot handle network congestion existing in mass device access authentication procedures. By the schemes in [10–15,17,18], each device still needs to send authentication request message to the network independently, which may result in network overload, when a mass of MTC devices simultaneously attach to the network. In addition, due to the lack of dynamic group member management mechanisms, these schemes [10,13–16] are not feasible in LTE networks. The group key management scheme in [19] cannot consider the mutual authentication between group members and the network in LTE networks. Note that the symbol "/" means these schemes can not involve these features or these features are beyond the scope of the schemes.

The remainder of this paper is organized as follows. In Section 2, we analyze the existing congestion avoidance proposals. In Section 3, the preliminary and the background of our solution are introduced. In Section 4, the proposed scheme is presented in details. In Section 5, the security analysis and formal verification on our solution are presented. The performance evaluation on our scheme is presented in Section 6. Finally, the paper is concluded in Section 7.
2. PRELIMINARY

2.1. MTC Network Architecture

As shown in Fig. 1, a MTC user, which is a control centre outside the network operator domain, can use the services provided by one or more MTC servers to operate a large number of MTCDs. A MTC server is a server, which can communicate to the LTE network itself, and to MTCDs via the LTE network. The MTC server also can be accessed by the MTC user with an application program interface (API). The LTE network allows the following two types of connections to the MTC server(s): the MTC server located within the operator domain with the control by the LTE network, and the MTC server collocated with the MTC user outside the operator domain without the control by the LTE network. The LTE network consists of Evolved Universal Terrestrial Radio Access Network (E-UTRAN) and Evolved Packet Core (EPC). E-UTRAN includes two types of base stations named as eNodeB (eNB) and Home eNodeB (HeNB). The EPC is comprised of a MME and a Service Gateway (SGW) together with a Packet Data Network Gateway (PDN GW). When a MTCD connects to the network, the MTCD can communicate with the MTC server and be controlled by the MTC user via the MTC server(s). To enable the communications between MTCDs and MTC server(s), the LTE network needs to authenticate and authorize the MTCDs before the MTCDs communication with the MTC server [4]. Same as a common UE, the MME represents the network to implement mutual authentication with the MTCD by the EPS-AKA [7].

![MTC Network Architecture](image_url)

**Figure 1.** MTC Network Architecture

2.2. Signaling Network Congestion

As shown in Fig. 2, signaling network congestion could be incurred when a large number of MTCDs try almost simultaneously to attach / activate / deactivate to the network [4]. In the 3GPP system, a lot of MTC applications could be supported such as mobile charging and metering / monitoring. In those applications, it is possible for a large number of metering or monitoring devices to become active almost simultaneously after a period of power outage or at precisely synchronous time intervals. When these MTCDs become active simultaneously, signaling overload and congestion could be triggered over the network nodes such as MME and HSS and thus the network may not be able to provide services for these MTCDs. In order to overcome the signaling congestions, there are two methods supported by the 3GPP committee [6]. Firstly, the network nodes should be able to reject or prevent the connection requests. The approach will

**Table 1.** Congestion Avoidance Proposals-Summary of Features Supported

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Supporting Group approach</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Supports mutual authentication between the group and the network</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Supports independent session key establishment between each device and the network</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Does not require an establishment of group message management list</td>
<td>/</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Device is not required to send authentication request independently</td>
<td>/</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>/</td>
<td>Yes</td>
</tr>
<tr>
<td>Supports dynamic group member management</td>
<td>/</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Provides Key Forward/Backward Secrecy (KFS/KBS)</td>
<td>/</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Signalling overhead</td>
<td>/</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>/</td>
<td>High</td>
<td>/</td>
<td>Low</td>
<td></td>
</tr>
<tr>
<td>Communication overhead</td>
<td>/</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>/</td>
<td>High</td>
<td>/</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Computational cost</td>
<td>/</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>/</td>
<td>High</td>
<td>/</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Withstands protocol attacks</td>
<td>/</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>/</td>
<td>No</td>
<td>/</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Provides non-repudiation verification</td>
<td>/</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>/</td>
<td>Yes</td>
<td>/</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>
bring a new problem that non-MTC traffic or the traffic from other MTC devices could be restricted. Moreover, in the blocked traffic at the particular MTC devices which cause the congestion, some significant data and messages cannot be timely received and/or sent by the network and thus quality of service for the MTC users will be impacted seriously. Another approach is that a MTC group will be formed with a large number of MTC devices for the LTE network to be able to handle easily. However, it is only applied to the communications between MTC devices and the MTC server without consideration of authentications between them in the current 3GPP standard.

It is the most important for a group of MTC devices to implement access authentication with the network before any communication, a new access authentication scheme for the simultaneous connection of multiple devices is required. The technique of aggregate signature can provide an efficient way to verify multiple devices at the same time and Identity-Based Cryptography (IBC) is much fit for the systems with low capabilities to guarantee system security owing to less storage, computational and communication cost required. Thus, we propose a new mass device access authentication scheme based on the improved ID-based aggregate signature to authenticate the group of MTC devices.

![Figure 2. Signaling Network Congestion](image)

2.3. Aggregate Signature and Identity-Based Cryptography

In 1984, Shamir [22] firstly proposed the concept of Identity-Based Cryptography (IBC). By the scheme, when a sender wants to send a message to a receiver, she can simply encrypt a message for a receiver using only the receiver’s identity information as a public key. Upon the receipt of the encrypted message, the receiver contacts the system’s trusted authority, called Private Key Generator (PKG) or Key Generate Center (KGC), to derive his private key through a secure out-of-band channel. Unlike conventional public key cryptography, IBC does not need to transmit, authenticate and maintain the public key certificates. In IBC, a public key is simply the receiver’s identity that uniquely identifies himself/herself, such as name or e-mail address.

