Computer Networks 57 (2013) 3492–3510

Contents lists available at ScienceDirect

Computer Networks

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

SE-AKA: A secure and efficient group authentication and key agreement protocol for LTE networks

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

[a] State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China
[b] Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada
[c] Division of Communication Engineering, School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Article history:
Received 27 January 2013
Received in revised form 31 July 2013
Accepted 6 August 2013
Available online 14 August 2013

Keywords:
Authentication and key agreement (AKA)
Privacy
Group communication
Long Term Evolution (LTE)
3rd Generation Partnership Project (3GPP)

ABSTRACT

To support Evolved Packet System (EPS) in the Long Term Evolution (LTE) networks, the 3rd Generation Partnership Project (3GPP) has proposed an authentication and key agreement (AKA) protocol, named EPS-AKA, which has become an emerging standard for fourth-generation (4G) wireless communications. However, due to the requirement of backward compatibility, EPS-AKA inevitably inherits some defects of its predecessor UMTS-AKA protocol that cannot resist several frequent attacks, i.e., redirection attack, man-in-the-middle attack, and DoS attack. Meanwhile, there are additional security issues associated with the EPS-AKA protocol, i.e., the lack of privacy-preservation and key forward/backward secrecy (KFS/KBS). In addition, there are new challenges with the emergence of group-based communication scenarios in authentication. In this paper, we propose a secure and efficient AKA protocol, called SE-AKA, which can fit in with all of the group authentication scenarios in the LTE networks. Specifically, SE-AKA uses Elliptic Curve Diffie-Hellman (ECDH) to realize KFS/KBS, and it also adopts an asymmetric key cryptosystem to protect users' privacy. For group authentication, it simplifies the whole authentication procedure by computing a group temporary key (GTK). Compared with other authentication protocols, SE-AKA cannot only provide strong security including privacy-preservation and KFS/KBS, but also provide a group authentication mechanism which can effectively authenticate group devices. Extensive security analysis and formal verification by using ***proverif*** have shown that the proposed SE-AKA is secure against various malicious attacks. In addition, elaborate performance evaluations in terms of communication, computational and storage overhead also demonstrates that SE-AKA is more efficient than those existing protocols.

1. Introduction

With the development of mobile communication systems, numerous authentication and key agreement (AKA) protocols have been proposed. To improve the security weaknesses in Global System for Mobile Communications (GSM) [1], UMTS-AKA, which is based on GSM's successor Universal Mobile Telecommunications System (UMTS), was proposed at the network level [2] for authenticating 3G mobile subscribers. UMTS-AKA can negotiate security keys between a subscriber and the serving network and then achieve mutual authentication between the two parties. UMTS-AKA can also successfully defeat most of the vulnerabilities found in GSM systems and ensure a more secure telecommunication environment. Nevertheless, it is still vulnerable to some sophisticated attacks, such as redirection and man-in-the-middle

* Corresponding author.
E-mail addresses: lcz.xidian@gmail.com (C. Lai), lihui@mail.xidian.edu.cn (H. Li), rxlu@ntu.edu.sg (R. Lu), xshen@bbcr.uwaterloo.ca (Xuemin (Sherman) Shen).

1389-1286/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2013.08.003

attacks. Recently, a novel authentication protocol dedicated for Evolved Packet System (EPS) has been proposed in the Long Term Evolution (LTE) project [3] by the 3rd Generation Partnership Project (3GPP), known as EPS-AKA [4], which is based on its predecessor UMTS-AKA protocol. Backward compatibility of EPS-AKA is an important factor for its wide acceptance, but it may also hinder progress and limit the design freedom. On one hand, EPS-AKA inevitably inherits some defects of UMTS-AKA and cannot resist known typical attacks found in UMTS-AKA, i.e., redirection attack that is discussed by [5,24], man-in-the-middle attack which is studied in [6,30], and DoS attack that is given in [29,30]; on the other hand, there are some additional security issues associated with the EPS-AKA protocol that cannot be neglected, i.e., the lack of privacy-preservation and key forward/backward secrecy (KFS/KBS). Although most of the existing studies of mobile communication protocols have focused on confidentiality and authentication requirements, yet privacy-preservation [7–9], another important issue in mobile communication networks, has not been well addressed. Recently, Arapinis et al. [10] highlight the privacy problems of the 3G network, they exposed two novel threats to the user privacy in 3G telephony systems, i.e., IMSI paging attack and AKA protocol linkability attack, which make it possible to trace and identify mobile telephony subscribers. At the same time, they propose amendments to these privacy issues. Moreover, EPS-AKA still uses a symmetric key K shared between the user equipment and the home subscriber server to perform authentication and key agreement. All subkeys are generated using K. Therefore, disclosure of K is equal to the disclosure of whole procedure of EPS-AKA, i.e., EPS-AKA does not provide KFS/KBS.[1]

With the emergence of group-based communication scenarios, there are a large number of user terminals with the same properties in a network, e.g., machine-type communication (MTC) [11–13]. These kinds of devices can form a group when they are in the same region, belong to the same applications, etc. [14–17]. If a large number of devices in a group need to access the network successively over a short period of time, available authentication methods will suffer from high network access latency until completing authentication procedures of all devices in the same group, especially when these devices roam in a visited domain which is far from their home domain. The reason is that every device must perform a full AKA authentication procedure with home authentication server, so authentication signaling in the network will increase. Meanwhile, the overload of home authentication server will increase due to frequently generating authentication vectors. To the best of our knowledge, most of existing authentication schemes on 3G/LTE networks do not have group authentication mechanism and are not suitable for the authentication of group-based communications, and few authentication protocols for group communications have been proposed. Ngo et al. [18] develop an individual and group authentication model for wireless network services, which uses dynamic key cryptography and group key management to provide authentication for individual and group of users and services; Aboudagga et al. [19] present an associated authentication protocol for mobile groups and individual nodes over heterogeneous domains. However, they are designed for specific scenarios and lack of universality. Recently, Fun et al. propose a novel group-based handover authentication scheme with privacy preservation for mobile WiMAX networks [20]. This scheme improves performance of group-based handover authentication in mobile WiMAX networks. However, it has not discussed the existing attacks, meanwhile, it is designed for WiMAX networks and may not be suitable for LTE networks. Cao et al. [21] propose a group-based authentication and key agreement for MTC in LTE networks, which can effectively authenticate a group of devices at the same time. However, their scheme is totally based on asymmetric cryptography by adopting bilinear pairing technique, which is costly in computation and may not be suitable for resource-constrained mobile device in LTE networks.

Considering security, effectiveness and universality simultaneously, we propose a secure and efficient authentication and key agreement protocol, called SE-AKA, for LTE networks in this paper. The main contributions of this paper are as follows.

- First, SE-AKA meets the security requirements defined in EPS-AKA and can resist the existing attacks including redirection, man-in-the-middle and denial-of-service attacks, etc. Besides, motivated by the research done by Arapinis et al. [10], we adopt an asymmetric key cryptosystem to enhance user's privacy-preservation in LTE networks. In addition, SE-AKA can guarantee KFS/KBS through combining Elliptic Curve Diffie-Hellman (ECDH). Furthermore, we use automatical analyzing tool ***ProVerif*** [22] to verify the security of SE-AKA to show its security strength.
- Second, the group authentication mechanism is designed which can efficiently authenticate devices in a group compared with the traditional protocols. The results of analysis show that the transmission overhead of the whole authentication is considerably reduced. The computational overhead of home subscriber server and the storage overhead in the serving network can also be decreased.
- Third, SE-AKA is proposed based on LTE network infrastructure which can fit in with all of the scenarios for performing group-based authentication in the LTE networks.

