Computer Networks 99 (2016) 66–81

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

GLARM: Group-based lightweight authentication scheme for resource-constrained machine to machine communications

Chengzhe Lai[a,b,*], Rongxing Lu[c], Dong Zheng[a,*], Hui Li[b], Xuemin (Sherman) Shen[d]

[a] National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an, 710121, China
[b] State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, 710071, China
[c] School of Electrical and Electronics Engineering, Nanyang Technological University, 639798, Singapore
[d] Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada

ARTICLE INFO

Article history:
Received 20 November 2015
Revised 11 January 2016
Accepted 3 February 2016
Available online 12 February 2016

Keywords:
Machine to machine (M2M)
Machine-type communications (MTC)
Group authentication and key agreement
Security
Lightweight
The 3rd generation partnership project (3GPP)

ABSTRACT

Supporting a massive number of machine to machine (M2M) devices has been considered as an essential requirement in M2M communications. Meanwhile, cyber security is of paramount importance in M2M; if M2M devices cannot securely access the networks through efficient authentication, all applications involving M2M cannot be widely accepted. One of research challenges in M2M is group authentication since a large number of M2M devices accessing the network simultaneously will cause a severe authentication signaling congestion. To solve this problem, as well as reduce authentication overhead of the previous schemes based on public key cryptosystems, we propose a novel lightweight group authentication scheme for resource-constrained M2M (GLARM) under the 3GPP network architecture, which consists of two protocols that can achieve efficient and secure group authentication in the 3GPP access case and non-3GPP access case, respectively. GLARM can not only authenticate all M2M devices simultaneously, but also minimize the authentication overhead. The security analysis shows that the proposed scheme can achieve the security goals, and prevent the various security threats. In addition, performance evaluation demonstrates its efficiency in terms of computation complexity and communication overhead.

1. Introduction

Machine-to-machine (M2M) communications [1], also named machine-type communication (MTC) [2], which is standardized by the 3rd generation partnership project (3GPP). M2M is an emerging technology empowering full mechanical automation (e.g., in the smart grid, smart transportation, etc.), and its rapid development will change our living styles vigorously. The M2M technology is drawing overwhelming attention in the standardization and industry areas, which has been actively engaged in by many standards forums and organizations, including IEEE, European Telecommunications Standards Institute (ETSI), Third generation partnership project (3GPP) and 3GPP2. Among them, the 3GPP MTC has been regarded as the promising solution facilitating M2M communications [3].

With the development of M2M technology, the requirements of efficiency, reliability and security (GRS) are being paid more attention by mobile network operators and research groups [4]. For *efficiency*, low power M2M devices have been highly attractive in many scenarios due to the

* Corresponding authors. Tel.: +86 13468922875.
E-mail addresses: lcz_xupt@163.com, lcz.xidian@gmail.com (C. Lai), rxlu@ntu.edu.sg (R. Lu), zhengdong@xupt.edu.cn (D. Zheng), lihui@mail.xidian.edu.cn (H. Li), sshen@uwaterloo.ca (X. (Sherman) Shen).

http://dx.doi.org/10.1016/j.comnet.2016.02.007
1389-1286/

fact that they can be deployed in a wide range of applications and easily retrofitted, thus significantly reducing installation costs. Moreover, thousands of low power M2M devices can work unattended for years, hence they need to be deployed and maintained at a very low cost. However, many traditional communication protocols cannot be applied to these resource-constrained M2M devices since they introduce too much overhead.

Cyber security is also of paramount importance in M2M communications; if M2M devices cannot securely access the networks through efficient authentication, all applications involving M2M cannot be widely accepted. However, the recent authentication and key agreement (AKA) protocols dedicated for 3GPP Evolved Packet System (EPS), known as EPS-AKA [5]; or for non-3GPP access networks (e.g., WLAN or WiMAX), known as EAP-AKA [6] cannot provide enough security [7–10]. In addition, to support M2M communications, the 3GPP mobile operator has to accommodate its network to support a large number of MTC devices, which can overload its network resources and introduce congestion in the network at both the data and control planes [11]. In fact, congestion may occur due to simultaneous authentication signaling messages from M2M devices. For instance, if a group of M2M devices detect a base station, they send their access authentication requests toward the core network at the same time, leading to congestion in the different nodes of the network, across the communication path. If a large number of M2M devices in a group need to access the network simultaneously, the traditional authentication protocols (e.g., EPS-AKA or EAP-AKA) will suffer from high signaling overhead, leading to authentication signaling congestion and decreasing the Quality of Service (QoS) of the network. The reason is that every device must perform a full AKA authentication procedure with the home authentication server, respectively. Considering *reliability*, these traditional AKA protocols are not suitable for large-scale M2M communications.

In order to speed up the process of authentication and avoid authentication signaling congestion in group-based communications, some batch verification schemes based on bilinear pairing have been proposed, such as [12–14], etc. Although these schemes can effectively authenticate a group of devices at the same time, they may not be suitable for resource-constrained devices, because the communication is an “expensive” resource due to its effect on the battery life of resource-constrained devices and the bandwidth of the channel. To fulfill the requirements of efficiency, reliability and security, a more efficient group authentication protocol dedicated for M2M devices in 3GPP networks is desirable.

In this paper, we focus on the problem of group AKA for resource-constrained M2M devices in 3GPP networks. On one hand, to address the issue of authentication signaling congestion, the core network needs to authenticate all M2M devices in a group at one time; on the other hand, the novel protocol should aim to reduce the authentication overhead in previous schemes. Therefore, we propose a lightweight group authentication scheme for resource-constrained M2M in 3GPP networks to achieve these goals. The proposed scheme can authenticate all MTC devices simultaneously based on aggregate message authentication codes (MACs). In addition, the proposed scheme considers both 3GPP access and non-3GPP access cases. The main contributions of this paper are as follows:

(1) We propose a novel group-based lightweight authentication scheme for resource-constrained M2M (GLARM) under the 3GPP network architecture, which consists of two protocols that can achieve efficient and secure group authentication in the 3GPP access case and non-3GPP access case, respectively. In addition, our scheme is developed based on the 3GPP standard and thus can be compatible with the 3GPP standard protocols.
(2) GLARM fills security requirements in previous protocols. Moreover, it can successfully resist some sophisticated attacks, such as redirection [8], man-in-the-middle attacks [7], etc. GLARM can implement mutual authentication and key agreement between multiple M2M devices and the core network simultaneously. Meanwhile, the network congestion because of mass M2M device connections can be avoided in the 3GPP networks and the QoS requirements can be guaranteed.
(3) Due to hardware-limited resources, the low-cost computation and communication authentication scheme is required. Since public key cryptosystems usually execute more computations, we adopt symmetric cryptosystems to reduce the computation cost. Furthermore, the number of round trips between M2M devices and the core network is reduced, which decreases the communication cost.

The remaining of this paper is organized as follows. In Section 2, we preview the related works. In Section 3, we introduce our network architecture, security and design goals. In Section 4, we present our GLARM, followed by its security analysis and performance evaluation in Sections 5 and 6, respectively. Finally, we draw our conclusions in Section 7.

2. Related works

There have been many research works on authentication and key agreement protocol in 3GPP networks.

For 3GPP access network (e.g., UMTS, LTE or LTE-Advanced), in 2005, Zhang and Fang [9] point out that 3GPP AKA has some security weaknesses. The first weakness is that it is vulnerable to a variant of false base station attack, which allows an adversary to redirect user traffic from one network to another. The second weakness is that it allows an adversary to use the authentication vectors (AVs) corrupted from one network to impersonate the other networks. The third weakness is that the use of synchronization between a mobile station (MS) and its home network (HN) incurs resynchronization. To overcome these weaknesses, Zhang and Fang propose an improved authentication and key agreement protocol called AP-AKA.

In 2010, Ou et al. [15] propose Cocktail-AKA to overcome the congenital defects of UMTS-AKA. Cocktail-AKA

uses two varieties of AVs (called MAV and PAV) to produce several effective AVs. In the protocol, each service network produces its own AVs (MAVs) in advance. These MAVs are produced only once but can be reused later. While authenticating the MS, the HLR/AuC calculates a private authentication vector (PAV) for MS. The PAV is transferred to the SGSN. Then, the SGSN uses the PAV and MAV to generate several effective AVs for subsequent authentications. Unfortunately, Cocktail-AKA is vulnerable to denial-of-service (DoS) attacks [16]. In 2011, Huang et al. [10] introduce a secure AKA (S-AKA) protocol which can resist the typical attacks and they also give the formal proof of the S-AKA protocol to guarantee its robustness. However, these existing protocols are not suitable for group-based communications due to lack of special group access authentication mechanism.

Chen et al. [17] propose a group authentication and key agreement protocol (G-AKA) for a group of MSs roaming from the same home network to a serving network. The protocol optimizes the performance of authentication of group communications, however, it cannot provide enough security and is vulnerable to redirection, man-in-the-middle attacks, etc. In our previous works [18,19], we also propose two group-based authentication protocols, which can not only reduce the authentication cost when a group of MSs access the network, but also can provide more security services. Nevertheless, they cannot authenticate all devices simultaneously.

For non-3GPP access network (e.g., WLAN or WiMAX), EAP-AKA' [20] or EAP-AKA [6] are the standard solutions for mutual authentication of the user equipment (UE) and the authentication, authorization and accounting (AAA) server, and for key agreement procedure for data protection in the radio-link layer [21]. Then, a lot of works [22–27] have been proposed to solve the security or performance issues of EAP-AKA protocol. Our recent works [28,29] aim to provide group access authentication based on EAP-AKA protocol. Similarly, they cannot authenticate all devices simultaneously.

In order to authenticate all devices simultaneously and avoid authentication signaling congestion in group-based communications, some batch verification schemes based on public key cryptosystems have been proposed. Huang et al. [30] introduce an anonymous batch authenticated and key agreement (ABAKA) scheme in vehicular ad hoc networks. ABAKA can efficiently authenticate multiple requests by one verification operation and negotiate a session key with each vehicle by one broadcast message. Cao et al. [13] propose a group-based authentication and key agreement for MTC in LTE networks, which can make multiple MTC devices be simultaneously authenticated by the network. However, this scheme adopts an ID-based aggregate signature scheme, which is considered to be suitable only for private networks because of malicious PKG problem. To overcome this problem, we [14] propose a new secure and efficient group roaming scheme for group-based MTC, named SEGR, by utilizing the certificateless aggregate signature. Although these schemes can effectively authenticate a group of devices at the same time, they may not be suitable for resource-constrained devices due to the large authentication costs.