In 2003, an aggregate signature scheme, which allows n signatures on n distinct messages from n distinct users to aggregate a single signature, has been proposed in [23]. This scheme largely reduces bandwidth and storage cost due to aggregation, and thus becomes more compelling for mobile devices with resource and energy constraints to reduce battery life. Since an ID-based encryption scheme based on the Weil pairing has been presented by Boneh and Franklin [24], a lot of ID-based aggregate signature schemes based on the bilinear pairings have been constructed [25–27]. Recently, an efficient ID-based aggregate signature scheme with constant pairing computations has been addressed [21]. The scheme works as follows.

Setup: KGC takes a security parameter k, and then generates system parameters and a master key, which works as follows.

1) Choose a k-bit prime p and generate two elliptic curve groups G1, G2 of order p, a generator \( P \) in \( G1 \) and an admissible pairing \( e: G1 \times G1 \rightarrow G2 \).
2) Choose a random number \( x \in R \) \( Z_p^* \) as the master key and compute the system public key \( PK = xP \).
3) Choose three cryptographic secure hash functions \( H_1: \{0, 1\}^* \rightarrow G1, H_2: \{0, 1\}^* \rightarrow Z_p \) and \( H_3: \{0, 1\}^* \rightarrow Z_p \).
4) Publish \( \{p, G1, G2, e, P, PK, H1, H2, H3\} \) as system parameters and keep the master key \( x \) secret.

Extract: Let \( A_1, \ldots, A_n \) be the set of users and each \( A_i \)'s identifier be \( ID_i \). For each user \( A_i \) (1 \( \leq i \leq n \)):

1) Compute \( Q_{ID_i} = H1(ID_i) \in G1 \).
2) Set the private key \( SI_{ID_i} \) to be \( xQ_{ID_i} \), where \( x \) is a master secret.

Sign: Each user \( A_i \) generates the signature on a message \( M_i \) as follows.

1) Choose \( r_i \in Z_p^* \) and compute \( U_i = r_i \cdot P \in G1 \).
2) Compute \( \sigma_i = H2(ID_i, M_i, U_i) \in Z_p \) and \( V_i = SI_{ID_i} + h_i \cdot r_i \cdot PK \in G1 \). The signature on is

\[ \sigma_i = (U_i, V_i) \]

Aggregation: Compute \( V = \sum_{i=1}^{n} V_i \) and output \( \sigma = (U_1, \ldots, U_n, V) \) as an aggregate signature.

Aggregation verify: Given an aggregate signature \( \sigma = (U_1, \ldots, U_n, V) \) as above.

1) Compute \( Q_{ID_i} = H1(ID_i) \) and \( h_i = H2(ID_i, M_i, U_i) \) for \( 1 \leq i \leq n \).
2) Verify \( e(V, P) \equiv e(\sum_{i=1}^{n} Q_{ID_i} + h_i \cdot U_i, PK) \).

If it holds, the aggregate signature is accepted.

3. THE PROPOSED SCHEME: GBAAM

In this section, we describe our proposed scheme, which consists of two phases, register phase and group-based access authentication phase as shown in Fig. 3 and Fig.
4. The notations used in the scheme are defined in Table II.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$</td>
<td>a $k$-bit prime</td>
</tr>
<tr>
<td>$Z_p$</td>
<td>a prime finite field</td>
</tr>
<tr>
<td>$G_1, G_2$</td>
<td>two elliptic curve groups</td>
</tr>
<tr>
<td>$P$</td>
<td>generator for the group $G_1$</td>
</tr>
<tr>
<td>$e$</td>
<td>A Tate or weil pairing $e : G_1 \times G_1 \rightarrow G_2$</td>
</tr>
<tr>
<td>$H_1(\cdot)$</td>
<td>a hash function $H_1 : {0, 1}^* \rightarrow G_1$</td>
</tr>
<tr>
<td>$H_2(\cdot)$</td>
<td>a hash function $H_2 : {0, 1}^* \rightarrow Z_p$</td>
</tr>
<tr>
<td>$H_3(\cdot)$</td>
<td>a hash function $H_3 : {0, 1}^* \rightarrow G_1 \rightarrow Z_p$</td>
</tr>
<tr>
<td>$T_{exp}, T_i$</td>
<td>The node $i$’s expiration time/ current time</td>
</tr>
<tr>
<td>$ID_i$</td>
<td>identity of node $i$, which is $ID_i = (\text{identifier} \parallel T_{exp})$</td>
</tr>
<tr>
<td>$x/ PK$</td>
<td>private/public key of KGC, $x \in R Z_p^*$ and $PK = xP$</td>
</tr>
<tr>
<td>$GID$</td>
<td>MTC group’s identity</td>
</tr>
<tr>
<td>$S_{ID_i}$</td>
<td>$i$’s private long-term key generated by KGC</td>
</tr>
<tr>
<td>$(S_{MME}, S_{MME})$</td>
<td>MME’s private long-term key</td>
</tr>
<tr>
<td>$SK_{MME}$</td>
<td>The session key between MME and MTCD$_i$</td>
</tr>
</tbody>
</table>

3.1. Basic Assumptions

Before the proposed scheme is described, we firstly give some basic assumptions for our scheme. We consider the traditional MTC service scenarios such as mobile charging and meter/monitoring where there are multiple MTCDs. In these scenarios, MTCDs are some type of reporting devices such as smart meters, sensor devices, and vital sign detectors, which are located at dense areas.