The remainder of this paper is organized as follows: In Section 2, we discuss the related works. In Section 3, we review the EPS-AKA protocol, introduce our network architecture, and recall Elliptic Curve Diffie-Hellman [32] as the preliminaries. Then, we present our SE-AKA protocol in Section 4, followed by its security analysis and performance evaluations in Section 5 and Section 6, respectively. Finally, We conclude this paper with remarks about future work in Section 7.

[1] The KFS is that any preceding key could not be disclosed if the long-term secret key K is compromised, and the KBS is that any following key could not be disclosed if the long-term secret key K is compromised.

2. Related work

There have been many research works on authentication and key agreement protocols in 3G/LTE networks. In 2003, Harn and Hsin [23] used the concept of hash chain and message authentication code (MAC) to design an ER-AKA protocol, which is expected to enhance the security of the original UMTS-AKA protocol. However, the protocol has greatly increased space and communication overhead in the hash chain's storage and transmission.

In 2005, Zhang and Fang [24] pointed 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 AP-AKA, it allows the entities to have the flexibility of selecting execution flows dependent on the MSs in the foreign networks (FNs) and the HN. Lee et al. [25] extend AP-AKA to make it more efficient. They found that the AP-AKA for 3GPP has three drawbacks as follows: (1) The FN must turn back to the HN for a request of another set of AVs when the MS stays in the FN for a long time and exhausts its set of AV for authentication. Additional, bandwidth consumption therefore is introduced between the FN and HN; (2) Each MS in the particular FN has n copies of AV. If there are m MSs in the FN, the FN must store m n authentication vectors. This is extra space overhead; and (3) When the n copies AVs are all consumed, FN must go back to HN to get another n copies AVs to authenticate MS. In this way, the authentication of an MS cannot be completed without the help by the HN of the MS, for each communication when the n copies are all used.

In 2005, X-AKA [26], a symmetric key-based authentication protocol, is proposed to prune off the transmission of AVs in UMTS-AKA and improve its bandwidth consumption. However, it does not resist redirection and man-in-the-middle attacks. Al-Saraireh and Yousef [27] design a symmetric key-based authentication protocol for UMTS networks. Al-Saraireh and Yousef's protocol mainly focuses on reducing the bandwidth required for transmitting AVs. Hence, the AVs are generated by MSs instead of by serving networks. Al-Saraireh and Yousef's protocol eliminates the cost of delivering AVs during authentication. The protocol, however, does not resolve the security issues in defeating redirection and man-in-the-middle attacks.

In 2010, Ou et al. [28] 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 [29].

In 2011, Huang et al. [30] 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, similar to other existing protocols, the protocol is not suitable for group-based communications due to lack of special group authentication mechanism.

Chen et al. [31] 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 also cannot provide enough security and is vulnerable to redirection, man-in-the-middle attacks, etc.

Different from above works, our focus is on providing a more secure, effective and universal AKA protocol for LTE networks. First, SE-AKA can resist all existing attacks found in previous works, and provide enhanced user's privacy-preservation and KFS/KBS that cannot be guaranteed by previous works. Second, it can provide the group authentication mechanism which can efficiently authenticate devices in a group. Third, SE-AKA can fit in the LTE networks with all of the scenarios for performing group-based authentication.

3. Preliminaries

3.1. Review of the EPS-AKA protocol

In this section, we first introduce EPS-AKA authentication procedure, which was proposed in the 3GPP release 9 for LTE networks. EPS-AKA can broadly be divided into two stages: (1) authentication data distribution, and (2) user authentication and key agreement. The former enables the home network (HN) of an mobile equipment (ME) to distribute authentication data to the serving network (SN) the ME device is visiting. The latter is to establish new session keys between the ME and the SN. The EPS-AKA protocol works as follows.

(1) An ME sends an access request message to the SN;
(2) Upon receiving access request by an ME, the SN launches an authentication procedure by asking the ME's identity;
(3) In response to the SN, the ME sends its identity to the SN;
(4) The SN sends an authentication data request message containing ME's identity to the HN for acquiring AVs;
(5) The HN first generates AVs for the SN, an authentication vector comprising a RAND, XRES, AUTN and K_{ASME} insteading of IK and CK in UMTS AV, which is the main difference between the EPS AV and UMTS AV. The AV is expressed as $AV = RAND\|XRES\|K_{ASME}\|AUTN$. $AUTN$ is calculated as $AUTN = SQN \oplus AK\|AMF\|MAC$. In order to prevent a UMTS AV attacker to

impersonate the EPS network, EPS AV and UMTS AV need to be isolated. At present, 3GPP uses the AMF in the AV to identify the network which the AV belongs to;

(6) The HN sends back an authentication data request message including the generated AV (for the corresponding ME) so that the SN is authorized to authenticate the requesting ME;
(7) Upon receipt of authentication vectors, the SN sends RAND and AUTN piggy-backed on authentication request to the ME, enabling the ME to verify the correctness of SQN and compute the RES;
(8) The ME verifies the correctness of SQN by computing MAC and comparing it with the MAC carried in AUTN. If matched, the ME computes and sends the corresponding response RES back to the SN in an authentication response message;
(9) Once the SN receives and verifies RES correctly, it chooses the corresponding K_{ASME} as the session key to protect its communication with the ME. In addition, the ME calculates its K_{ASME} accordingly. Hence both the ME and SN reach a common session key, which terminates the EPS-AKA protocol.

3.2. Network architecture

Fig. 1 shows our considered network architecture in the roaming scenario which is based on 3GPP standard [4], and can be divided into three domains, namely access network domain, serving network domain and home network domain. The main entities involved in the network architecture are presented in Table 1.

3.2.1. Access network domain

Access network domain mainly consists of MEs and base station (BS). An ME can be any kind of 3GPP standard mobile devices. Moreover, HeNB and eNB are two kinds of BSs for MEs to access 3GPP network. Different from the eNB, an HeNB is typically installed by a subscriber in residence or a small office to increase the indoor coverage for voice and high speed data service.

3.2.2. Serving network domain

The *serving network* (SN) provides access services for MEs. In the LTE network, the MME locates in SN and provides access services for MEs. The MME is responsible for all the functions relevant to the users and the control plane session management. When an ME connects to the SN, the MME firstly contacts with the HSS to obtain the corresponding authentication data and then represents the SN to perform a mutual authentication with the ME.

Table 1
Main entities involved in the network architecture.

Entity	Abbreviation
Mobile Equipment	ME
Evolved Node B	eNB
Home Evolved Node B	HeNB
Mobile Management Entity	MME
Serving Gateway	S-GW
Home Subscriber Server	HSS
Group Management Server	GMS

3.2.3. Home network domain

The *home network* (HN) provides authentication and management services for MEs. In the LTE network, the HSS locates in HN and provides authentication and management services for MEs. In addition, we add a new server to home network domain, named group management server (GMS), to manage the group that MEs form, e.g., in the MTC, MTC server can implement this function. The interfaces between GMS and HSS/MME are secure, since the GMS locates in the trusted HN regulated by the operator.