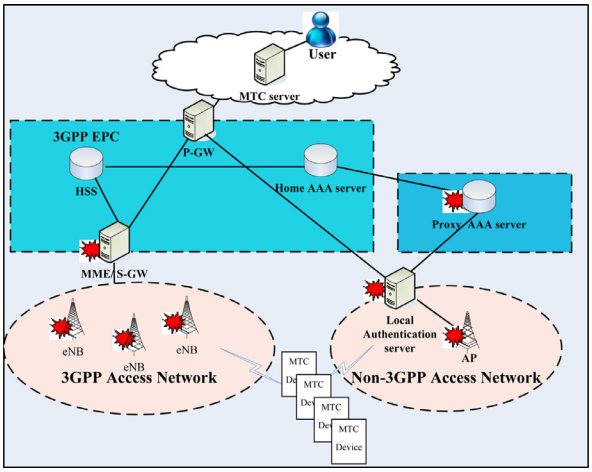


Fig. 1. Network architecture.

3. Network architecture, security and design goals

In this section, we introduce the network architecture, and identify our security and design goals.

3.1. Network architecture

As shown in Fig. 1, our considered network architecture is based on 3GPP standard, and can be divided into three domains:

(1) Access networks, including 3GPP access network, which is composed of eNodeBs (eNBs), and non-3GPP access network, which consists of wireless points (APs) and local authentication server (LAS);
(2) Evolved Packet Core (EPC), including mobile management entity (MME) (or AAA server) that performs access authentication function, home subscriber server (HSS), serving gateway (S-GW) and packet data network gateway (P-GW);
(3) Non-3GPP domain, e.g., Internet. In our considered network architecture, MTC server is located outside the EPC.

The congestion could happen at different locations:

- The radio part as a lot of MTC devices are connected to the same eNB/AP and consequently use the same channels leading to high contention.
- The EPC, mainly in: (i) the MME (or LAS), which is responsible for managing the attachment of devices to the network; (ii) the S-GW in charge of carrying the traffic; (iii) the P-GW as a lot of MTC devices will send and receive data through it.

Our proposed scheme focuses on the congestion issue in the entities that used to transmit authentication messages, i.e., eNBs/APs, the MME/LAS/AAA server.

3.2. Security and design goals

Our goal is to propose a lightweight group access authentication scheme for resource-constrained M2M in

Table 1
Protocol notation.

Notation	Definition
r_x	The rand number generated by x
ID_x	The identity of x
TID_x	The temporary identity of x
$K_{G_{i-j}}$	The shared secret key between the jth MTCD and HSS in Gi
GK_i	The group key of the ith group
GTK_{Gi}	The group temporary key of the ith group
IK	Integrity key
CK	Cipher key
MSK	Master session key
AK	Authentication key
MAC_x	Message authentication code computed by x
MAC_{G_i}	Message authentication code computed by $MTCD_{leader}$ in Gi
$XRES_x$	Expected response computed by x
$XRES_{G_i}$	Expected response for Gi computed by HSS
RES_x	Authentication response computed by x
RES_{G_i}	Authentication response computed by $MTCD_{leader}$ in Gi
$AUTH_x$	The authentication token generated by x
LAI	Location area identification
AMF	Authentication management field
$f^1 - f^5$	Authentication and key generation function

3GPP networks. In particular, the following goals should be achieved.

The security requirements should be guaranteed in the proposed GLARM. The security requirements includes: (i) Entity Mutual Authentication and Secure Key Agreement, and (ii) Attacks resistance. For (i), All M2M devices must be authenticated successfully by the core network, after successful authentication, the secure channels should be established between all M2M devices and the MME/LAS, respectively; For (ii), GLARM should resist all the existing attacks including redirection, man-in-the-middle, etc.

Efficiency. As described above, all devices in a group must be authenticated by the core network at the same time, the authentication signaling congestion should be avoided, and the computation complexity and communication overhead should be reduced.

4. The proposed GLARM scheme

In this section, we present our GLARM scheme, which consists two protocols (i.e., GLARM-1 and GLARM-2) and each protocol has two phases: initialization phase, and group authentication and key agreement phase. In the GLARM, the core network can authenticate all MTC devices in a group simultaneously based on a novel technology, named aggregate message authentication codes (AMACs), which is proposed by Katz et al.[31], and extended in [32,33].

4.1. Initialization phase

The initialization phase works as follows, and the used notations in the proposed GLARM protocol are shown in Table 1.

- Each MTC device has an identity (ID_{Gi-j}), which is a private identity that identifies MTC device and should be installed in the MTC device by the supplier in order to allow the MTC device to register in a 3GPP network.
- Each MTC device has a pre-shared secret key (K_{Gi-j}) with HSS when it is first registered in HSS.
- Each MTC device calculates its temporary ID TID as follows

$$\kappa = TID_{Gi-j} = E_{K_{G_{i-j}}}(ID_{Gi-j}). \quad (1)$$

  Then, the HSS store κ when the MTC device is first registered in HSS.
- The MTC devices form groups based on certain principles (e.g., belong to the same application, within the same region, etc.), then the supplier provides a group identity (ID_{Gi}) and a group key (GK_i) to each group for authentication.

4.2. Group authentication and key agreement phase

4.2.1. 3GPP access case (GLARM-1)

A secure communication channel between the MME and the HSS has already been established (based on Diameter protocol [34]) and can provide security services to the transmitted data. We assume that MTC devices in a group, without loss of generality, defined as G1. Firstly, a group leader of MTCDs in the group ($MTCD_{leader}$) will be selected [1]. When MTC devices in G1 detect the eNB, the proposed GLARM-1 works as follows and is shown in Fig. 2.

- ***Step-1:*** Each MTC device calculates its MAC as

$$MAC_{MTCD_{G1-j}} = f^1_{K_{G1-j}}(ID_{G1}||ID_{G1-j}||r_{G1-j}), \quad (2)$$

  and generates its authentication message $M_{MTCD_{G1-j}} = (ID_{G1}||ID_{G1-j}||r_{G1-j})$. Then, all MTC devices send their $MAC_{MTCD_{G1-j}}$ and $M_{MTCD_{G1-j}}$ to the $MTCD_{leader}$.
- ***Step-2:*** When the $MTCD_{leader}$ receives all $MACs$ from group members, it calculates

$$MAC_{G1} = MAC_{MTCD_{G1-1}} \oplus MAC_{MTCD_{G1-2}} \oplus \cdots \oplus MAC_{MTCD_{G1-n}} \oplus f^1_{GK_1}(LAI), \quad (3)$$

  where $\oplus$ represents XOR.
- ***Step-3:*** $MTCD_{leader} \rightarrow$ MME: ($AUTH_{G1}$).
  The $MTCD_{leader}$ generates $AUTH_{G1} = (M_{MTCD_{G1-1}}||...||M_{MTCD_{G1-n}}||MAC_{G1})$, and sends it to the MME.
- ***Step-4:*** MME $\rightarrow$ HSS: **Group authentication data request** ($AUTH_{G1}||LAI'$).
  Because the MME knows the LAI' of the base station (BS) forwarding $AUTH_{G1}$, it forwards $AUTH_{G1}$ to the HSS together with the BS's LAI'. When the HSS receives group authentication data request message contained $AUTH_{G1}$, it first computes $f^1_{GK_1}(LAI')$ using GK_1 and then verifies the received MAC_{G1} using K_{G1-j}. By checking MAC_{G1}, the HSS can verify whether the LAI' reported by the MME is the same as that recognized by the MTC_{leader}.
- ***Step-5:*** HSS $\rightarrow$ MME: **Group authentication data response (GAV)**.

[1] $MTCD_{leader}$ can be selected based on the communication capability, storage status and battery status of each MTCD or can be assigned in advance according to the requirement of applications.

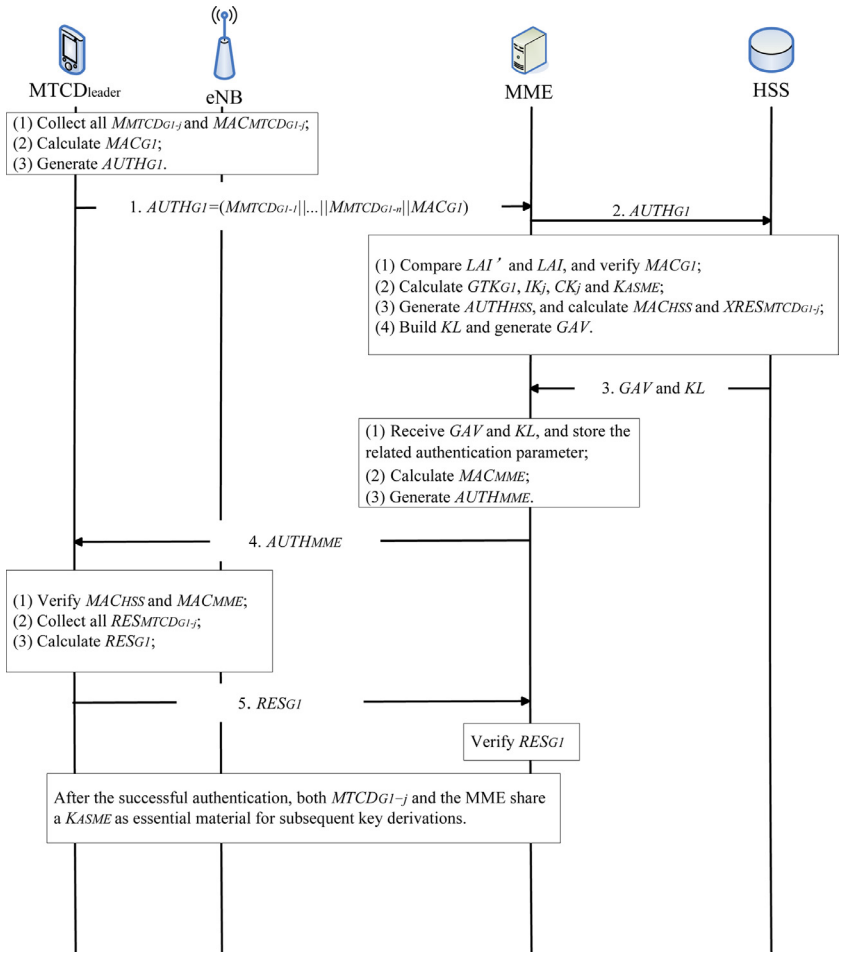


Fig. 2. The GLARM-1 protocol.