MTC devices support multiple communication technologies both mobile broadband technology and local area networking technology. Mobile broadband technology involves mobile WiMAX and LTE. Local area networking technology can be divided into wireless personal area network (WPAN) including Bluetooth, ZigBee, and UWB, and other coming technologies such as power line communications (PLC) [9]. Owing to the characteristics of low power, high capacity, broadcast and robust security, ZigBee technology has been widely applied in Internet of Things (IoT) or sensor network.

In this paper, it is assumed that MTCDs support both the LTE and ZigBee communication.

3.2. Register Phase

In this phase, a MTC group will be constructed by a mass of MTCDs and an identity of the MTC group, GID, will be embed into MTCDs in the device initialization process according to the specification made by 3GPP committee [4]. The same MTC group will exist in the same area, and/or have the same MTC features and/or belong to the same MTC user. A group leader of the group will be selected based on the communication capability, communication link quality, storage status and battery status of each MTC. When each MTC in the MTC group registers to the network, it contacts the Key Generate Center (KGC), provides identifiers, and then receives its private key. The MMEs have the same function as the MTCs to obtain these private keys after expiration time. Only the authenticated MTCDs and the MMEs can get the private keys from the KGC. The KGC can be integrated with the HSS, which has pre-established secure channels with the MMEs by using the network domain security (NDS/IP) [28]. This phase can be described as follows. Here, same as Setup in [21], KGC generates $\{p, G_1, G_2, e, P, PK_1, H_1, H_2, H_3\}$ as system parameters and then publishes the system parameters and keeps the master key $x$ secret.

![Figure 3. Register Phase](image_url)

Let $MTCD_1, ..., MTCD_n$ be the MTC group members. As shown in Fig. 3, each $MTCD_i$ executes the following procedure when it first registers to the network.

1.2.3. Step 1, 2 and 3 is similar to the EPS-AKA except that the $MTCD_i$’s GID is added to the identity response message and the authentication data request message, respectively.

4. Upon the receipt of the authentication data request message, the HSS/KGC checks whether $MTCD_i$ is a legal member of the MTC group GID. The HSS/KGC keeps a group list of the identifiers of the group members, which belongs to the same MTC group GID. If $ID_{MTCD_i}$ exists in the GID’s group list, The HSS/KGC generates an ordered array of EPS authentication vectors (AVs) for $MTCD_i$, and computes $Q_{ID_i} = H_1(ID_{MTCD_i} \parallel GID)$. Then the HSS/KGC obtains the private key $S_{ID_i} = xQ_{ID_i}$. Finally, the long-term private key $S_{ID_i}$ will be embed in the authentication data response.
message and sent to the MME. Otherwise, the HSS/KGC sends the request failure message with the cause to the MTC Di.

5.6. Step 5 and 6 is similar to the EPS-AKA except that the MME needs to keep $S_{ID_i}$ secret.

7. After the MME verifies the $MTCD_i$ successfully, the MME derives the key $K_{ASME}$ and encrypts $S_{ID_i}$ with $K_{ASME}$. Then, the encrypted message and public parameters would be piggybacked in user authentication success message and sent to the $MTCD_i$.

8. Upon the receipt of the user authentication success message, the $MTCD_i$ decrypts the encrypted message with $K_{ASME}$ and keep the $S_{ID_i}$ secret.

Note that $K_{ASME}$ is the existing key derived between the MME and $MTCD_i$ during the 3GPP EPS AKA procedure.

Same as the MTCs, when MMEs first register to the network or the secret keys expire, the HSS/KGC also generates the ID-based long term key for them. When the HSS/KGC receives the MME’s identifier $ID_{MME}$, the HSS/KGC works for the MME as follows.

1) Chooses a random number $r \in R Z_p^{*}$, computes $R_{MME} = rP$ and $h = H_2(ID_{MME} \parallel R_{MME})$.
2) Computes $s_{MME} = r + h \cdot e$.

After the MME is verified successfully, The pair $(s_{MME}, R_{MME})$ as the MME’s long-term private key are encapsulated in the last message of Internet Key Exchange Protocol Version 1 (IKEv1) or IKEv2-based [26] device authentication and sent to the corresponding MME securely. Upon the receipt of the message, the MME can validate its long-term private key by checking $s_{MME}P = R_{MME} + H_3(ID_{MME} \parallel R_{MME})PK$.

After the register phase, $S_{ID_i}$ and $(s_{MME}, R_{MME})$ as the long-term secret keys stored in $MTCD_i$ and MME, respectively, will be only employed for the subsequent group-based access authentication. Only when the MME requests the authentication data from the HSS again, the long-term secret keys will be updated by the HSS. In addition, the HSS will add the long-term secret keys to the corresponding nodes subscription information stored in the HSS database.

3.3. Group-based Access Authentication Phase

In this phase, mutual authentications between all of $MTCD_i$ in the group and the MME are accomplished when the entire MTC group attaches to the network simultaneously. And a distinct session key between the MME and each $MTCD_i$ will be created. This process is shown in Fig. 4, which is described in detail as follows.

1. Signature Generation of $MTCD_i$.

In order to access to the network, each $MTCD_i$ executes the following procedures.

1) Chooses $r_{MTCD_i} \in R Z_p^{*}$ and computes $U_i = r_{MTCD_i}P$.

2) Computes $h_i = H_2(ID_{MTCD_i}, GID, T_i, U_i)$ and $V_i = S_{ID_{MTCD_i}} + h_i \cdot T_{MTCD_i} \cdot PK$ where $T_i$ is the timestamp of $MTCD_i$.

Then, $MTCD_i$ generates the access request message $(ID_{MTCD_i}, U_i, V_i, GID, T_i)$ and sends it to the group leader.