3.3. Elliptic Curve Diffie-Hellman

In this work, we use Elliptic Curve Diffie-Hellman (ECDH) to realize KFS/KBS. ECDH can be described as follows: Alice and Bob publicly agree on an elliptic curve E over a large finite field $\mathbb{F}_q$ and a point P on that curve. Then, Alice and Bob each selects random numbers a and b, respectively. Using elliptic curve point-addition, Alice and Bob each publicly compute aP and bP on E. Then, Alice and Bob send their own computed values to each other. When Alice receives bP, she computes $a(bP)$. Similarly, when Bob receives aP, he computes $b(aP)$. Finally, Alice and Bob agree a shared secret abP. The shared secret calculated by both parties is equal, because $a(bP) = abP = baP = b(aP)$ [32]. However, the original ECDH is insecure and

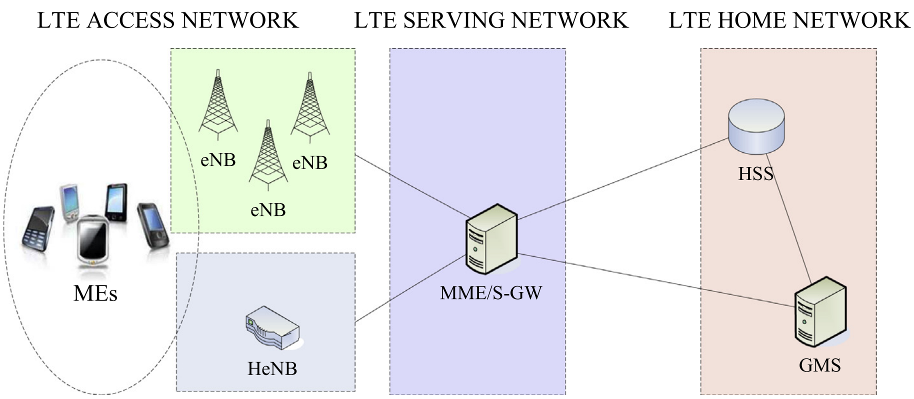


Fig. 1. Network architecture.

vulnerable to man-in-the-middle (MITM) attack. Krawczyk [33] proposed a provable secure and efficient DH key exchange approach, named SIGn-and-MAc (SIGMA) to solve this problem.

4. Proposed authentication protocol

In this section, we propose a secure and efficient authentication and key agreement protocol for LTE networks (SE-AKA) to facilitate the ME/MEs that have been subscribed in the HN to roam in an SN which is far from HN. Table 2 shows the used notations in the SE-AKA protocol.

4.1. Preparation and initialization

- Each ME has an identity (PID_{ME}/TID_{ME}) which is a private identity that identifies ME and should be installed in the ME by the supplier in order to allow the ME to register in a 3GPP network.
- Each ME has a pre-shared secret key with HN when it is first registered in HN.
- A lightweight public key infrastructure (PKI) [10] is adopted to provide each HN with a private/public key pair (Pub_{HN},Pri_{HN}). The public key of HN can be stored in ME's trusted environment, Universal Subscriber Identity Module (USIM)/Universal Integrated Circuit Card (UICC). This public key makes it possible for an ME to encrypt privacy related information such as International Mobile Subscriber Identification Number (IMSI), and deliver them to the network in a confidential manner.
- The MEs form several groups based on certain principles (belonging to one and the same application/ within the same region/ having the same behavior), using the grouping algorithm [34,35], then the supplier provides a group key (GK) to each group for authentication. As shown in Table 3, we create a Group Information Management List (GIML) to manage information of MEs and groups, the GIML contains fields of group identity, ME temporary identity (TID_{ME}) for each ME[2] and the large and unique synchronization value SV which will behave as a sequence number for synchronization between the ME and its SN.

4.2. Protocol execution for the first equipment

(**a1**)–(**a5**) describe how the MME distributes authentication data for the first ME of the group visiting the SN. A secure communication channel between the SN and the HN has already been established (based on Diameter protocol [36]) and can provide security services to the transmitted data. Let ME_{G1-1} be the first ME initiating authentication in the group G1. Our authentication protocol is shown in Figs. 2 and 3, and the detailed steps are as follows.

(**a1**) ME_{G1-1} → MME: **Access Request**.
(**a2**) MME → ME_{G1-1}: **Identity Request**.
(**a3**) ME_{G1-1} → MME: ($AUTH_{G1}$).
ME_{G1-1} generates $AUTH_{G1}$ as follows: $AUTH_{G1} = (ID_{G1}\|TID_{ME_{G1-1}}\|R_{G1-1}\|MAC_{G1}\|T_{G1})$, where MAC_{G1} is calculated as $MAC_{G1} = f^1_{key_{G1-1}}(ID_{G1}\|TID_{ME_{G1-1}}\|R_{G1-1}\|T_{G1}\|LAI)$. Since $TID_{ME_{G1-1}}$ represents ME_{G1-1}'s temporary identity, if HN needs to require ME_{G1-1}'s permanent identity ($PID_{ME_{G1-1}}$) when necessary, $\{PID_{ME_{G1-1}}\}\_Pub_{HN}$ will be sent to HN.
(**a4**) MME → HSS: **Authentication Data Request** ($AUTH_{G1}$, LAI).
When the HSS receives authentication data request message contained ME_{G1-1}'s $AUTH_{G1}$, the HSS verifies the received MAC_{G1} in $AUTH_{G1}$ using key_{G1-1}. Since 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. By checking MAC_{G1}, the HSS can verify whether the LAI reported by the MME is the same as that recognized by the ME.
(**a5**) HSS → MME: **Authentication Data Response** ($AUTH_{HSS}$).
Once verification passes, the HSS retrieves the corresponding group key GK_{G1} to generate a group temporary key $GTK_{G1} = f^2_{GK_{G1}}(R_{HSS}\|AMF)$. Then the HSS

Table 2
Protocol notation.

Notation	Definition
R_x	The rand number generated by x
T_x	The time stamp generated by x
PID_x	The permanent identity of x
TID_x	The temporary identity of x
key_{x-y}	The shared secret key between x and y
GK_{Gi}	The group authentication key of the ith group
GTK_{Gi}	The group temporary key of the ith group
$KGK_{ME_{Gi-j}}$	The key generation key between ME_{Gi-j} and SN
MAC_x	The message authentication code computed by x
LAI	Location area identification
AMF	Authentication management field
f^1_k	MAC generation function using k
f^2_k	GTK generation function using k
f^3_k	KGK generation function using k

Table 3
Group information management list.

Group	Group ID	ME ID	Synchronization value
G1	ID_{G1}	TID_{G1-1}	SV_{G1-1}
		TID_{G1-2}	SV_{G1-2}
		⋮	⋮
		TID_{G1-n}	SV_{G1-n}
G2	ID_{G2}	TID_{G2-1}	SV_{G2-1}
		⋮	⋮

[2] Note that, to ensure user identity privacy, the permanent identity of an ME like IMSI should be confidentiality protected. It should never be transmitted in plain text. In EPS-AKA, a Globally Unique Temporary Identity (GUTI, ME's temporary identity) is transmitted instead of the IMSI for identity presentation. In spite of this security arrangement, there are occasions when the IMSI may be transmitted in plain text. We will discuss the solutions when the IMSI needs to be transmitted in the channel later.