Once verification succeeds, the HSS needs to proceed as follows:

1. The HSS utilizes the corresponding group key GK_{G1} to generate a group temporary key as
$$GTK_{G1} = f^3_{GK_1}(r_{HSS}). \quad (4)$$
2. The HSS calculates each MTC device's IK_j and CK_j as follows
$$IK_j = f^4_{K_{G_{1-j}}}(r_{HSS}). \quad (5)$$
$$CK_j = f^5_{K_{G_{1-j}}}(r_{HSS}). \quad (6)$$
Then, it calculates a new key called Key for Access Security Management Entity (K_{ASME}) for each MTC device and the MME by using a Key Derivation Function (KDF) as specified in 3GPP TS 33.401 [5], which contains following input parameters: CK, IK, each MTC device's identity and MME's identity.
$$K^{MTCD_{G1-j}}_{ASME} = KDF(CK_j||IK_j||ID_{MME}||ID_{G1-j}). \quad (7)$$
Next, the HSS builds a Key List (KL) for all MTC devices in G1, shown as in Table 2.

Table 2
Key list in G1 (3GPP case).

Group	Group ID	MTCD ID	K_{ASME}
G1	ID_{G1}	$ID_{MTCD_{1-1}}$	$K^{MTCD_{G1-1}}_{ASME}$
		$ID_{MTCD_{1-2}}$	$K^{MTCD_{G1-2}}_{ASME}$
		⋮	⋮
		$ID_{MTCD_{1-n}}$	$K^{MTCD_{G1-n}}_{ASME}$

3. The HSS generates $AUTH_{HSS} = (r_{HSS}||AMF||MAC_{HSS})$, where
$$MAC_{HSS} = f^1_{GK_1}(ID_{HSS}||r_{HSS}||AMF), \quad (8)$$
and calculates
$$XRES_{G1} = XRES_{MTCD_{G1-1}} \oplus \cdots \oplus XRES_{MTCD_{G1-n}}, \quad (9)$$
where
$$XRES_{MTCD_{G1-j}} = f^2_{K_{G1-j}}(ID_{G1}||ID_{G1-j}||r_{HSS}). \quad (10)$$

4. HSS generates a group authentication vector (GAV) as $GAV = (r_{HSS}||XRES_{G1}||GTK_{G1}||AUTH_{HSS})$, and sends GAV and KL to the MME.

- ***Step-6:*** MME → $MTCD_{leader}$: **Group authentication request ($AUTH_{MME}$)**.
  After acquiring $AUTH_{HSS}$ for G1, the MME performs mutual authentication with $MTCD_{G1-j}$ by generating $AUTH_{MME}$ as follows: $AUTH_{MME} = (ID_{MME}||MAC_{MME}||MAC_{HSS}||r_{HSS}||r_{MME}||AMF)$, where

$$MAC_{MME} = f^1_{GTK_{G1}}(ID_{MME}||MAC_{HSS}||r_{MME}||r_{HSS}) \quad (11)$$

- ***Step-7:*** $MTCD_{leader}$ → MME: **Group authentication response (RES_{G1})**.
  On receiving the message from the MME, $MTCD_{leader}$ sends $AUTH_{MME}$ to all MTC devices in G1, each MTC devices verifies the received MAC_{MME} in $AUTH_{MME}$ as follows:
  1. $MTCD_{G1-j}$ computes

$$GTK_{G1} = f^3_{GK_{G1}}(r_{HSS}); \quad (12)$$

  2. $MTCD_{G1-j}$ computes

$$MAC'_{HSS} = f^1_{GK_1}(ID_{HSS}||r_{HSS}||AMF), \quad (13)$$

  and verifies whether MAC'_{HSS} equals MAC_{HSS} or not;
  3. $MTCD_{G1-j}$ computes

$$MAC'_{MME} = f^1_{GTK_{G1}}(ID_{MME}||MAC_{HSS}||r_{MME}||r_{HSS}). \quad (14)$$

  4. $MTCD_{G1-j}$ verifies whether MAC'_{MME} equals MAC_{MME} or not. If MAC'_{MME} is not same as MAC_{MME}, the HSS or the MME is not valid. Therefore, the $MTCD_{G1-j}$ terminates the procedure and sends MAC failure (Mac_Fail) message.

  If verification passes, $MTCD_{G1-j}$ computes its own $K^{MTCD_{G1-j}}_{ASME}$ similar to ***Step-5-2***, and calculates

$$RES_{MTCD_{G1-j}} = f^2_{K_{G1-j}}(ID_{G1}||ID_{G1-j}||r_{HSS}), \quad (15)$$

  and sends it to $MTCD_{leader}$. Then, $MTCD_{leader}$ calculates

$$RES_{G1} = RES_{MTCD_{G1-1}} \oplus \cdots \oplus RES_{MTCD_{G1-n}} \quad (16)$$

  and sends it to the MME.
- ***Step-8:*** MME → $MTCD_{G1-j}$: **Authentication acknowledge**.
  When the MME receives the group authentication response message carrying RES_{G1}, it compares $XRES_{G1}$ with RES_{G1}. If RES_{G1} equals $XRES_{G1}$, the verification is successful, it enables the KL for subsequent secure communication and sends authentication acknowledge to all MTC devices in G1, and the full authentication and key agreement procedure is completed.

After the successful authentication, both $MTCD_{G1-j}$ and the MME share a $K^{MTCD_{G1-j}}_{ASME}$ as essential material for subsequent key derivations. $K^{MTCD_{G1-j}}_{ASME}$'s function is the same as K_{ASME}'s [5].

4.2.2. Non-3GPP access case (GLARM-2)

A secure communication channel among the local authentication server (LAS), PAAA, HAAA and HSS has already been established and can provide security services to the transmitted data. When non-3GPP MTC devices in G1 detect the AP, similarly, a group leader of MTCDs in the group ($MTCD_{leader}$) will be selected. The proposed GLARM-2 is shown in Fig. 3, and works as follows.

- ***Step-1:*** To respond to the identity request, since the network that MTC devices access is untrusted non-3GPP type, each MTC device should calculate its temporary ID TID as follows

$$TID^1_{G1-j} = E_{K_{G_{1-j}}}(ID_{G1-j}). \quad (17)$$

$$TID^2_{G1-j} = E_{K_{G_{1-j}}}(ID_{G1-j}||r_{G1-j}). \quad (18)$$

- ***Step-2:*** Each MTC device calculates its MAC as follows

$$MAC_{MTCD_{G1-j}} = f^1_{K_{G1-j}}(ID_{G1}||ID_{G1-j}||r_{G1-j}), \quad (19)$$

  and generates its authentication message $M_{MTCD_{G1-j}} = (ID_{G1}||TID_{G1-j})$. Then, all MTC devices send their $MAC_{MTCD_{G1-j}}$ and $M_{MTCD_{G1-j}}$ to the $MTCD_{leader}$.
- ***Step-3:*** When the $MTCD_{leader}$ receives all MACs, it calculates

$$MAC_{G1} = MAC_{MTCD_{G1-1}} \oplus MAC_{MTCD_{G1-2}} \oplus \cdots \oplus MAC_{MTCD_{G1-n}}, \quad (20)$$

  where $\oplus$ represents XOR.
- ***Step-4:*** $MTCD_{leader}$ → AP: ($AUTH_{G1}$).
  The $MTCD_{leader}$ generates $AUTH_{G1} = (M_{MTCD_{G1-1}}||...||M_{MTCD_{G1-n}}||MAC_{G1})$, and sends it to the AP.
- ***Step-5:*** AP → HSS: **Group authentication data request** ($AUTH_{G1}$).
  AP forwards $AUTH_{G1}$ to the HSS through the LAS and PAAA/HAAA. When the HSS receives group authentication data request message contained $AUTH_{G1}$,
  1. Searches its database to find ID_{G1-j} and then extracts the correct encryption key: If $\kappa = TID^1_{G1-j}$ and ID_{G1-j} is a prefix of $D_{K_{G_{1-j}}}(TID^2_{G1-j})$, the HSS retrieves the suffix of $D_{K_{G_{1-j}}}(TID^2_{G1-j})$ as r_{G1-j}.
  2. Computes

$$MAC'_{MTCD_{G1-j}} = f^1_{K_{G1-j}}(ID_{G1}||D_{K_{G_{1-j}}}(TID^2_{G1-j})) \quad (21)$$

  and generates MAC'_{G1}. By comparing MAC_{G1} and MAC'_{G1}, the HSS can accept the validity of G1.
- ***Step-6:*** HSS → LAS: **Group authentication data response (GAV)**.
  Once verification succeeds, the HSS needs to proceed as follows:
  1. The HSS utilizes the corresponding group key GK_{G1} to generate a group temporary key as

$$GTK_{G1} = f^3_{GK_1}(r_{HSS}). \quad (22)$$

  2. The HSS calculates each MTC device's IK_j and CK_j as follows

$$IK_j = f^4_{K_{G_{1-j}}}(r_{HSS}||r_{G1-j}). \quad (23)$$

$$CK_j = f^5_{K_{G_{1-j}}}(r_{HSS}||r_{G1-j}). \quad (24)$$

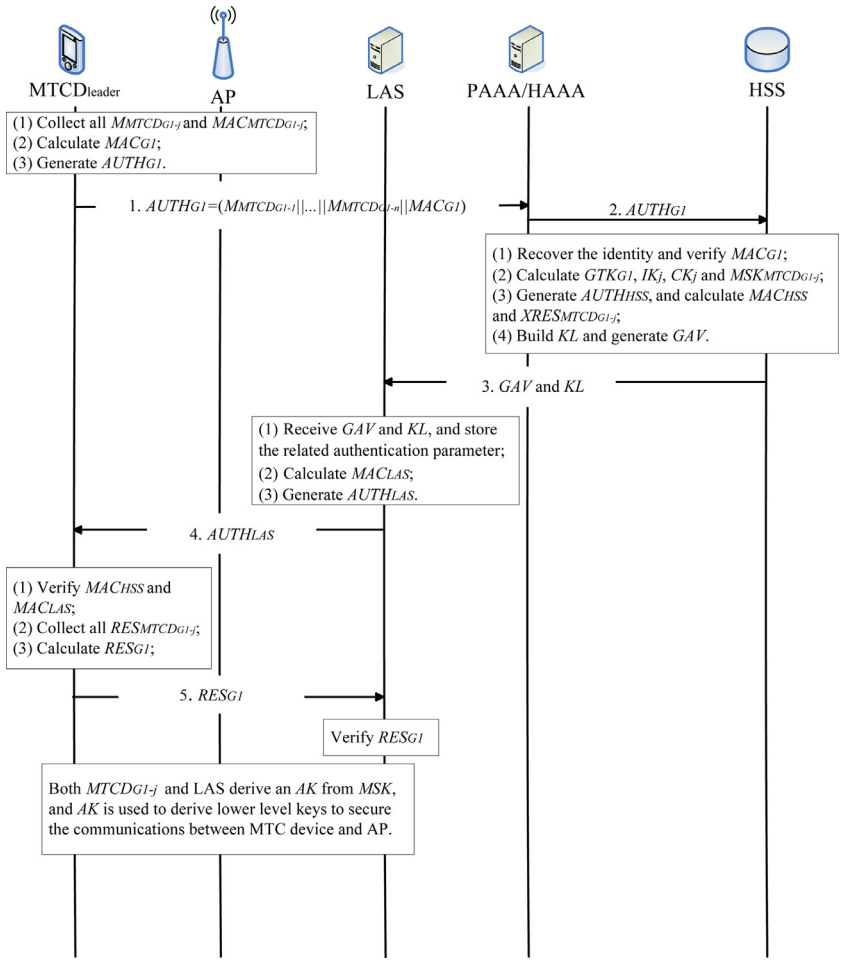


Fig. 3. The GLARM-2 protocol.