2. Aggregation

In this step, all of the signatures will be aggregated to a single signature. The group leader first gathers all of the access request messages from other group members by using ZigBee. Then, it computes $V = \sum_{i=1}^{n} V_i$ and sends the aggregation request message $(ID_{MTCD_1}, ..., ID_{MTCD_n}, U_1, ..., U_n, T_1, ..., T_n, V, GID)$ to the MME.

3. MME Authenticates the MTC Group

On receiving the message from the group leader, the MME works as follows.

1) For each $MTCD_i$ ($i = 1, ..., n$), checks if $T_{exp}$ is expired and $T_i$ is valid. If it holds, then goes to step 2. Otherwise, sends the request failure message with the cause to the group leader to notify the $MTCD_i$ invalid.
2) Computes $h_i = H_2(ID_{MTCD_i}, GID, T_i, U_i)$ and $Q_{ID_i} = H_1(ID_{MTCD_i} \parallel GID)$ for $i = 1, ..., n$.
3) Verifies $\epsilon(V, P) = \epsilon(\sum_{i=1}^{n} [Q_{ID_i} + h_i U_i] PK)$.
4) If it holds, authenticates the MTC group. Then, chooses a new random number $r_{MME} \in Z_p^{*}$ and computes $U_{MME} = r_{MME}P$. By using the MME’s long-term private key $(s_{MME}, R_{MME})$.

5) Generates the session key $SK_{MME} = r_{MME} U_i = r_{MME} T_{MTCD_i} P$ with $MTCD_i$, ($1 \leq i \leq n$) to protect the subsequent communications. Note that the session keys will be distinct because the different $T_{MTCD_i}$ have been sent from distinct $MTCD_i$.

4. Authenticates the MME

Upon receipt of the access response message, the group leader broadcasts the message to each $MTCD_i$. Then each $MTCD_i$ executes the following operation.

1) Computes the MME’s public key $PK_{MME} = H_3(ID_{MME}, R_{MME})PK + R_{MME}$.

<table>
<thead>
<tr>
<th>$MTCD_1, ..., MTCD_n$</th>
<th>Group Leader (MTCA)</th>
<th>eNBI</th>
<th>MME</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{1. Access authentication}$</td>
<td>$(ID_{MTCD_i}, GID, T_i) i \leq n$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{2. Aggregation request message}$</td>
<td>$(ID_{MTCD_1}, ..., ID_{MTCD_n}, U_1, ..., U_n, T_1, ..., T_n, V, GID)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{3. Access response message}$</td>
<td>$(ID_{MTCD_1}, ..., ID_{MTCD_n}, U_1, ..., U_n, T_1, ..., T_n, V, GID)$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4. Group-based Access Authentication Phase**
2) Verifies the signature $\sigma_{MME}$ using $PK_{MME}$ according to the ECDSA verification.

If successful, generates the session key $SK_{MMEi} = r_{MTCDi}U_{MMEi} = r_{MTCDi}f_{MME}P$. Otherwise, sends the response failure message with the cause to the MME.

Note that the MME’s public key can only be computed at the first time and be stored in $MTCDi$. When $MTCDi$ connects to the network again, it can directly use $PK_{MME}$ without any delay.

3.4. Verification Failure

If the verification of aggregate signature is failure, i.e. $e(V, P) \neq e(\sum_{i=1}^{n}[Q_i + h_iU_i], PK)$, there are invalid signatures from rogue $MTCDi$ or attackers. The batch verification and "divide-and-conquer" approach [18] can be employed to detect the invalid signatures quickly, which works as follows.

1. The MME sends a request failure message to the group leader to request all of $V_i (1 \leq i \leq n)$.
2. Upon receiving the messages from the eNB, the MME can divide the $V_1, ..., V_n$ into several subgroups and then separately aggregate each subgroup to a single signature and check the validity of the signature from each subgroup.
3. If the signature from the subgroup is still invalid, the MME executes the same operation as that in step (2).
4. Finally, the MME finds out all of invalid signatures, and then sends the request failure message to these MTCDs via the group leader and refuse the connection with them.

3.5. Dynamic Group Member Management Mechanism

If a MTC device enters/leave to/from the area of a MTC group, or the MTC user wants to control/delete a new/old MTC device, the dynamic group member management mechanism will be applied to deal with this problem, which is described as follows.

Join group: After the new MTCD successfully registers to the network, the MME checks its location, MTC features and the corresponding MTC user. If it satisfies the features of the MTC group GID, then the MME sends the update group list message with the identifier of the new MTCD and GID to the HSS/KGC. Then, the HSS/KGC adds the new identifier to the corresponding group list. Finally, the new MTCD executes the register phase to get its private key.

Leave group: If a MTCD does not meet the characteristics of the relevant group or has been compromised by an attacker, the MME sends the update group list messages with the corresponding identifier and GID to the HSS/KGC when the MTCD enters to the network. Then, the HSS/KGC delets the identifier from the group list and set its private key expire.

4. SECURITY EVALUATION

In this section, security objectives of our scheme are analyzed and then formal verification tools by the TLA+ and TLC are employed to show that the proposed scheme can work correctly to achieve these security objectives.

4.1. Security Analysis

Our solution has the following security objectives in the design.

Mutual authentication: Mutual authentications between the MME and the group of MTCDs can be achieved based on the technique of aggregate signature and ECDSA. On the one hand, to be authenticated by the MME, each MTCD in the group computes the signature pair $(U_i, V_i)$ by using its long-term private key. Then, the aggregate signature easily generated by sum of all the signatures from the group will be checked by the MME. Only the legitimate MTCDs can generate the valid signature and then obtain the legitimate aggregate signature. Once invalid signatures exist in the aggregate signature, the aggregate signature is not valid. Thus, only the aggregate signature is validated, all of MTCDs in the group can be trusted by the MME. On the other hand, each MTCD in the group can authenticate the MME by checking the ECDSA signature.