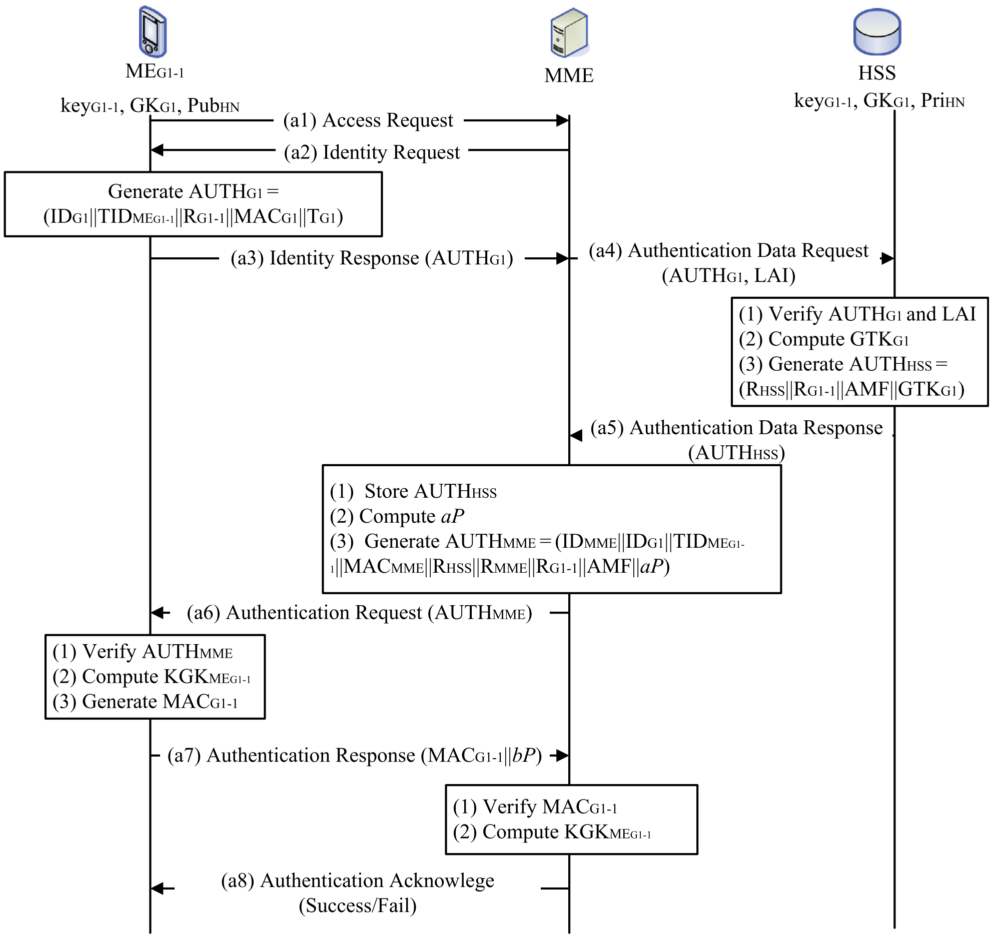


Fig. 2. The SE-AKA protocol.

generates $AUTH_{HSS}$, $AUTH_{HSS} = (R_{HSS}\|R_{G1-1}\|AMF\|GTK_{G1})$, and sends $AUTH_{HSS}$ together with G1's information stored in the Group Information Management List (Table 3) to the MME. The MME receives and stores them for future use.

(**a6**)–(**a8**) are authentication and key agreement phase.

(a6) MME → ME_{G1-1}: **Authentication Request** ($AUTH_{MME}$).

After acquiring $AUTH_{HSS}$ for group G1, the MME selects random number a and computes aP on E. Then the MME performs mutual authentication with ME_{G1-1} by generating $AUTH_{MME}$ as follows: $AUTH_{MME} = (ID_{MME}\|ID_{G1}\|TID_{ME_{G1-1}}\|MAC_{MME}\|R_{HSS}\|R_{MME}\|R_{G1-1}\|AMF\|ap)$, where $MAC_{MME} = f^1_{GTK_{G1}}(ID_{MME}\|ID_{G1}\|TID_{ME_{G1-1}}\|R_{MME}\|R_{HSS}\|R_{G1-1}\|AMF\|ap\|SV_{G1-1}+i)$, where SV_{G1-1} can be got from (a5) and i represents the ith run of mutual authentication with ME_{G1-1}.

(a7) ME_{G1-1} → MME: **Authentication Response** ($MAC_{G1-1}\|bP$).

On receiving the message from the MME, ME_{G1-1} verifies the received MAC_{MME} in $AUTH_{MME}$ as follows:

(1) ME_{G1-1} computes $GTK_{G1} = f^2_{GK_{G1}}(R_{HSS}\|AMF)$;
(2) ME_{G1-1} computes $MAC'_{MME} = f^1_{GTK_{G1}}(ID_{MME}\|ID_{G1}\|TID_{ME_{G1-1}}\|R_{MME}\|R_{HSS}\|R_{G1-1}\|AMF\|ap\|SV_{G1-1}+i)$;
(3) The ME_{G1-1} verifies whether MAC'_{MME} equals MAC_{MME} or not. If MAC'_{MME} is not the same as MAC_{MME}, the HSS or the MME server is not valid. Therefore, the ME_{G1-1} terminates the procedure and sends MAC failure (Mac_Fail) message. Meanwhile, the ME_{G1-1} will send $\{FAIL, PID_{ME_{G1-1}}, rand\}\_Pub_{HN}$ to require a new MAC verification.

If verification passes, ME_{G1-1} selects random number b and computes bP on E, and calculates $KGK_{ME_{G1-1}} = f^3_{GTK_{G1}}(ID_{MME}\|TID_{ME_{G1-1}}\|R_{MME}\|R_{G1-1}\|abP)$ for subsequent sessions with the MME and $MAC_{ME_{G1-1}} = f^1_{KGK_{ME_{G1-1}}}(ID_{MME}\|ID_{G1}\|TID_{ME_{G1-1}}\|R_{MME}\|abP\ \|bP\|SV_{G1-1}+i)$;

(a8) MME → ME_{G1-1}: **Authentication Acknowledge**.