Table 3
Key list in G1 (non-3GPP case).

Group	Group ID	MTCD ID	*MSK*
G1	ID_{G1}	$TID_{MTCD_{1-1}}$	$MSK_{MTCD_{G1-1}}$
		$TID_{MTCD_{1-2}}$	$MSK_{MTCD_{G1-2}}$
		⋮	⋮
		$TID_{MTCD_{1-n}}$	$MSK_{MTCD_{G1-n}}$

Then, it calculates a Master Session Key (*MSK*) for each MTC device and the local authentication server (LAS) by using a hash function (SHA-1 or SHA-256), which contains following input parameters: CK, IK, each MTC device's identity and the local authentication server's identity.

$$MSK_{MTCD_{G1-j}} = hash(CK_j||IK_j||ID_{LAS}||ID_{G1-j}). \tag{25}$$

Then, the HSS builds a Key List (KL) for all MTC devices in G1, shown as in Table 3.

3. The HSS generates $AUTH_{HSS} = (r_{HSS}||ID_{HSS}||MAC_{HSS})$, where

$$MAC_{HSS} = f^1_{GK_1}(ID_{HSS}||r_{HSS}), \tag{26}$$

and calculates

$$XRES_{G1} = XRES_{MTCD_{G1-1}} \oplus \cdots \oplus XRES_{MTCD_{G1-n}}, \tag{27}$$

where

$$XRES_{MTCD_{G1-j}} = f^2_{K_{G1-j}}(ID_{G1}||ID_{G1-j}||r_{HSS}||r_{G1-j}). \tag{28}$$

4. The HSS generates a group authentication vector (GAV) as $GAV = (r_{HSS}||XRES_{G1}||GTK_{G1}||AUTH_{HSS})$, and sends *GAV* and KL to the LAS.

- ***Step-7:*** LAS → $MTCD_{leader}$: **Group authentication request ($AUTH_{LAS}$).** After acquiring $AUTH_{HSS}$ for G1, the LAS performs mutual authentication with $MTCD_{G1-j}$ by generating $AUTH_{LAS}$ as follows: $AUTH_{LAS} = (ID_{LAS}||MAC_{LAS}||MAC_{HSS}||r_{HSS}||r_{LAS})$, where

$$MAC_{LAS} = f^1_{GTK_{G1}}(ID_{LAS}||MAC_{HSS}||r_{LAS}||r_{HSS}). \tag{29}$$

- ***Step-8:*** $MTCD_{leader} \rightarrow$ LAS: **Group authentication response (RES_{G1})**.
  On receiving the message from the LAS, $MTCD_{leader}$ sends $AUTH_{LAS}$ to all MTC devices in G1, each MTC devices verifies the received MAC_{LAS} in $AUTH_{LAS}$ as follows:
  1. $MTCD_{G1-j}$ computes
  $$GTK_{G1} = f^3_{GK_{G1}}(r_{HSS}); \tag{30}$$
  2. $MTCD_{G1-j}$ computes
  $$MAC'_{HSS} = f^1_{GK_1}(ID_{HSS}||r_{HSS}), \tag{31}$$
  and verifies whether MAC'_{HSS} equals MAC_{HSS} or not;
  3. $MTCD_{G1-j}$ computes
  $$MAC'_{LAS} = f^1_{GTK_{G1}}(ID_{LAS}||MAC_{HSS}||r_{LAS}||r_{HSS}); \tag{32}$$
  4. The $MTCD_{G1-j}$ verifies whether MAC'_{LAS} equals MAC_{LAS} or not. If MAC'_{LAS} is not same as MAC_{LAS}, the HSS or the LAS is not valid. Therefore, the $MTCD_{G1-j}$ terminates the procedure and sends MAC failure (Mac_Fail) message.

  If verification passes, similarly, $MTCD_{G1-j}$ computes its own $MSK_{MTCD_{G1-j}}$ and calculates
  $$RES_{MTCD_{G1-j}} = f^2_{K_{G1-j}}(ID_{G1}||ID_{G1-j}||r_{HSS}||r_{G1-j}), \tag{33}$$
  and sends it to $MTCD_{leader}$. Then, $MTCD_{leader}$ calculates
  $$RES_{G1} = RES_{MTCD_{G1-1}} \oplus \cdots \oplus RES_{MTCD_{G1-n}} \tag{34}$$
  and sends it to the LAS.
- ***Step-9:*** LAS $\rightarrow$ $MTCD_{G1-j}$: **Authentication acknowledge**.
  When the LAS receives the group authentication response message carrying RES_{G1}, it compares $XRES_{G1}$ with RES_{G1}. If RES_{G1} equals $XRES_{G1}$, the verification is successful, it enables the KL for subsequent secure communication and sends authentication acknowledge to all MTC devices in G1, and the full authentication and key agreement procedure is completed.

After the successful authentication, both $MTCD_{G1-j}$ and the LAS share an $MSK_{MTCD_{G1-j}}$ as essential material for subsequent key derivations. In addition, the LAS and MTC device derive an Authorization Key (AK) from MSK, and AK is used to derive lower level keys to secure the communications between MTC device and AP [26].

4.3. Group member joining/leaving the group

In our scheme, the group key (GK) can be used to generate MAC_{HSS} and MAC_{MME}/MAC_{LAS}, and then MTC devices can use GK to authenticate HSS and MME/LAS. Therefore, when group members join or leave the group, the GK need to be updated immediately since it will influence the security of the system. Moreover, if the GK is used to encrypt group messages, the group which formed by MTC devices requires backward and forward secrecy. Backward secrecy is required that a new MTC device cannot get messages exchanged before it joined the group. Forward secrecy is required that a leaving or expelled MTC device cannot continue accessing the group's communication (if it keeps receiving the messages).

When an MTC device wants to leave the group, the HSS will revoke the binding relationship between the MTC device and the group that it belongs to, thus the MTC device cannot longer communicate with the core network as the group member. Moreover, in order to prevent the old MTC device to decrypt the new packets of the group which it was able to sniff, the group key must be updated when the old MTC device leaves the group. After the old MTC device leaves the group, all members of the group should share a new group key. Similarly, when an MTC device wants to join the group, an access control of the group is necessary for it, and it needs to perform a full AKA authentication procedure with the HSS. Meanwhile, the group key must be updated when the new MTC device wants to join a group. After the new MTC device joins the group, all members of the group should share a new group key. In that case, the new MTC device cannot decrypt the old packets of the group before it joins in. The group key upgrade of group communication has been widely studied, and it is out of scope for this paper and specific technology can be found in Refs. [35,36].

5. Security analysis

In this section, we analyze the security properties of the proposed GLARM scheme. In specific, our analysis will focus on how the proposed GLARM achieves all the security goals above.

Mutual authentication and key agreement: For the mutual authentication, in the schemes GLARM-1 and GLARM-2, all the MTC devices in the G1 first calculate their $MAC_{MTCD_{G1-j}}$, and send them to the MTC_{leader}. Then, the MTC_{leader} collects all $MAC_{MTCD_{G1-j}}$ and calculates $MAC_{G1} = MAC_{MTCD_{G1-1}} \oplus \cdots \oplus MAC_{MTCD_{G1-n}} \oplus f^1_{GK_1}(LAI)$. By verifying MAC_{G1}, the HSS can identify all MTC devices and the group. Then, the HSS calculates MAC_{HSS} and $XRES_{G_1}$, and generates GAV for all MTC devices in G1. The HSS sends GAV containing MAC_{HSS} and $XRES_{G_1}$ to the MME/LAS; then, the MME/LAS calculates MAC_{MME}/MAC_{LAS} and generates $AUTH_{MME}/AUTH_{LAS}$, and sends them to all MTC devices in G1. By verifying $AUTH_{MME}/AUTH_{LAS}$, all MTC devices can trust the HSS and the MME/LAS. Finally, all MTC devices in the G1 calculate their $RES_{MTCD_{G1-j}}$, and send them to the MTC_{leader}. The MTC_{leader} collects all $RES_{MTCD_{G1-j}}$ and calculates $RES_{G1} = RES_{MTCD_{G1-1}} \oplus \cdots \oplus RES_{MTCD_{G1-n}}$. By verifying RES_{G1}, the MME/LAS can authenticate all MTC devices in G1. In conclusion, the GLARM-1 and GLARM-2 can achieve the mutual authentication.

It is worth noting that, in our proposed scheme, the batch authentication is mainly based on the aggregate message authentication code, whose security has been proofed in [31]. In addition, the security of the scheme is highly related to the shared keys, including K_{Gi-j} and GK_i, which is a kind of symmetric cryptosystem. Therefore, the K_{Gi-j} and GK_i must be securely stored. Once the exposure of K_{Gi-j} and GK_i, the authentication fails. For the key agreement procedure, because all keys used among entities are computed on either peer side

directly, without being transmitted over any insecure communication channels, the key agreement procedure is secure.

Authentication failure issue: Due to the introduction of group authentication, GLARM has several advantages such as lower authentication delay and transmission overhead. However, the expense of the group authentication is that, once an invalid access authentication request exists in a batch of requests, the group authentication may lose its efficacy. Note that the invalid access authentication request could be inserted by the malicious attackers. This problem commonly accompanies other batch-based verification schemes [30,37]. To deal with this problem, we carefully analyze what happens if the issue occurs.

We assume that at most 1% MTC devices in a group can be compromised and send an invalid access authentication message aggregated by the group leader, and then be forward to the HSS. While the number of MTC devices (denoted as N_{GM}) is assumed to be 1000, the largest number of compromised MTC devices (which is denoted as N_C) is $N_{GM} \times 1\% = 1000 \times 1\% = 10$. In a period, the number of requests that the HSS can process in a group authentication is defined as N_{GA}. Then, when one or more malicious requests are within N_{GA}, the re-authentications are needed to be performed.