Session key establishment: To protect the communication over air interface between the MME and each MTCD in the group, the distinct session key will be negotiated with different MTCD by using the Elliptic Curve Diffie-Hellman (ECDH) key agreement protocol. The random points $U_{MME}$ and $U_i$ serving as the exchange key parameters are chosen by the MTCDs in the group and the MME, respectively. Due to the difficulty of Computational Diffie-Hellman (CDH) problem, only the legitimate MME and $MTCDi$ can generate the session key $SK_{MME}$. Key Forward Secrecy/Backward Secrecy (KFS/KBS) and Non-repudiation verification: KFS and KBS can be achieved by our solution which is an important property in wireless communication. KFS and KBS means that even if a long-term secret key is compromised, the preceding and succeeding session keys still remains secret. By our scheme, even if the long-term secret key is stolen, the preceding and succeeding session keys could not be exposed because the freshness of DH session keys is guaranteed by the random values of $MTCDi$ and the MME. In addition, owing to the use of digital signature, our scheme can provide non-repudiation verification in the contestable business.

Withstanding protocol attacks: Since the session key is established based on the CDH problem by our scheme, a Man-in-the-Middle (MitM) adversary or other MTCDs in the MTC group could not obtain the session key of $MTCDi$ using the public values from the communication channel between $MTCDi$ and the MME. Without the long-term private keys, the attacker could not forge a valid ECDSA signature and an aggregate signature to deceive $MTCDi$ or the MME. Thus, it is infeasible for
an adversary to make impersonation attacks by modifying the public values. In addition, our scheme can withstand the replay attacks by using the time stamp $T_i$.

**Signaling congestion avoidance:** Our scheme can avoid signaling congestion in terms of low signaling overhead and fast verification. To avoid signaling overload, a mass of access request messages from a group of MTCDs will be aggregated into a single request message by a group leader and then the single message is sent to the MME. In addition, the MME only computes a signature and sends it to the group leader via a single signaling message. Therefore, it can largely reduce the signaling overhead and alleviate the burden of the MME. To have a fast verification, the MME can simultaneously authenticate a group of MTCDs by adopting the technique of aggregate signature and quickly create the session keys with the MTC group. Thus, our scheme can ensure QoS for the MTC users without restriction on the access requests. Furthermore, the more devices form the group, the better the performance advantages emerge.

### 4.2. Formal Verification

In this paper, we examine various security properties of the proposed scheme by a formal verification toolkit called Temporal Logic Actions (TLA+) [29] and an automatic model-checker tool called Temporal Logic Checker (TLC) [30]. TLA+ is a formal specification language to describe and reason about asynchronous, nondeterministic, concurrent systems based on Zermelo-Frankel set theory, first-order logic and the linear-time temporal logic TLA. The TLA+ model checker, called TLC, is a state-of-the-art model analyzer that enumerates the reachable states in a finite-state model of a specification written in an expressive subset of TLA+, and checks that invariance properties written in TLA+ hold in each of these states. When the TLC discovers an error, a minimal-length sequence of states leading from an initial state to the state with error will be reported. Following the way of the verification in [31], firstly the model of the proposed scheme, the attacker model and the security properties to be checked are specified in TLA+ language. Then the TLC will be executed to explore the entire state space of the model of the proposed scheme. If there are attack behaviors that violate the properties found by TLC, it will output the attack trace. Otherwise, all of the states of the model will be reached without a state left, and thus security properties will be approved to achieve.

The specific procedure is described as follows.

1. **General Mode**

In general, the canonical form of a TLA+ formula describing a network protocol is:

$$\text{Protocol} = \text{Init} \land \bigwedge_{x} [\state(x) \land \text{Next}(x)] \land L$$

where $\text{Init}$ is a predicate describing the initial state, $\text{Next}$ is an action describing the next-state relation, is the set of all program variables, and $L$ is a liveness property written as the conjunction of fairness conditions on actions. The conjunct $\bigwedge_{x} [\state(x) \land \text{Next}(x)]$ asserts that every step is either a Next step or else leaves all variables unchanged. A protocol may include multiple participants, for example many MTCDs and a MME in the model of the proposed scheme. Its TLA+ specification can be described:

$$\text{Protocol} = \text{MTCD}_1 \land \ldots \land \text{MTCD}_n \land \text{MME}$$

where all of participants are served as independent systems with their own initial states and next-state relations. They can communicate mutually with each other by means of their shared variables.

For a security protocol, it is generally assumed that it is secure in ideal environment but may have problems under attacks. Therefore, an attacker can be specified with appropriate capabilities as $\text{Attacker}$ in TLA+ and the requested secure properties of the protocol will be formally described as $\text{Property}$ in TLA+. The attacker should be at least as powerful as a set of cooperating attackers. And the security properties in the traditional correctness verification should be checked to show the robustness of the protocol under attacks. These security properties such as secrecy and authentication are meaningful only when there is an attacker. As a result, the checking of security properties of the protocol under attacks in TLA+ can be stated as:

$$\text{Protocol} \land \text{Attacker} \Rightarrow \text{Property}$$

2. **Protocol Specification.**

To describe definitely the model of the proposed scheme in TLA+, first, an $EC$ model should be additionally established to define the elliptic curve $E$ over a prime finite field $F_p$ and all of the operations under the elliptic curve group including point addition $P_A$, point multiplication $PM$, Tate pairing $TP$, the three hash functions $H1$, $H2$ and $H3$, and the ECDSA signature $EC\text{Sign}$ and verification $EC\text{Verify}$. Then, the proposed scheme can be specified. As all of the MTCDs in the same group can independently accomplish the authentication and creates its session key with the MME, there is no interaction between them. In order to simplify the process, we assume that there are only two MTCDs, $\text{MTCD}_1$ and $\text{MTCD}_2$ in the MTC group with $\text{MTCD}_2$ to be the group leader. Moreover, in the group access authentication phase, since the eNB only transfers the messages from the group leader or the MME without other operations and has established a secure channel with the MME, the operation of it can be combined with that of the MME. Therefore, the model of the proposed scheme will mainly include three basic participants: $\text{MTCD}_1, \text{MTCD}_2$ (the group leader) and the MME, and three messages which are described as follows.