When the MME receives an authentication response message carrying $MAC_{ME_{G1-1}}$, it also computes $KGK_{ME_{G1-1}} = f^3_{GTK_{G1}}(ID_{MME}\|TID_{ME_{G1-1}}\|R_{MME}\|R_{G1-1}\|abP)$. Then it checks whether ME_{G1-1} has generated the correct response. If verification is successful, it sends authentica-

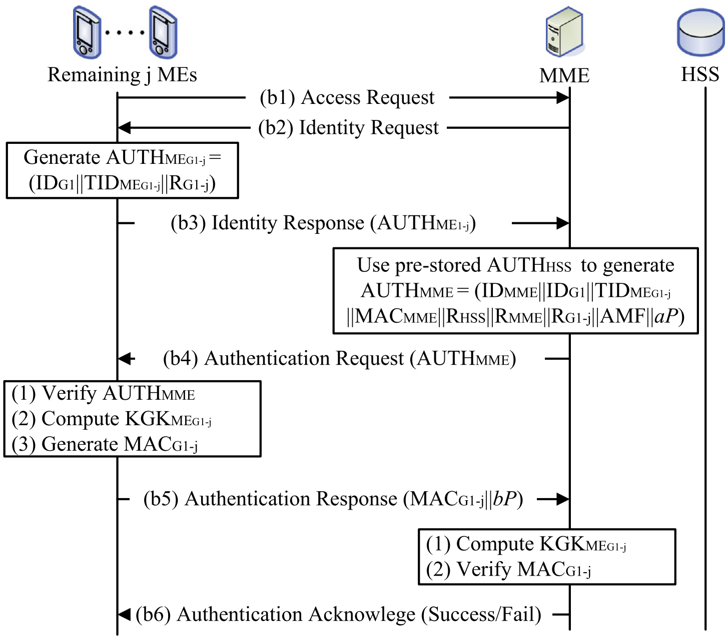


Fig. 3. The authentication procedure of remaining MEs.

tion acknowledge to ME_{G1-1}, and the full authentication and key agreement procedure for the first ME is completed.

After a successful authentication, both ME_{G1-1} and its SN share a key generation key $KGK_{ME_{G1-1}}$ as essential material for subsequent key derivations. $KGK_{ME_{G1-1}}$'s function is the same as K_{ASME}'s [4].

4.3. Protocol execution for the remaining equipments of the same group

When the second ME (ME_{G1-2}) in the same group wants to access the serving network, the MME performs mutual authentication and key agreement with ME_{G1-2} locally using the existing GTK_{G1}.

Step (**b1**)–(**b2**) are similar to ME_{G1-1}'s.

(**b3**) $ME_{G1-2} \rightarrow$ MME: $(AUTH_{ME_{G1-2}})$.
ME_{G1-2} generates $AUTN_{ME_{G1-2}}$ as follows: $AUTH_{ME_{G1-2}} = (ID_{G1}\|TID_{ME_{G1-2}}\|R_{G1-2})$. Note that, ME_{G1-2} does not need to send MAC_{G1} to the MME, because the MME can authenticate ME_{G1-2} directly without the HSS's assistance, therefore MAC_{G1} does not need to be used. The ME can perform mutual authentication with MME directly.
Similarly, $TID_{ME_{G1-2}}$ represents ME_{G1-2}'s temporary identity, if HN needs to require ME_{G1-2}'s permanent identity, $\{PID_{ME_{G1-2}}\}\_Pub_{HN}$ will be sent instead.

(**b4**) MME $\rightarrow ME_{G1-2}$: **Authentication Request** $(AUTH_{MME})$.
When the MME receives $AUTH_{ME_{G1-2}}$, it first selects random number a and computes aP on E. Then, the MME begins to perform mutual authentication with ME_{G1-2} by generating $AUTH_{MME}$ as follows: $AUTH_{MME} = (ID_{MME}\|ID_{G1}\|TID_{ME_{G1-2}}\|MAC_{MME}\|R_{HSS}\|R_{MME}\|R_{G1-2}\|AMF)$, where $MAC_{MME} = f^1_{GTK_{G1}}(ID_{MME}\|ID_{G1}\|TID_{ME_{G1-2}}\|R_{MME}\|R_{HSS}\|R_{G1-2}\|AMF\|ap\|SV_{G1-2}+i)$. i represents the i-th run of mutual authentication with ME_{G1-2}.

(**b5**) $ME_{G1-2} \rightarrow$ MME: **Authentication Response** $(MAC_{G1-2}\|bP)$.
On receiving the message from the MME, ME_{G1-2} verifies the received MAC_{MME} in $AUTH_{MME}$ as follows:
(1) ME_{G1-2} computes $GTK_{G1} = f^2_{GK_{G1}}(R_{HSS}\|AMF)$;
(2) ME_{G1-2} computes $MAC'_{MME} = f^1_{GTK_{G1}}(ID_{MME}\|ID_{G1}\|TID_{ME_{G1-2}}\|R_{MME}\|R_{HSS}\|R_{G1-2}\|AMF\|ap\|SV_{G1-2}+i)$;
(3) The ME_{G1-2} verifies whether MAC'_{MME} equals MAC_{MME} or not. If MAC'_{MME} is not the same MAC_{MME}, the HSS or the MME server is not valid. Therefore, the ME_{G1-2} terminates the procedure and sends MAC failure (Mac_Fail) message. Meanwhile, the ME_{G1-2} will send $\{FAIL, PID_{ME_{G1-2}}, rand\}\_Pub_{HN}$ to require a new MAC verification.

If verification passes, ME_{G1-2} selects random number b and computes bP on E, and calculates $KGK_{ME_{G1-2}} = f^3_{GTK_{G1}}(ID_{MME}\|TID_{ME_{G1-2}}\|R_{MME}\|R_{G1-2}\|abP)$ for subsequent sessions with the MME and $MAC_{ME_{G1-2}} = f^3_{KGK_{ME_{G1-2}}}(ID_{MME}\|ID_{G1}\|TID_{ME_{G1-2}}\|R_{MME}\|LAI\|bP\|abP\|SV_{G1-2}+i)$;

(**b6**) MME $\rightarrow ME_{G1-2}$: **Authentication Acknowledge**.
When the MME receives an authentication response message carrying $MAC_{ME_{G1-2}}$, it also computes $KGK_{ME_{G1-2}} = f^3_{GTK_{G1}}(ID_{MME}\|TID_{ME_{G1-2}}\|R_{MME}\|R_{G1-2}\|abP)$, then it checks whether ME_{G1-2} has generated the correct response. Since the MME knows the LAI' of the BS forwarding $AUTH_{ME_{G1-2}}$, it can verify whether the LAI' forwarded by the BS is the same as that recognized by the ME_{G1-2} through by checking MAC_{G1-2}.

The remaining MEs perform the authentication and key agreement procedures similar to ME_{G1-2}'s until all devices complete authentication.

4.4. Group member joining/leaving the group

The group which formed by MEs requires backward and forward secrecy. Backward secrecy is required that a new ME cannot get messages exchanged before it joined the group. Forward secrecy is required that a leaving or expelled ME cannot continue accessing the group's communication (if it keeps receiving the messages).

In this paper, we can use the GMS to manage the group member joining/leaving the group. When an ME wants to leave the group, the GMS will revoke the binding relationship between the ME and the group that it belongs to, thus the ME cannot longer communicate with the SN as the group member. Moreover, in order to prevent the old ME to decrypt the new packets of the group which it was able to sniff, the group key must be updated when the old ME leaves the group. After the old ME leaves the group, all members of the group should share a new group key. Similarly, when an ME 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 SN. Meanwhile, the group key must be updated when the new ME wants to join a group. After the new ME joins the group, all members of the group should share a new group key. In that case, the new ME 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 reference [37,38].

5. Security analysis

In this section, both security analysis and formal verification are conducted to demonstrate that SE-AKA can meet the security requirements.