Let $Pr\{i\}$ represent the probability that exactly i invalid access authentication requests sent from N_C are being sent to the HSS. The probability follows the hypergeometric distribution $\mathcal{H}(i,\ N_{GM}, N_C, N_{GA})$ as follows:

$$Pr\{X = i\} = \frac{\binom{N_{GA} - N_C}{N_{GM} - i}\binom{N_C}{i}}{\binom{N_{GA}}{N_{GM}}}, \quad i = 0, 1, \ldots, 10 \tag{35}$$

That is, there are N_{GA} requests can be authenticated, i invalid requests sent from N_C, and $N_{GM} - i$ valid requests sent from $N_{GA} - N_C$. Let $\mathcal{R}$ be the event that re-authentication is required to successfully verify all valid access authentication requests N_{GA}. Then, $Pr\{R\}$ can be represented as

$$\begin{aligned} Pr\{R\} &= Pr\{i = 1\} + \cdots + Pr\{i = 10\} \\ &= \frac{\binom{N_{GA} - N_C}{N_{GM} - 1}\binom{N_C}{1}}{\binom{N_{GA}}{N_{GM}}} + \cdots + \frac{\binom{N_{GA} - N_C}{N_{GM} - 10}\binom{N_C}{10}}{\binom{N_{GA}}{N_{GM}}} \\ &= \frac{\sum_{i=1}^{10}\binom{N_{GA} - N_C}{N_{GM} - i}\binom{N_C}{i}}{\binom{N_{GA}}{N_{GM}}} \end{aligned} \tag{36}$$

Once there is at least an invalid access authentication request, it leads to the failure of group authentication and thus re-authentications are required. Then, we demonstrate the relationship between the number of compromised MTC devices and that of access authentication requests from Eq. (36). We can conclude that the probability of rebatch verification is almost negligible while only one or two invalid requests ($i = 1$ or 2) in a batch. Indeed, invalid access authentication requests is a kind of DoS attacks since they lead to the failure of group authentication. Fortunately, we can find efficient solutions from previous works [30] to detect attackers, and weaken this attack.

Privacy preservation (anonymity): The privacy preservation (i.e., anonymity) issue is mainly for the non-3GPP case since the non-3GPP access network is untrusted. Therefore, anonymity must be guaranteed in the non-3GPP case. In the GLARM-2, Step-1, to respond to the identity request, each MTC device should calculate its temporary ID TID as $TID^1_{G1-j}=E_{K_{G_{1-j}}}(ID_{G1-j})$ and $TID^2_{G1-j}=E_{K_{G_{1-j}}}(ID_{G1-j}||r_{G1-j})$. When the HSS receives group authentication data request message contained $AUTH_{G1}$, it searches its database to find ID_{G1-j} and then extracts the correct encryption key: If $\kappa=TID^1_{G1-j}$ and ID_{G1-j} is a prefix of $D_{K_{G_{1-j}}}(TID^2_{G1-j})$, the HSS retrieves the suffix of $D_{K_{G_{1-j}}}(TID^2_{G1-j})$ as r_{G1-j}. Therefore, the MTC device's real identity is not transferred in plain text over the non-3GPP access network. In the GLARM-1, there is no such a privacy preservation mechanism since the eNB is controlled by the mobile operator and thus is trusted. Actually, GLARM-1 can also equip with this mechanism. In this sense, GLARM-2 could be considered as the enhanced privacy preservation version of GLARM-1.

Resistance to redirection attack: Redirection attack is mainly for the 3GPP case, and an adversary initiates a redirection attack by simulating a BS to obtain user information and by impersonating an MTC device to forward user messages to its destination. The redirection attack fails if the adversary fails to obtain user information by impersonating a BS. Without the user information, the adversary cannot impersonate any MTC device and connect to a legitimate BS. To impersonate a BS, the adversary either transmits signals with stronger power or jams the spectrum and tries to entrap the MTC device to establish the connection with the faked BSs. In the GLARM-1, the $MTCD_{leader}$ embeds the LAI of the BS in MAC_{G1} and sends MAC_{G1} to the MME. The authentication request is rejected if the HSS fails to match the LAI reported by the MME and the LAI embedded in MAC_{G1}.

Resistance to DoS attack: During the authentication, a malicious MTC device may launch a DoS attack either to the HSS or to the MME/LAS. If the malicious MTC device forges message (i.e., invalid MAC), the forged message can be detected by the HSS through checking MAC_{Gi}. However, if the malicious MTC device sends the invalid MACs all the time, which leads to failed verification. Obviously, this is a kind of DoS attack. Fortunately, we can find efficient solutions from previous works, such as [30]. In addition, we have demonstrated that for the GLARM, the re-authentications cost is fairly small compared with other scheme. That is, even though the malicious MTC device keeps sending the invalid MACs, the HSS can detect the invalid MAC and quickly re-authenticate the legitimate MTC devices in the group. Therefore, the DoS attack can be weaken in our proposed scheme.

The proof of MITM attack is similar to SE-AKA, which can be found in [19]. Finally, from our analysis and comparison, we derive the properties of the

Table 4
Comparisons among the authentication and key agreement schemes.

	EPS-AKA [5]	EAP-AKA [6]	AP-AKA [9]	S-AKA [10]
TOC	Symmetric	Symmetric	Symmetric	Symmetric
BSR	Yes	Yes	Yes	Yes
FTS	Yes	Yes	No	No
PPR	No	No	No	No
RRA	No	No	Yes	Yes
RMA	No	No	Yes	Yes
COA	No	No	No	No
MCO	No	No	No	No
MCC	No	No	No	No
SGA	No	No	No	No
GAS	No	No	No	No
	G-AKA [17]	SE-AKA [19]	Scheme [13,14]	GLARM-1 and GLARM-2
TOC	Symmetric	Hybrid	Asymmetric	Symmetric
BSR	Yes	Yes	Yes	Yes
FTS	Yes	No	No	Yes
PPR	No	Yes	No	Yes
RRA	No	Yes	No	Yes
RMA	No	Yes	Yes	Yes
COA	Partially	Partially	Yes	Yes
MCO	Partially	Partially	Yes	Yes
MCC	Partially	Partially	No	Yes
SGA	Yes	Yes	Yes	Yes
SCA	No	No	Yes	Yes

TOC: type of cryptosystem; BSR: basic security requirements; FTS: follow the standard; PPR: privacy preservation; RRA: resistance to redirection attack; RMA: resistance to MITM attack; COA: Congestion avoidance; MCO: minimize communication overhead; MCC: minimize computation complexity; SGA: support group authentication; SCA: authentication signaling congestion avoidance.

relative AKA protocols as follows, and show the results in Table 4. According to Table 4, we confirm that the proposed GLARM scheme is superior to other relevant protocols for resource-constrained M2M devices in 3GPP networks.

6. Performance evaluation

In this section, we evaluate the performance of the proposed GLARM scheme in terms of computation complexity and communication overhead.

6.1. Computation complexity

Due to the use of symmetric cryptography, including [5,6,9,10,17], GLARM-1 and GLARM-2, their computation overheads are fairly small. We assume the hash operation T_{hash} takes 0.02 milliseconds (ms). Thus, we mainly consider the cost of the following operations, including a point multiplication T_{mul}, a pairing operation T_{pair} and a map to point hash operation T_{mtp}. According to [13], T_{mtp} takes the same time as T_{mul} (= 0.6 ms) and T_{pair} = 4.5 ms. The cost of XOR can be negligible. Moreover, n represents the number of MTCDs in a group, m represents the number of groups.

We present the computation complexity comparison of the proposed GLARM scheme and other relative AKA protocols. The comparison of computation complexity is shown in Table 5 and in Fig. 4. From the figure, we can clearly see that the computation complexity of our proposed GLARM protocol is much less than schemes [13,14,19], and is close to other existing protocols. However, among these existing protocols, only scheme [13,14] have the similar functionality as ours.

Next, we give an analysis of the verification cost for re-batch verifications in the GLARM.

We elaborately evaluate the verification cost of our scheme if there is at least one invalid access request. By our scheme, the network has to compute all keys and authentication parameters in the first verification (T_{1th}), which takes $(4T_{hash})n + 3T_{hash}$, while a re-verification needs T_{hash}. For Similar schemes [13,14], the first verification takes $T_{1th} = nT_{tmp} + nT_{mul} + 2T_{pair}$, while a re-verification needs $2T_{pair}$. To be precise, there are the two following cases for batch verification by using the detection process [30], which are worst case and average case described as follows.

In the worst case, a valid MAC is always with the invalid MAC in the same 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 MAC in this case T_{worst} is:

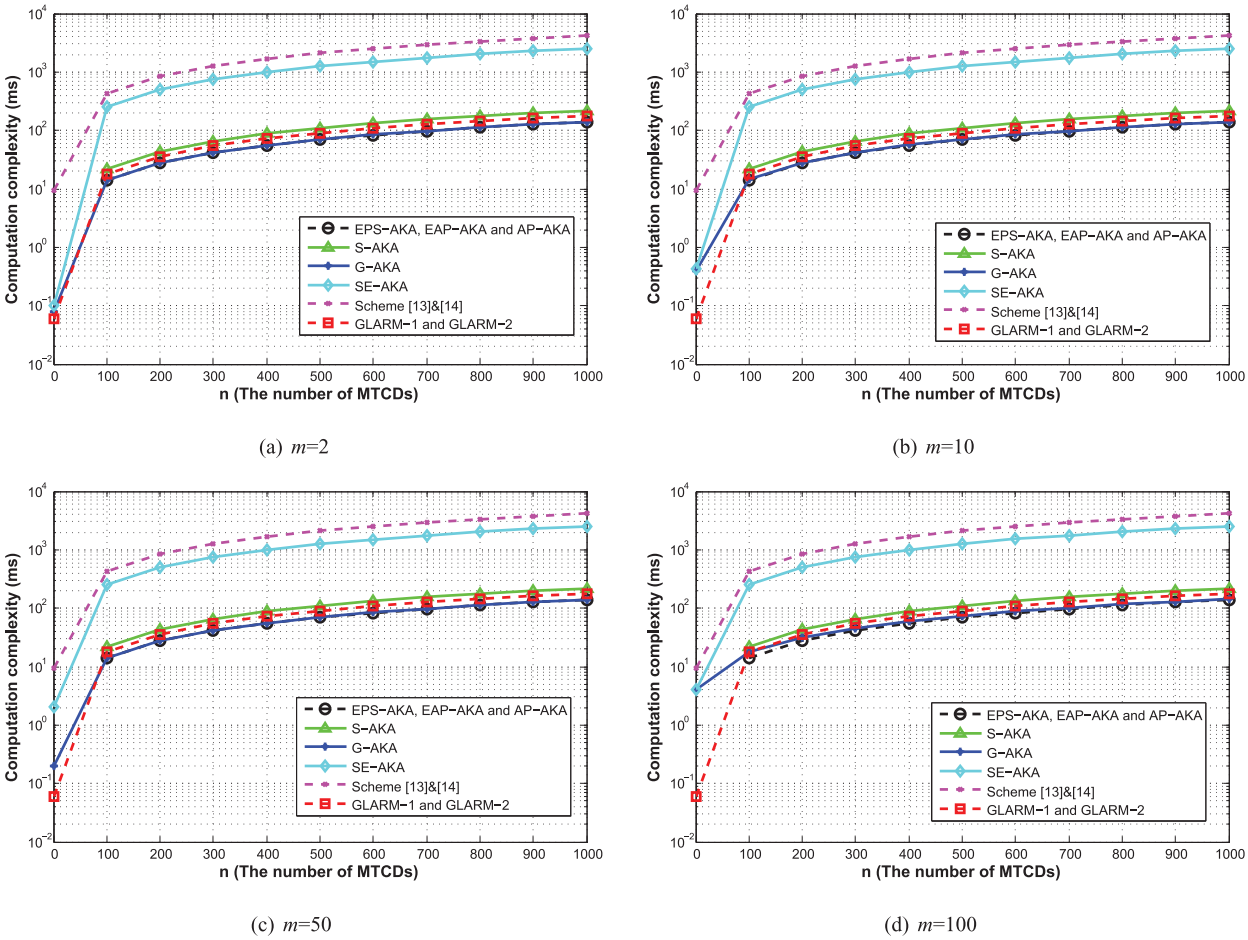
$$T_{worst} = T_{1th} + 2\left\lceil \log_2^n \right\rceil T_{re}. \tag{37}$$