1. **ACCESS REQ:** $\text{MTCD}_1 \rightarrow \text{MTCD}_2 : (ID_{\text{MTCD}_1}, U_1, V_1, GID, T_1)$
2. **AGGREGATION REQ:** $\text{MTCD}_2 \rightarrow \text{MME} : (ID_{\text{MTCD}_1}, ID_{\text{MTCD}_2}, U_2, T_1, T_2, GID, V)$
3. **ACCESS RES:** $\text{MME} \rightarrow \text{MTCD}_2, \text{MTCD}_2 \rightarrow \text{MTCD}_1 : (ID_{\text{MME}}, R_{\text{MME}}, U_{\text{MME}}, T_{\text{MME}}, \sigma_{\text{MME}})$

Subsequently, the corresponding initial state predicate and the state transition actions for each participant should
be specified. As shown in Fig. 5, we take the MTCD1 as an example to show the state transition relations. The initial state state of MTCD1 is s0. After the initialization, MTCD1 sends ACCESS REQ message to the MTCD2 and the state state is changed to s1. When MTCD1 receives ACCESS RES from the MME via the MTCD2, MTCD1 validates the signature by the ECverify operation. If it holds, the state state will be changed to s2 or s2', where if an attacker forges a legal signature by its own random number, the state state will be s2', otherwise it will be s2. If the signature is not correct, the state state will be s2''. When the state state has been changed to s2'', MTCD1 will change its state state to s0 and restart this process.

Figure 5. The State Transition Relations of MTCD1

3. Attacker Specification

The Attacker model is similar to the Dolev-Yao intruder model. By this model, an intruder has full control over the communication channel via the air interface so that all messages sent by the participants will go to the intruder. The intruder can intercept and store the messages on the channel, replay old messages, analyze and modify messages as far as it knows the required keys, and can send any new messages that it composed to whoever at its will, impersonating other agents.

Since both the communication between MTCDs and the group leader and that between the group leader and the MME are over the wireless interface, the adversary can invade the two channels simultaneously. Without loss of generality, we make the following assumptions in the design of the model.

1. The ideal elliptic curve cryptography will be carried out. For example, it is assumed that scalar multiplication, Tate pairing and hash functions are secure nonreversible functions and thus, intruders cannot reverse to get the secret information.

2. The random numbers cannot be derived by guess or exhaustion. To avoid the state space explosion, which is one of the most common issues faced in formal verification process, a small prime p has been selected and let the random numbers be constant values to restrict the state space to a small finite one, which is no impact on property checker.

3. The maximal number CNTN of messages that the attacker can corrupt is limited. The scenarios where an adversary continuously corrupts messages since they can be easily detected in reality will be ignored.

4. Property Specification

In this phase, the requested security properties will be specified. According to our security analysis, the following security properties are stated in TLA+ language as shown in Fig. 6.

1) Mutual authentication (Mulauth): It shows that MTCD1, MTCD2 and MME should simultaneously reach the state "Muth_SUCCESS" infinite times. It shows that MTCD2 and MTCD3 trust MME iff MME trusts MTCD1 and MTCD2.

2) Withstanding protocol attacks (Attackerstate): It shows that none of the participants MTCD1, MTCD2 and MME could reach the state "IN_SUCCESS" at any time. If the property is violated, the adversary can successfully attack the proposed scheme.

3) Secrecy of (secrecy): R1, R2 and Rm are the secret random numbers generated by MTCD1, MTCD2 and MME, respectively. At any time, the intruder cannot obtain any one of these random numbers. Since R1, R2 and Rm are only known by MTCD1, MTCD2 and MME respectively, the session keys SK_SMME, and SK_MME are secret.

Figure 6. Security Properties

5. PERFORMANCE ANALYSIS

In this section, we first evaluate the performance of the GBAAM by comparing it with the existing protocols including EPS AKA [7], GAKA [10], DGBAKA [11, 12], MTC-AKA [13], SE-AKA [14], EG-AKA [15], LGTH [16] and ABAKA [18] in terms of the signaling cost and the communication cost. Then, we further analyze the computational cost of the GBAAM. Due to the use of symmetric cryptography, the authentication latency of the EPS AKA, GAKA, DGBAKA, MTC-AKA and LGTH may be nearly zero. Thus, only the GBAAM, the ABKAS, the SE-AKA and the EG-AKA will be evaluated on the
computational cost. Finally, we illustrate the verification cost of the GBAAM when the verification of aggregate signature is failure. It is assumed that $n$ MTCs form a group and access authentication is performed $t$ times by each MTC. Since mutual authentication and key agreement between the devices and the network are not accomplished in DGA [9], OBBA [17] and KGM [19], we ignore them in the performance evaluation. The notation used in the performance analysis are defined in Table III.

### 5.1. Signaling Overhead

On the signaling overhead, we mainly compare the GBAAM with the schemes in [7, 10–16, 18] in terms of the number of signaling messages. For the EPS AKA [7], since the MME can derive $m$ EPS authentication vectors (AV) from the HSS for a MTC in the first access authentication procedure, the MME can subsequently perform an authentication process with the MTC immediately using an AV retrieved from the HSS without contacting HSS. Therefore, the signaling cost incurred in the GBAAM is the same as that in the LGTH, which is much less than that of other schemes. Compared the two pictures, the more MTCs ($n$) in a group and the authentication times ($t$), the better performance advantages emerge.