5.1. Security analysis

The SE-AKA protocol adopts the same secured architecture as the EPS-AKA protocol. Therefore, it has the same security threshold in most situations. SE-AKA protocol can reach same security requirements with EPS-AKA protocol as follows:

5.1.1. Entity mutual authentication

In the proposed SE-AKA protocol, an ME is identified by its PID_{ME}/TID_{ME} and group ID ID_{G_i}. The first ME ME_{Gi-1} uses $AUTH_{Gi}$ to get $AUTH_{HSS}$ containing GTK for group G_i from HSS in the HN and performs a mutual authentication with HN. MAC_{Gi} is only generated by ME_{Gi-1} using pre-shared key_{Gi-1} with the HN, at the same time, ME_{Gi-1} can authenticate the HN by the unique GTK sourced from the real HN. Moreover, a mutual authentication between ME_{Gi-1} and its SN is also carried out. This is because ME_{Gi-1} authenticates its SN by comparing its computed MAC_{MME} with that in $AUTH_{MME}$, the SN only acquires a correct GTK from the HN to prove itself legitimate. On the other hand, the SN can authenticate ME_{Gi-1} by checking whether the returned $MAC_{ME_{G_{i-1}}}$ from ME_{Gi-1} is correct. Note that, a secure communication channel between the MME and the HSS has already been established and can provide security services to the transmitted data. For the remaining MEs in the same group, they only need to perform mutual authentications with their SN locally.

5.1.2. Confidentiality

Confidentiality includes cipher algorithm agreement, cipher key agreement, confidentiality of user data and confidentiality of signaling data. Our SE-AKA protocol follows the mechanism of the EPS-AKA protocol and is successful with these demands. The SN carries the field of the AMF in the $AUTH_{MME}$ to meet the feature of cipher algorithm agreement. The random numbers and the identities collocate with a group key to make the feature of cipher key agreement (GTK, see Section 4.2). All the user data and signaling data will be encrypted with the subsequent key drived from KGK that the ME and the SN agree on in each time session.

5.1.3. Data integrity

Data integrity includes integrity algorithm agreement, integrity key agreement, data integrity and origin authentication of signaling data. Similar to the demand of confidentiality, the SN carries the field of the AMF in the $AUTH_{MME}$ to meet the feature of integrity algorithm agreement. The random numbers and the identities collocate with a group key to make the feature of integrity key agreement. All the user data will be verified with the subsequent key drived from KGK that the ME and the SN agree with in each time session. In addition, the original authentication of signaling data will be protected with the message authentication code (MAC).

5.1.4. Secure key derivation

In our SE-AKA protocol, the KGK is computed by an ME and its SN respectively. In addition, our protocol uses ECDH to generate $KGK_{ME_{Gi-j}}$ without involving key_{Gi-j}. Furthermore, all dedicated keys among entities will be derived from KGK on either peer side directly, without being transmitted over any communication channels. Therefore, the KGK and all dedicated keys are prevented from being disclosed, attacked, or intercepted by adversaries.

The security properties provided by the proposed SE-AKA are as follows.

5.1.5. Enhanced privacy-preservation

To ensure user privacy, the permanent identity of an ME like IMSI should be confidentiality protected. It should never be transmitted without protection. In EPS-AKA, a GUTI (ME's temporary identity) is transmitted instead of the IMSI for identity presentation. In spite of this security arrangement, there are occasions when the IMSI may be transmitted in plain text. We discuss two typical cases. (1) When the network cannot know the ME's temporary identity, it will require the ME's permanent identity. Thus, if the ME's permanent identity is transmitted in plain text, adversary can get it and launch the attacks related identity; and (2) In the case of MAC verification failure, the MAC failure message (Mac_Fail) contained $(Fail, PID_{ME_{Gi-j}}, rand)$ will be sent to network to require a new MAC verification procedure. Therefore, ME's permanent identity also may be leak. These two cases can expose user privacy. In this paper, a lightweight public key infrastructure (PKI) [10] is adopted to provide each HN with a private/public key pair (Pub_{HN},Pri_{HN}). When the network requires the ME's permanent identity $PID_{ME_{Gi-j}}$, such as Section 4.2-(a3) and 4.2-(a7), we can send $PID_{ME_{Gi-j}}$ encrypted by Pub_{HN} to the network instead of sending $PID_{ME_{Gi-j}}$ in plain text.

5.1.6. Key forward/backward secrecy (KFS/KBS)

To provide KFS and KBS between the ME and the SN, our protocol uses ECDH. While generating $KGK_{ME_{Gi-j}}$, our protocol uses aP and bPthat are not related with key_{Gi-j}. Therefore, if disclosure of key_{Gi-j} occurs, attackers cannot guess $KGK_{ME_{Gi-j}}$. In other words, guessing $KGK_{ME_{Gi-j}}$ is a computationally difficult problem.

5.1.7. Perfect forward/backward secrecy (PFS/PBS)

To provide backward and forward secrecy (PFS/PBS), the Section 4.4 gives the details of the method.

PFS guarantees that when a new ME wants to join the group, an access control of the group is necessary, and it needs to perform a full AKA authentication procedure with the SN. The group key must also be updated when the new ME joins a group, so that even if the new ME is able to sniff the old packets of the group, it cannot decrypt them.

PBS guarantees that when an old ME leaves the group, the GMS will revoke the binding relationship between the ME and the group that it belongs to, thus the ME can no longer communicate with the SN as the group member. Moreover, the group key must be updated when the old ME leaves the group, so that the old ME cannot decrypt the new packets of the group after it leaves.

5.2. Resistance to attacks

Besides the security properties mentioned above, our protocol can resist the following attacks.

5.2.1. Replay attack

In our protocol, random number R_{Gi-j} generated by ME_{Gi-j}, R_{HSS} generated by the HSS and R_{MME} generated by MME temporarily use in generating challenge messages toward the opposite side, respectively. Since these random numbers using in each authentication procedure are different, even if an attacker acquires a random number in an authentication procedure, it still cannot fake challenge messages by reusing the random number in a new authentication procedure. Meanwhile, $IV_{ME_{Gi-j}} + i$ generated by the ME can keep both sides involving the authentication synchronized throughout AKA processing. An out-of-sync situation will lead to authentication failure. Therefore, our SE-AKA protocol can prevent replay attacks.

5.2.2. Redirection attack

An adversary initiates a redirection attack by simulating a BS to obtain user information and by impersonating an ME 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 ME 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 ME to establish the connection with the faked BSs. In SE-AKA, the first ME embeds the LAI of the BS in MAC_{G1} and sends MAC_{G1} to the MME in Section 4.2-(a3). The authentication request is rejected if the HSS fails to match the LAI reported by the MME and the LAI embedded in MAC_{G1}. When the remaining MEs want to access the SN, they embed the LAI of the BS in MAC_{G1-j}, and then send MAC_{G1-j} to the MME, since the MME knows the LAI of the BS forwarding $AUTH_{ME_{G1-j}}$, it can verify whether the LAI' forwarded by the BS is the same as that recognized by the ME through checking MAC_{G1-j}. Similarly, the authentication request is rejected if the MME fails to verify MAC_{G1-j}.

5.2.3. Man-in-the-middle (MitM) attack

If a member of a group is able to sniff $AUTH_{G1}$ (Section 4.2-(a3)) and also $AUTH_{MME}$ (Section 4.2-(a6)), it still cannot compute the KGK. Although these messages are sent without protection and the attacker may be able to catch these data, it cannot use them against the network. For instance, in Section 4.2-(a3), the attacker reads the random number of the device (R_{G1-1}) and AMF, while in Section 4.2-(a6), it reads the random number of the MME (R_{MME}) and the random number of HSS (R_{HSS}). With this information and the group key, it is not able to compute a MitM attack, because it cannot get the SV that prestored between MEs and the 3GPP network. SV is securely stored in related entities and not transmitted over insecure communication channels. Meanwhile, the process that generates KGK (Section 4.2-(a7) and (a8)) guarantees that KGK cannot be computed by adversary, even it can get all authentication data transmitted over the communication channels.