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 in this case T_{ave} is:

$$T_{ave} = T_{1th} + \frac{1}{\left\lceil \log_2^n \right\rceil + 1} \sum_{i=2}^{\left\lceil \log_2^n \right\rceil} T_{re}. \tag{38}$$

Table 5
Comparison of computation complexity.

ms	EPS and EAP-AKA [5,6]	AP-AKA [9]	S-AKA [10]	G-AKA [17]
Each MTCD	$2T_{hash} = 0.04$	$2T_{hash} = 0.04$	$5T_{hash} = 0.1$	$4T_{hash} = 0.08$
The network	$(5T_{hash})n = 0.1n$	$(4T_{hash})n = 0.08n$	$(6T_{hash})n = 0.12$	$(3T_{hash})n + (2T_{hash})m = 0.06n + 0.04m$
Total	$0.14n$	$0.12n$	$0.22n$	$0.14n + 0.04m$
ms	SE-AKA [19]	Scheme [13,14]	GLARM-1	GLARM-2
Each MTCD	The first one: $4T_{hash} + 2T_{mul} = 1.28$ The remaining: $3T_{hash} + 2T_{mul} = 1.26$	$4T_{mul} = 2.4$	$3T_{hash} = 0.06$	$5T_{hash} = 0.1$
The network	$(3T_{hash} + 2T_{mul})n + (2T_{hash})m = 1.26n + 0.04m$	$nT_{mtp} + (2n+1)T_{mul} + 2T_{pair} = 9.6 + 1.8n$	$(4T_{hash})n + 3T_{hash} = 0.08n + 0.06$	$(4T_{hash})n + 3T_{hash} = 0.08n + 0.06$
Total	$2.52n + 0.04m + 0.02$	$9.6 + 4.2n$	$0.14n + 0.06$	$0.18n + 0.06$



(a) m=2 (b) m=10

(c) m=50 (d) m=100

Fig. 4. Comparison of computation complexity.

Fig. 5 shows the results of the total verification cost in the worst case and in the average case when there are 1000 MTC devices. According to the Fig. 5, we can conclude that the total verification cost of the GLARM is far less than that of [13,14]. Specifically, we can see that for the GLARM, the total verification cost in the worst case and in the average case are very close since there is minimal re-authentications cost in our proposed scheme.

6.2. Communication overhead

In this section, we evaluate the communication overhead of our scheme in terms of the signaling overhead and bandwidth consumption.

- *Signaling overhead*

The proposed GLARM scheme enables the $MTCD_{leader}$ to aggregate a collection of each MTCD's MACs in the same

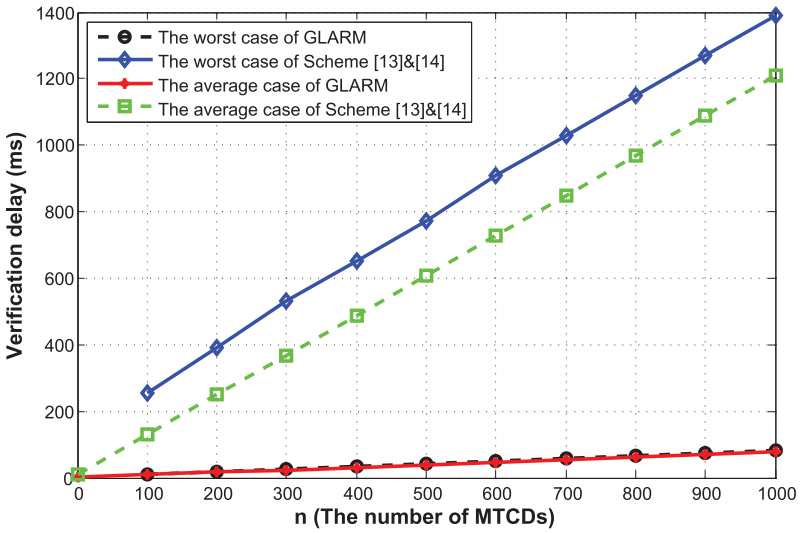


Fig. 5. Analysis results for verification cost of the network.

group. It largely reduces the signaling overhead of authentication. We assume that there are n MTCDs forming m groups, obviously, $n > m$. For the EPS-AKA and EAP-AKA, there are 2 messages between the MME/LAS and HSS, since the MME/LAS needs to derive authentication vectors (AV) from the HSS for an MTC device, and there are 6 messages between each MTC device and the MME/LAS. For the schemes [13,14], each MTCD requires 2 extra signaling exchanges with the KGC for generating long-term key in the register phase. For the proposed GLARM-1 and GLARM-2 schemes, because all MTC devices in a group can be authenticated at one time through calculating MAC_{G_i}, $XRES_{G_i}$ and RES_{G_i}, for all MTC devices, there are 2 messages between the MME/LAS and HSS, and there are 4 messages between MTC_{leader} and the MME/LAS.

Fig. 6 shows the comparison of the number of signaling messages with the change of the number of MTC devices. According to Fig. 6, we can see that the signaling overhead of the GLARM does not change with n, and only depends on m; therefore, the signaling overhead incurred in the GLARM scheme is much less than that of other existing schemes. The proposed GLARM can largely reduce the authentication signaling overhead and alleviate the burden of eNBs/APs and the MME/LAS/AAA server. Thus, the GLARM scheme can ensure QoS for MTC devices without restriction on the access requests.

- *Bandwidth consumption*

In order to analyze the bandwidth consumption, we assume that x AVs are transmitted every time the HSS successfully authenticates one MTC device, and there are n MTC devices forming m groups. Table 6 is the setting of parameters for evaluating bandwidth consumption.

The bandwidth consumption of AKA protocols are as follows, where bw_{first} represents the bandwidth consumption of the authentication of the first MTC device. The specific calculation process of (1)–(5) can be found in [19], and we give the concrete computation procedure of (6) and (7).

Table 6
Setting of parameters.

Parameters	Value (bits)
PID/TID	128
AMF	48
LAI	40
GTK	128
Hash value/MAC	64
Random number (RN)	128
ECDH key	192

(1) Bandwidth analysis of EPS-AKA and EAP-AKA: The sizes of authentication messages are calculated as follows.

$$bw_{each} = \sum_{i=1}^{5} |Message_i| = 704 + 608x \quad \text{bits.} \tag{39}$$

The overall bandwidth consumption for n devices is calculated as $(704 + 608x)n$ bits.

(2) Bandwidth analysis of AP-AKA: The sizes of authentication messages are calculated as follows.

$$bw_{each} = \sum_{i=1}^{6} |Message_i| = 1250 + 544x \quad \text{bits.} \tag{40}$$

The overall bandwidth consumption for n devices is calculated as $(1250 + 544x)n$ bits.

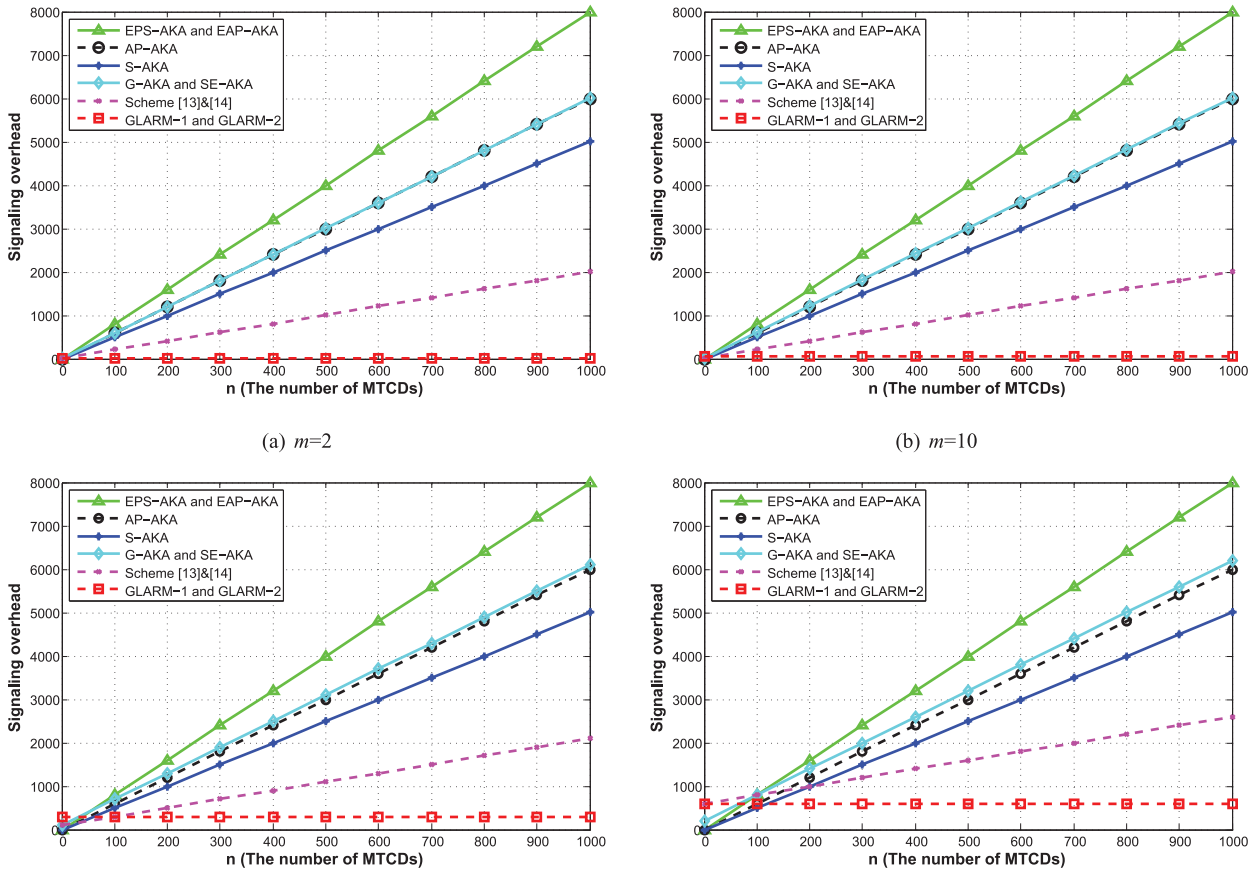
(3) Bandwidth analysis of S-AKA: The sizes of authentication messages are calculated as follows.

$$bw_{each} = \sum_{i=1}^{5} |Message_i| = 1312 \quad \text{bits.} \tag{41}$$

The overall bandwidth consumption for n devices is calculated as $1312n$ bits.