### 5.2. Communication Overhead

In this section, the communication overhead of the GBAAM will be analyzed compared with other schemes. It is assumed that the transmission cost incurred by delivering an authentication packet between the MTC and the MME is one unit, the cost between the MME and the HSS or the KGC is fraction $a$ of a unit, the cost between the MTCs and the group leader is fraction $b$ of a unit, respectively. The signaling cost incurred in the GBAAM is the same as that in the LGTH, which is much less than that of other schemes.

![Graph showing the total number of signaling messages](image.png)

Figure 8. Comparison of the Number of Signaling Messages When $t=10$

Table III. Definition of Notation in the Performance Analysis

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>Number of authentications performed by each MTCD</td>
</tr>
<tr>
<td>$m$</td>
<td>Number of authentication vectors derived by MME before the need to re-contact HSS</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of the MTCs in a group</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Transmission cost between the MME and the HSS or the KGC</td>
</tr>
<tr>
<td>$b$</td>
<td>Transmission cost between the MTCs and the group leader</td>
</tr>
<tr>
<td>$T_{\text{mul}}$</td>
<td>Cost of a point multiplication</td>
</tr>
<tr>
<td>$T_{\text{pair}}$</td>
<td>Cost of a pairing operation</td>
</tr>
<tr>
<td>$T_{\text{mtp}}$</td>
<td>Cost of a map to point hash operation</td>
</tr>
<tr>
<td>$T_{\text{fir}}$</td>
<td>Cost of first verification</td>
</tr>
<tr>
<td>$T_{\text{rv}}$</td>
<td>Cost of a re-verification</td>
</tr>
</tbody>
</table>
Table IV. Comparison of Signaling Overhead

<table>
<thead>
<tr>
<th>Scheme</th>
<th>The first access process</th>
<th>The subsequent (t-1) process</th>
<th>The total</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPS-AKA [7]</td>
<td>7n</td>
<td>5n(t-1)</td>
<td>7n+5n(t-1)</td>
</tr>
<tr>
<td>GAKA [10]/DGBAKA [11, 12]/MTC-AKA [13]/SE-AKA [14]/EG-AKA [15]</td>
<td>2+5n</td>
<td>5n(t-1)</td>
<td>2+5nt</td>
</tr>
<tr>
<td>LGTH [16]</td>
<td>7n</td>
<td>(n+3)(t-1)</td>
<td>7n+(n+3)(t-1)</td>
</tr>
<tr>
<td>ABAKA [18]</td>
<td>2n</td>
<td>2n(t-1)</td>
<td>2nt</td>
</tr>
<tr>
<td>GBAAM</td>
<td>7n</td>
<td>(n+3)(t-1)</td>
<td>7n+(n+3)(t-1)</td>
</tr>
</tbody>
</table>

order to facilitate the analysis, it is assumed that $a = 0.2$ and $b = 0.01$. Fig. 10 and Fig. 11 show that the results for the communication cost with the change of the number of MTCDs when $t = 10$ and $t = 20$, respectively. From the Fig. 10 and Fig. 11, it is obvious that the communication cost of the GBAAM is same as that in the LGTH, which is much lower than that of other schemes.

5.3. Computational Cost

For the computational cost, only the cost of the following operations will be considered including a point multiplication $T_{mul}$, a pairing operation $T_{pair}$, and a map to point hash operation $T_{mtp}$ while other operations such as point addition and one-way hash function will be ignored. According to the scheme in [32], the time costs of the primitive cryptography operations are shown in Table VI by using C/C++ OPENSSL library [33] on a Celeron 1.1GHz processor as a MTCD and a Dual-Core 2.6GHz processor as a MME, where $T_{mtp}$ takes the same time as $T_{mul}$ [18]. When $n$ devices simultaneously send access request messages to the network, the comparison of the computational cost in the reference schemes is shown in Table VII. From the Table VII, the computational cost of each device by the GBAAM is much less than that by the ABAKA [18]. Fig. 12 shows the analysis results for the computational cost of the network (i.e. MME in our scheme). According to the Fig. 12, the computational overhead of the network by the GBAAM is approximate to that by the ABAKA with the increase of MTCDs. Note that due to the integration of symmetric cryptography and ECDH algorithms in SE-AKA and EG-AKA, the computational cost by the SE-AKA and EG-AKA is much better than that by the GBAAM and the ABAKA.
5.4. Verification Cost

Owing to the aggregation signature, the GBAAM can largely reduce the signaling cost and achieve low communication overhead and computational overhead. However, once an invalid access request exists in the aggregation signature generation, the GBAAM may not have advantages in efficiency. To improve the efficiency, a batch verification mechanism and a “divide-and-conquer” approach have been adopted so that the invalid access requests can be quickly detected. V2V communication researched in [18] is a special type of M2M communication. In comparison with other MTCDs, vehicle is more vulnerable to packet loss or sending bogus messages due to its characteristics of the mobility. In order to illustrate the verification cost of the GBAAM in the worst case, we have employed the analysis results of the probability of invalid requests in [18]. According to the scheme in [18], the probability of only one invalid access request in a batch is 0.42 and that of two invalid requests is 0.18, while the probability of more than two invalid requests in a batch is almost negligible, which is approximately lower than 0.06. We elaborately evaluate the verification cost of the GBAAM when there are one or two invalid access requests. By the GBAAM, the MME has to compute all of the related private keys of or two invalid access requests. By the GBAAM, the MME has to compute all of the related private keys of MT devices in the first verification \( T_{fir} \), which takes \( T_{fir} = nT_{mtsp} + nT_{mal} + 2T_{pair} \), while a re-verification \( T_{re} \) needs only \( 2T_{pair} \). To be precise, there are two cases for the batch verification by using the proposed detection process, which are worst case and average case [18] described as follows.