5.2.4. DoS attack

During the authentication of the first ME, a malicious ME may launch a DoS attack either to its HSS or to the visited MME.

If the ME forges message in (**a3**), the forged message can be detected by the HSS through checking T_{G1} and comparing LAI contained in MAC_{G1} with LAI received from BS.

During the authentication of the remaining MEs, a malicious ME may launch a DoS attack to the visited MME.

If the ME forges message in (**b3**), the forged message can be detected by the visited MME through checking MAC_{G1-j} containing LAI sent by ME.

Because the proposed SE-AKA is designed for group communication scenario, therefore, this security mechanism can resist the DoS attack launched by multiple devices.

5.2.5. Impersonate attack

In our protocol, all the MEs of a group share a common GTK. If an ME, without loss of generality, supposes that

ME_{G1-1} intends to impersonate another ME in the same group, for example, ME_{G1-j}. ME_{G1-1} may eavesdrop traffic between ME_{G1-j} and the SN, but ME_{G1-1} cannot generate unique R_{G1-j} and $SV_{ME_{G1-j}}$, thus ME_{G1-1} cannot generate a correct $MAC_{ME_{G1-j}}$ to impersonate ME_{G1-j} to perform a successful authentication with the SN. Similarly, ME_{G1-1} cannot get the KGK between ME_{G1-j} and the SN, therefore, it cannot decrypt traffic between ME_{G1-j} and the SN. In summary, the SN can easily distinguish one ME from another even though all MEs use the same GTK. In addition, one ME cannot decrypt traffic between any other ME and the SN.

5.3. Formal verification

5.3.1. ProVerif

We will use ProVerif to verify the security of our protocol. ProVerif is a tool for automatically analyzing the security of cryptographic protocols. Cryptographic primitives are modeled as functions, and messages are represented by terms built over an infinite set of names $a, b, c, \ldots$, an infinite set of variables $x, y, z, \ldots$ and a finite set of function symbols $f_1, \ldots, f_n$. Function symbols represent cryptographic primitives that can be applied to messages. The effect of applying function symbols to terms is described by a set of reduction rules. The syntax of ProVerif calculus processes is given by the Table 4 [22]. ProVerif can be run under Windows or Linux/Mac, in this paper, we conduct the experiments with ProVerif running on a 2.30 GHz-processor 4 GB-memory computing machine to test the proposed SE-AKA protocol under Windows.[3]

5.3.2. Specification of our protocol

The primary goal of our proposed protocol is to provide mutual authentication and key agreement services between MEs and the serving network (SN). Moreover, privacy-preservation (anonymity) and key forward/backward secrecy (KFS/KBS) of our protocol are also need to be verified. The ability of our protocol to resist the typical attacks has been discussed in Section 5.2. Thus, the main security goals to be verified are as follows, and their individual specific requirements have been described in Section 5.1.

- *Mutual authentication between MEs and the SN*;
- *Secrecy of* $KGK_{ME_{Gi-j}}$;
- *Privacy-preservation (anonymity)*;
- *Key forward/backward secrecy (KFS/KBS).*

Because the communication between the SN and the HN is secure, and all authentication procedures between MEs and their SN in a group can be considered the same, thus we only need to verify an authentication procedure among them, without loss of generality, between ME_{G1-1} and its SN.

First, we formalize the basic cryptographic primitives used by the SE-AKA protocol as follows. A symmetric encryption and an asymmetric encryption are defined in Tables A.1 and A.2, respectively; the Diffie-Hellman key agreement is given in Table A.3, see Appendix A.

[3] User manual and tutorial can be downloaded in http://prosecco.gforge.inria.fr/personal/bblanche/proverif/manual.pdf.

Table 4
Main process grammar.

P, *Q*::=	Processes
0	Null process
P|*Q*	Parallel composition
!*P*	Replication
new n;P	Name restriction
in(M,x);P	Message input
out(M,N);P	Message output
if M = N then P else Q	Conditional
let M = D in P else Q	Term evaluation
$R(M_1, \ldots, M_k)$	Macrousage

We further model four security goals in this paper:

(1) *Mutual authentication between MEs and the SN*: We declare the events:
- **event** acceptsMMEparam (key), which is used by the MME to record the belief that it has accepted to run the protocol with the ME and the supplied symmetric key.
- **event** termMMEparam (key), which denotes the MME's belief that it has terminated a protocol run with the ME with the symmetric key supplied as the first argument.
- **event** acceptsMEparam (key), which is used by the ME to record the belief that it has accepted to run the protocol with the MME and the supplied symmetric key.
- **event** termMEparam (key), which denotes the ME's belief that it has terminated a protocol run with the MME with the symmetric key supplied as the first argument.

Next, we use the basic correspondence assertion **event**(**termME**(key)) = = > **event**(**acceptsMME**(key)), and the injective correspondence assertion **inj-event**(**termMME**(key)) = = > **inj-event**(**acceptsME**(key)) to test if SE-AKA can achieve mutual authentication.

(2) *Secrecy of* $KGK_{ME_{Gi-j}}$ *and Key forward/backward secrecy (KFS/KBS)*: We first define a **query** attacker (s), where *s* is session key shared between the ME and the MME. Internally, ProVerif attempts to prove that a state in which the session key *s* is known to the adversary is unreachable (that is, it tests the query **not** attacker, and the query is true when the *s* are *not* derivable by the adversary).

(3) *Privacy-preservation (anonymity)*: Finally, we use observational equivalence, i.e., construct **choice**[PID_{me},TID_{me}] to represent the terms that differ between PID_{me} and TID_{me}. If PID_{me} and TID_{me} are undistinguishable, we say that the SE-AKA satisfies anonymity.

5.3.3. Results of analysis

The verification results are shown in Figs. 4–6. Fig. 4 shows that **RESULT event (termME (*x*_25)) ==> event (acceptsMME (*x*_25)) is true** and **RESULT inj-event (termMME (*x*_1957)) == > inj-event (acceptsME (*x*_1957)) is true**. We can conclude that there has been a successful mutual authentication between ME and its SN. Fig. 5 shows that

Fig. 4. Verification result of mutual authentication between ME and its SN.

RESULT not attacker (s [ ]) is true. It manifests that secrecy of KGK_{Gi-j} and FKS/FBS are hold. Fig. 6 shows that **Observational equivalence is true** ⟨**bad not derivable**⟩. It indicates that the anonymity of our protocol is hold, because adversary cannot get ME's *PID* from the communication.

5.4. Comparison

Table 5 lists the security properties among the 3GPP AKA protocols. We have demonstrated that our protocol can provide the most comprehensive security performance compared to the other AKA protocols.