(4) Bandwidth analysis of G-AKA: The sizes of authentication messages are calculated as follows.

$$bw_{first} = \sum_{i=1}^{5} |Message_i| = 1888 \quad \text{bits.} \tag{42}$$



(a) m=2 (b) m=10

(c) m=50 (d) m=100

Fig. 6. Comparison of signaling overhead.

$$bw_{remaining} = \sum_{i=1}^{2} |Message_i| = 880 \quad \text{bits.} \tag{43}$$

The overall bandwidth consumption for n devices is calculated as $1888m + 880(n - m)$ bits.

(5) Bandwidth analysis of SE-AKA: The sizes of authentication messages are calculated as follows.

$$bw_{first} = \sum_{i=1}^{5} |Message_i| = 2184 \quad \text{bits.} \tag{44}$$

$$bw_{remaining} = \sum_{i=1}^{3} |Message_i| = 1328 \quad \text{bits.} \tag{45}$$

The overall bandwidth consumption for n devices is calculated as $2184m + 1328(n - m)$ bits.

(6) Bandwidth analysis of schemes [13,14]: The sizes of authentication messages are calculated as follows, we assume the length of signature is 320 bits.

$$bw_{overall} = \sum_{i=1}^{2} |Message_i| = 704\frac{n}{m} + 576 \quad \text{bits.} \tag{46}$$

- $Message_1 = |\sigma| + |m_i|\frac{n}{m} = 704\frac{n}{m} + 320$ bits.
- $Message_2 = |MAC| + |ECDH\ key| = 256$ bits

The overall bandwidth consumption for n devices is calculated as $(704\frac{n}{m} + 576)m$ bits.

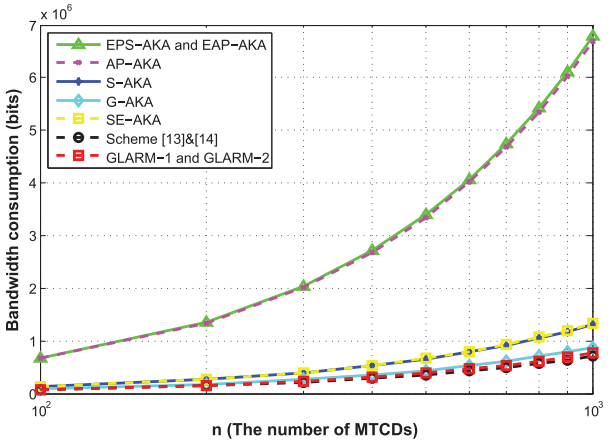
(7) Bandwidth analysis of GLARM-1 and GLARM-2: The sizes of authentication messages are calculated as follows.

$$bw_{overall} = \sum_{i=1}^{5} |Message_i| = 768\frac{n}{m} + 1440 \quad \text{bits.} \tag{47}$$

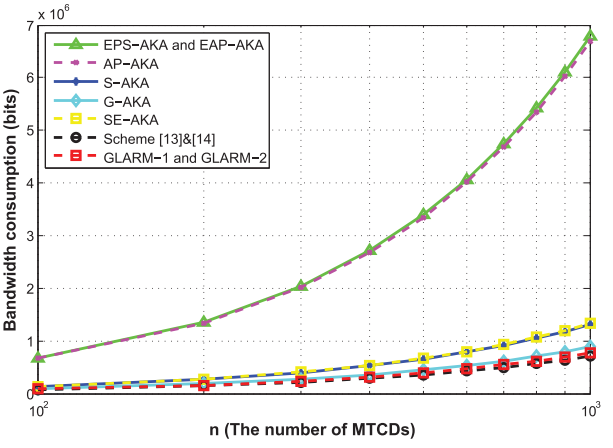
- $Message_1 = |M_{MTCD_{G1-j}}|\frac{n}{m} + |MAC| = 384\frac{n}{m} + 64$ bits
- $Message_2 = Message_1 = 384\frac{n}{m} + 64$ bits
- $Message_3 = 2|RN| + |XRES| + |GTK| + |AMF| + |MAC| = 560$ bits
- $Message_4 = 2|ID| + 2|MAC| + 2|RN| + |AMF| = 688$ bits
- $Message_5 = |RES| = 64$ bits

The overall bandwidth consumption for n devices is calculated as $(768\frac{n}{m} + 1440)m$ bits.

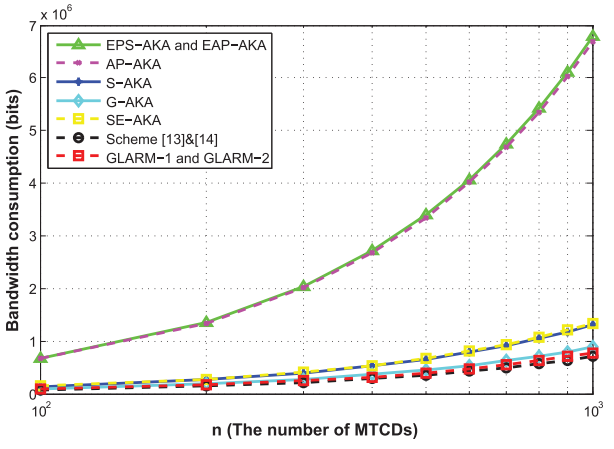
We present the bandwidth consumption comparison of the proposed GLARM scheme and other relative AKA protocols. The comparison of bandwidth consumption is shown in Fig. 7. From the figure, we can clearly see that the



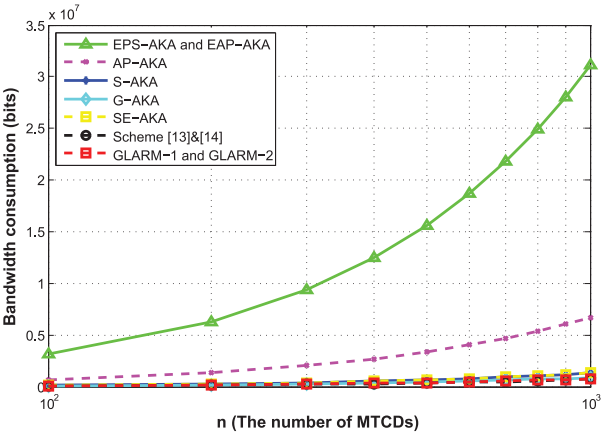
(a) x=10, m=2



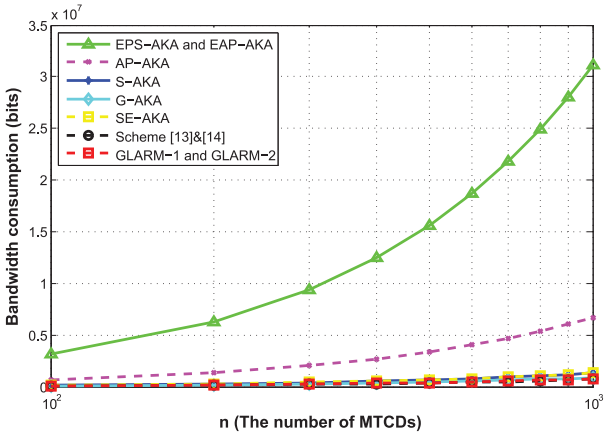
(b) x=10, m=5



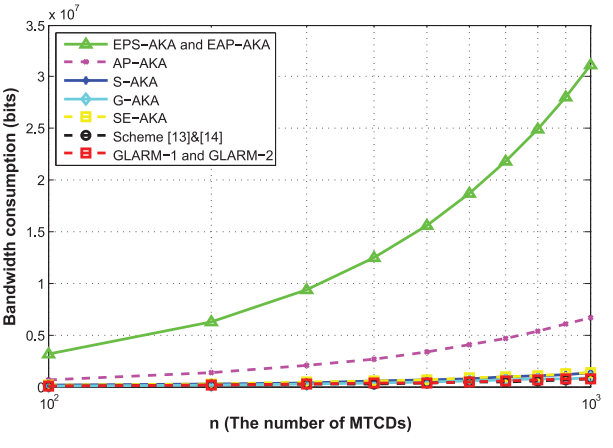
(c) x=10, m=10



(d) x=50, m=2



(e) x=50, m=5



(f) x=50, m=10

Fig. 7. Comparison of bandwidth consumption.

bandwidth consumption of our proposed GLARM protocol is much less than EPS-AKA, EAP-AKA and AP-AKA, and is close to other existing protocols.

7. Conclusions

In this paper, we have proposed a lightweight group authentication scheme, named GLARM, that supports the group authentication of a massive number of MTC devices in 3GPP networks. GLARM can not only authenticate all MTC devices simultaneously, but also minimize the authentication overhead. In the proposed GLARM, the MME/LAS can authenticate all MTC devices in the same group simultaneously based on aggregate MACs. The security analysis demonstrates that the GLARM can provide robust security protection. Particularly, our GLARM outperforms the similar schemes in the aspect of the re-authentication cost. The performance evaluation shows that GLARM is more efficient than the existing schemes in terms of computation complexity and communication overhead. Specifically, the signaling overhead of the GLARM is reduced at least by 60%, which can efficiently alleviate the authentication signaling congestion in the 3GPP networks.

Acknowledgments

This work is financially supported by the National Natural Science Foundation of China Research Grant (61502386, 61472472, 61402354, U1401251), the International Science and Technology Cooperation and Exchange Plan in Shaanxi Province of China (2015KW-010), and Scientific Research Program Funded by Shaanxi Provincial Education Department (15JK1669).