In the worst case, a valid signature is always with the invalid signature in the same batch (i.e. subgroup) until the last group division. In this case, the detection process with binary divisions has to execute at most \( \lceil \log_2 n \rceil \) re-verifications. Thus, the total verification cost for an invalid signature \( T_{worst1} \) and that for two invalid signatures \( T_{worst2} \) in this case are as follows.

\[
T_{worst1} = T_{fir} + 2\lceil \log_2 n \rceil T_{re} \quad (1)
\]

\[
T_{worst2} = T_{fir} + 2T_{re} + 4(\lceil \log_2 n \rceil - 1)T_{re} \quad (2)
\]

The average case can be obtained as the total verification delay in all possible cases divided by the number of possible cases. Then, the total verification cost for an invalid signature \( T_{ave1} \) and that for two invalid signatures \( T_{ave2} \) in this case are as follows.

\[
T_{ave1} = \frac{1}{\lceil \log_2 n \rceil + 1} \sum_{i=1}^{\lceil \log_2 n \rceil} (T_{fir} + 2iT_{re}) \quad (3)
\]

\[
T_{ave2} = \frac{1}{\lceil \log_2 n \rceil + 1} \sum_{i=1}^{\lceil \log_2 n \rceil} (T_{fir} + 2T_{re} + 4(i - 1)T_{re}) \quad (4)
\]
Fig. 13 and Fig. 14 show the results of the total verification cost in the worst case and in the average case for an invalid request and two invalid requests, respectively. According to the Fig. 13 and Fig. 14, when there are 4500 MTCDs, the total verification cost in the average case is less than 5000 ms, i.e. verifying a MTCD takes approximately 1.2 ms on average even if there are one or two invalid access requests.

Note that the communication between the MTC group and the MTC server in MTC service scenarios can start after the group-based access authentication procedures are accomplished.

In general, on the signaling and communication cost, since a lot of access request messages from a group can be aggregated into an aggregation request message and then are authenticated by the MME directly by the aggregation signature with only 2 message exchanges between the group leader and the MME without contacting the HSS except for the registration, the signaling overheads and communication overheads can be much reduced, and thus the signaling congestion can be avoided at the network nodes such as MME and HSS. In addition, a lot of access request messages are transmitted inside the MTC group by using other technologies such as Zigbee, there are two message exchanges between the eNB and the MTC group, and thus, the radio access network signaling congestion can also be avoided. Due to the use of the same idea, the LGTH [16] has the same advantage with the GBAAM in terms of signaling cost and communication cost. However, the LGTH requires an establishment of group message management list and involve complex group key management mechanisms, which increase the complexity of the entire system. In addition, there is no dynamic group member management mechanism in the LGTH, which will be impracticable for MTC in LTE networks. On the computational cost, compared with the SE-AKA and the EG-AKA, which have adopted various pre-distribution and group key management mechanisms for mutual authentication and employed the ECDH algorithms for key agreement between the MTCD and the network, the GBAAM incurs more computational overheads due to the employment of aggregate signature, which needs some public key operations such as point multiplication operations and pairing operations to bring much more operation time compared with symmetric key operations and a few of point multiplication operations employed by the SE-AKA and the EG-AKA. The objective of the proposed solution is to reduce the signaling cost and the communication cost and enhance the security of the EPS-AKA, and thus we have adopted the technique of the aggregate signature. As a result, it could be an issue to affect the deployment of the LTE networks for MTC. However, the GBAAM can simplify the access authentication procedure, where mutual authentication can be directly achieved by the use of MTCD and MME’s private key without involvement of the complex group key management mechanisms. In addition, by adopting a digital signature, we can provide non-repudiation verification in the contestable business. Therefore, our scheme is much more secure and simpler than the SE-AKA, EG-AKA and other AKA schemes in [7,10–13,16] and can meet the reliability and flexibility of the next generation network. Compared with the ABKA, although the network requires two extra pairing operations by the GBAAM compared with only one point multiplication operation by the ABKA, both of them need to take 3n point multiplication operations. With the increase of the MTCD n, the constants can be ignored. Thus, the computational cost incurred by the GBAAM approximates to that by the ABKA. On the verification cost, the GBAAM takes only two pairing operations in a re-verification, which is constant, and thus does not incur too many verification costs with the increase of the number of MTCDs even if there are one or two invalid access requests.
6. CONCLUSION

In the MTC supported by the LTE networks, a large number of MTCs simultaneously activated to access to the network will cause network overload. Moreover, each device has to perform a full EPS-AKA process to achieve the access authentication resulting in a severe signaling congestion at the network nodes such as MME and HSS. In this paper, we have proposed a group-based access authentication scheme by the technique of aggregation signature. By our proposed scheme, the network can simultaneously trust the mass of MTCs and generate independent session key with each MTC. Our analysis indicates that the proposed scheme can provide robust security protection and avoid the signaling congestion at main network nodes.

7. ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No. 61402354, U1401251, 61402351, and 61303218), the Fundamental Research Funds for the Central Universities(No. 7214605301 and 7214605303). And it is also supported by The Open Subject for the State Key Laboratory of Information Security (No. 2015-MS-15).

REFERENCES

6. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Security Aspects of Machine-Type Communications (Rel 12), 3GPP TR 33.868 V0.15.0, Sept. 2013.