References

[1] S. Gilani, The promise of M2M: How pervasive connected machines are fueling the next wireless revolution, White Pap. Mentor Graph. (2009).
[2] 3GPP TR 23.888 V1.4.0, System Improvements for Machine-Type Communications (2011).
[3] S.-Y. Lien, K.-C. Chen, Y. Lin, Toward ubiquitous massive accesses in 3GPP machine-to-machine communications, IEEE Commun. Mag. 49 (4) (2011) 66–74.
[4] R. Lu, X. Li, X. Liang, X. Shen, X. Lin, GRS: The green, reliability, and security of emerging machine to machine communications, IEEE Commun. Mag. 49 (4) (2011) 28–35.
[5] 3GPP TS 33.401 V12.5.0, 3GPP System Architecture Evolution (SAE); Security architecture (2012).
[6] RFC 4187, Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA) (2006).
[7] U. Meyer, S. Wetzel, A man-in-the-middle attack on UMTS, in: Proceedings of the Third ACM Workshop on Wireless security, 2004, pp. 90–97.
[8] M. Zhang, Provably-secure enhancement on 3GPP authentication and key agreement protocol, Cryptology (2003). ePring Archive (http://eprint. iacr. ors/).
[9] M. Zhang, Y. Fang, Security analysis and enhancements of 3GPP authentication and key agreement protocol, IEEE Trans. Wirel. Commun. 4 (2) (2005) 734–742.
[10] Y.-L. Huang, C.-Y. Shen, S.W. Shieh, S-AKA: a provable and secure authentication key agreement protocol for UMTS networks, IEEE Trans. Veh. Technol. 60 (9) (2011) 4509–4519.
[11] A. Amokrane, A. Ksentini, Y. Hadjadj-Aoul, T. Taleb, Congestion control for machine type communications, in: Proceedings of the of IEEE International Conference on Communication ICC, 2012.
[12] R. Lu, X. Liang, X. Li, X. Lin, X. Shen, EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications, IEEE Trans. Parallel Distrib. Syst. 23 (9) (2012) 1621–1631.
[13] J. Cao, M. Ma, H. Li, A group-based authentication and key agreement for MTC in LTE networks, in: Proceedings of the of IEEE Global Communication Conference Globecom, 2012, pp. 1017–1022.
[14] C. Lai, H. Li, R. Lu, R. Jiang, X. Shen, SEGR: A secure and efficient group roaming scheme for machine to machine communications between 3GPP and WiMAX networks, in: Proceedings of the of IEEE International Conference on Communication, 2014, pp. 1011–1016.
[15] H.-H. Ou, M.-S. Hwang, J.-K. Jan, A cocktail protocol with the authentication and key agreement on the UMTS, J. Syst. Softw. 83 (2) (2010) 316–325.
[16] S. Wu, Y. Zhu, Q. Pu, Security analysis of a cocktail protocol with the authentication and key agreement on the UMTS, IEEE Commun. Lett. 14 (4) (2010) 366–368.
[17] Y.-W. Chen, J.-T. Wang, K.-H. Chi, C.-C. Tseng, Group-based authentication and key agreement, Wirel. Pers. Commun. 62 (4) (2012) 965–979.
[18] C. Lai, H. Li, X. Li, J. Cao, A novel group access authentication and key agreement protocol for machine-type communication, Trans. Emerg. Telecommun. Technol. 26 (3) (2015) 414–431.
[19] C. Lai, H. Li, R. Lu, X.S. Shen, SE-AKA: A secure and efficient group authentication and key agreement protocol for LTE networks, Comput. Netw. 57 (17) (2013) 3492–3510.
[20] J. Arkko, V. Lehtovirta, P. Eronen, Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA) (2009).
[21] 3GPP TS 33.402 V11.3.1, 3G System Architecture Evolution (SAE): Security architecture (2011).
[22] H. Mun, K. Han, K. Kim, 3G-WLAN interworking: security analysis and new authentication and key agreement based on EAP-AKA, in: Proceedings of the of IEEE Wireless Telecommunications Symposium WTS, 2009, pp. 1–8.
[23] C. Ntantogian, C. Xenakis, One-pass EAP-AKA authentication in 3G-WLAN integrated networks, Wirel. Pers. Commun. 48 (4) (2009) 569–584.
[24] K. Ahmavaara, H. Haverinen, R. Pichna, Interworking architecture between 3GPP and WLAN systems, IEEE Commun. Mag. 41 (11) (2003) 74–81.
[25] P. Prasithsangaree, P. Krishnamurthy, A new authentication mechanism for loosely coupled 3G-WLAN integrated networks, in: Proceedings of the of IEEE Vehicular Technology Conference, 5, 2004, pp. 2998–3003.
[26] A.A. Al Shidhani, V.C. Leung, Fast and secure reauthentications for 3GPP subscribers during WiMAX-WLAN handovers, IEEE Trans. Dependable Secur. Comput. 8 (5) (2011) 699–713.
[27] S. Othmen, F. Zarai, M.S. Obaidat, A. Belghith, Re-authentication protocol from WLAN to LTE (ReP WLAN-LTE), in: Proceedings of the of IEEE Global Communication Conference GLOBECOM, 2013, pp. 1446–1451.
[28] R. Jiang, C. Lai, J. Luo, X. Wang, H. Wang, EAP-Based Group Authentication and Key Agreement Protocol for Machine-Type Communications, Int. J. Distrib. Sensor Netw. 2013 (2013) 1–14.
[29] T. Yang, C. Lai, R. Lu, R. Jiang, EAPSG: Efficient authentication protocol for secure group communications in maritime wideband communication networks, Peer-to-Peer Networking and Applications (2014) 1–14.
[30] J.-L. Huang, L.-Y. Yeh, H.-Y. Chien, ABAKA: An anonymous batch authenticated and key agreement scheme for value-added services in vehicular ad hoc networks, IEEE Trans. Veh. Technol. 60 (1) (2011) 248–262.
[31] J. Katz, A. Lindell, Aggregate message authentication codes, in: Topics in Cryptology–CT-RSA 2008, Springer, 2008, pp. 155–169.
[32] V. Kolesnikov, W. Lee, J. Hong, MAC aggregation resilient to DoS attacks, in: Proceedings of the IEEE International Conference on Smart Grid Communications (SmartGridComm), 2011, pp. 226–231.
[33] V. Kolesnikov, MAC aggregation with message multiplicity, in: Security and Cryptography for Networks, Springer, 2012, pp. 445–460.
[34] V. Fajardo, J. Arkko, J. Loughney, G. Zorn, Diameter Base Protocol, Technical Report, Internet Engineering Task Force (IETF), 2012.
[35] M.J. Moyer, J.R. Rao, P. Rohatgi, A survey of security issues in multicast communications, IEEE Netw. 13 (6) (1999) 12–23.
[36] S. Rafaeli, D. Hutchison, A survey of key management for secure group communication, ACM Comput. Surv. CSUR 35 (3) (2003) 309–329.
[37] C. Zhang, R. Lu, X. Lin, P.-H. Ho, X. Shen, An efficient identity-based batch verification scheme for vehicular sensor networks, in: Proceedings of the IEEE Conference on Computer Communication INFOCOM, 2008.

Chengzhe Lai received his degree in B.S. in Information Security from Xi'an University of Posts and Telecommunications in 2008 and a Ph.D. degree from Xidian University in 2014. He was a visiting Ph.D. student with the Broadband Communications Research (BBCR) Group, University of Waterloo from 2012 to 2014. At present, he is with the School of Telecommunication and Information Engineering, Xi'an University of Posts and Telecommunications and with the National Engineering Laboratory for Wireless Security, Xi'an, China. He is also a visiting researcher of the State Key Laboratory of Integrated Services Networks and State Key Laboratory of Information Security. His research interests include wireless network security, privacy preservation, and M2M communications security.

Rongxing Lu received his Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China in 2006 and a Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2012. He is currently an Assistant Professor with the Division of Communication Engineering, School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore. His research interests include wireless network security, applied cryptography, and trusted computing.

Dong Zheng received an M.S. degree in mathematics from Shaanxi Normal University, Xi'an, China, in 1988, and a Ph.D. degree in communication engineering from Xidian University, Xi'an, in 1999. He was a Postdoctoral Fellow in the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, from 1999 to 2001 and a Research Fellow at Hong Kong University, Hong Kong, in 2002. He was a Professor in the School of Information Security Engineering, Shanghai Jiao Tong University. He is also with the State Key Laboratory of Integrated Service Networks, Xidian University. He is currently a Professor in the School of Telecommunication and Information Engineering, Xi'an University of Posts and Telecommunications and is also connected with the National Engineering Laboratory for Wireless Security, Xi'an, China. His research interests include provable security and new cryptographic technology.

Hui Li received B.Sc. degree from Fudan University in 1990, M.A.Sc. and Ph.D. degrees from Xidian University in 1993 and 1998. Since June 2005, he has been the professor in the school of Telecommunications Engineering, Xidian University, Xi'an Shaanxi, China. His research interests are in the areas of cryptography, wireless network security, information theory and network coding. He is a co-author of two books. He served as technique committee co-chairs of ISPEC 2009 and IAS 2009.

Xuemin (Sherman) Shen received his B.Sc. degree from Dalian Maritime University, China, in 1982, and M.Sc. and Ph.D. degrees from Rutgers University, New Jersey, in 1987 and 1990, all in electrical engineering. He is a professor and university research chair in the Department of Electrical and Computer Engineering, University of Waterloo. His research focuses on resource management in interconnected wireless/wired networks, UWB wireless communications networks, wireless network security, wireless body area networks, and vehicular ad hoc and sensor networks. He is a co-author of three books, and has published more than 400 papers and book chapters in wireless communications and networks, control, and filtering. He is Editor-in-Chief of IEEE Network, and will serve as a Technical Program Committee Co-Chair for IEEE INFOCOM 2014. He is the Chair of the IEEE ComSoc Technical Committee on Wireless Communications, and P2P Communications and Networking, and a voting member of GITC. He was a Founding Area Editor for IEEE Transactions on Wireless Communications, and a Guest Editor for IEEE JSAC, IEEE Wireless Communications, and IEEE Communications Magazine. He also served as the Technical Program Committee Chair for GLOBECOM'07, Tutorial Chair for ICC'08, and Symposia Chair for ICC'10. He received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004, 2007, and 2010 from the University of Waterloo, and the Premier's Research Excellence Award in 2003 from the Province of Ontario, Canada. He is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, a Fellow of the Engineering Institute of Canada, a Fellow of Canadian Academy of Engineering, and was a ComSoc Distinguished Lecturer.