6. Performance evaluation

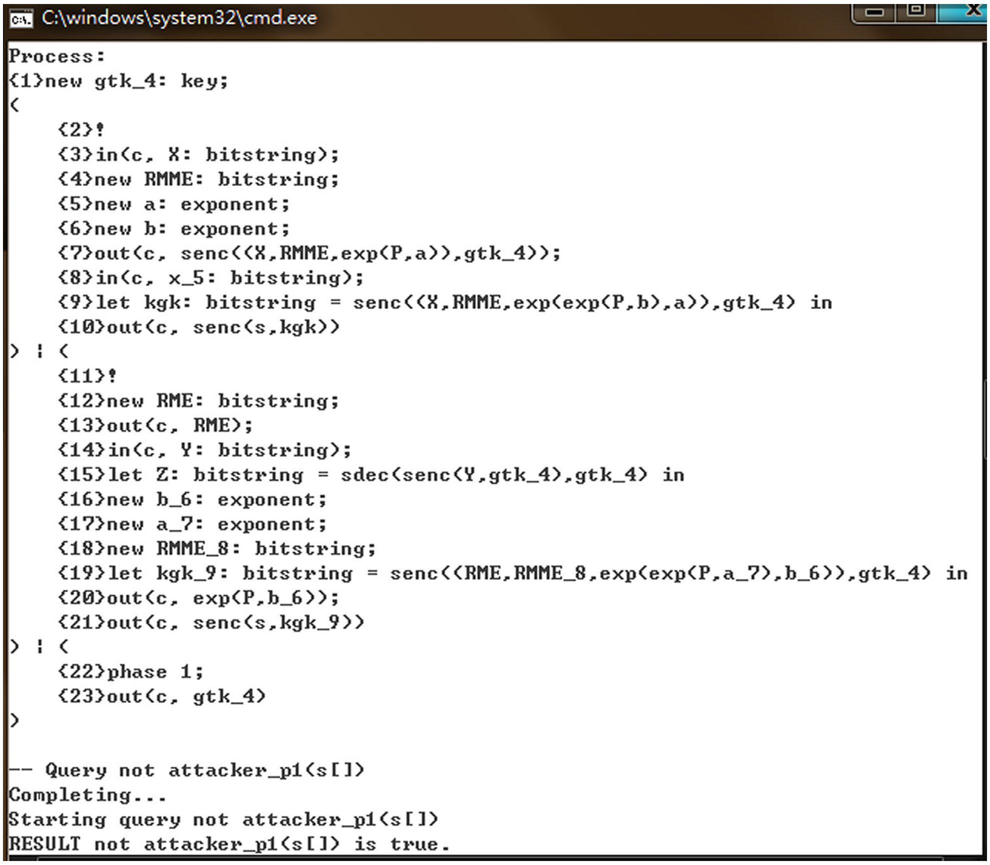
In this section, we compare our SE-AKA protocol with the existing traditional protocols in terms of bandwidth consumption, authentication transmission overhead, computational and storage overhead. We have simulated the proposed SE-AKA in MATLAB running on a 2.30 GHz-processor 4 GB-memory computing machine.

6.1. Communication overhead

- *Bandwidth consumption*

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

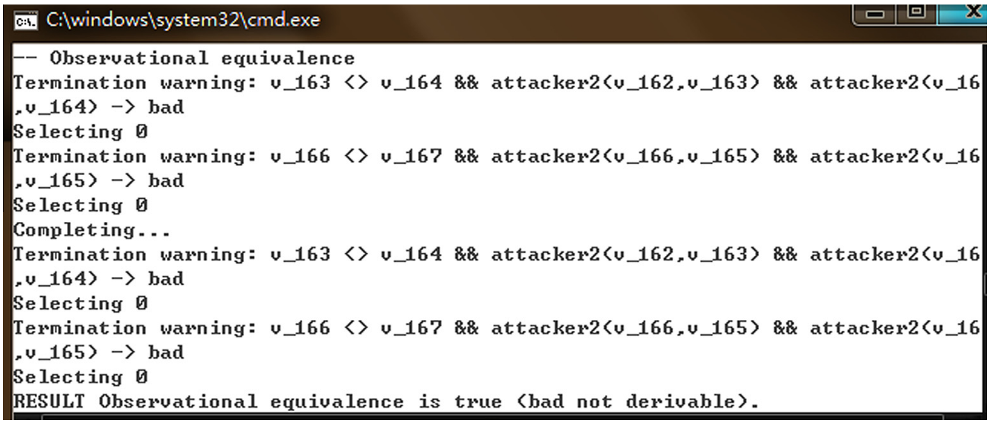
  The bandwidth consumption of AKA protocols are as follows, where bw_{first} represents the bandwidth consumption of the authentication of the first ME. The specific calculation process of (1)–(5) can be found in [30,39], we give the concrete computation procedure of (6) and (7).



```
C:\windows\system32\cmd.exe
Process:
{1}new gtk_4: key;
(
    {2}!
    {3}in(c, X: bitstring);
    {4}new RMME: bitstring;
    {5}new a: exponent;
    {6}new b: exponent;
    {7}out(c, senc((X,RMME,exp(P,a)),gtk_4));
    {8}in(c, x_5: bitstring);
    {9}let kgk: bitstring = senc((X,RMME,exp(exp(P,b),a)),gtk_4) in
    {10}out(c, senc(s,kgk))
) | (
    {11}!
    {12}new RME: bitstring;
    {13}out(c, RME);
    {14}in(c, Y: bitstring);
    {15}let Z: bitstring = sdec(senc(Y,gtk_4),gtk_4) in
    {16}new b_6: exponent;
    {17}new a_7: exponent;
    {18}new RMME_8: bitstring;
    {19}let kgk_9: bitstring = senc((RME,RMME_8,exp(exp(P,a_7),b_6)),gtk_4) in
    {20}out(c, exp(P,b_6));
    {21}out(c, senc(s,kgk_9))
) | (
    {22}phase 1;
    {23}out(c, gtk_4)
)

-- Query not attacker_p1(s[])
Completing...
Starting query not attacker_p1(s[])
RESULT not attacker_p1(s[]) is true.
```

Fig. 5. Verification result of secrecy of KGK and KFS/KBS.



```
C:\windows\system32\cmd.exe
-- Observational equivalence
Termination warning: v_163 <> v_164 && attacker2(v_162,v_163) && attacker2(v_16
,v_164) -> bad
Selecting 0
Termination warning: v_166 <> v_167 && attacker2(v_166,v_165) && attacker2(v_16
,v_165) -> bad
Selecting 0
Completing...
Termination warning: v_163 <> v_164 && attacker2(v_162,v_163) && attacker2(v_16
,v_164) -> bad
Selecting 0
Termination warning: v_166 <> v_167 && attacker2(v_166,v_165) && attacker2(v_16
,v_165) -> bad
Selecting 0
RESULT Observational equivalence is true (bad not derivable).
```

Fig. 6. Verification result of privacy-preservation.

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

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

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

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

$$bw_{first} = \sum_{i=1}^{5} |Message_i| = 1312 \quad bits. \tag{2}$$

Table 5
Comparisons of security properties among the 3GPP AKA protocols.

	SE-AKA	UMTS-AKA [2]	EPS-AKA [4]	AP-AKA [24]
Type of cryptosystem	Symmetric and ECDH	Symmetric	Symmetric	Symmetric
Secure against redirection attack	Yes	No	No	Yes
Secure against man-in-the middle attack	Yes	No	No	Yes
Secure against DoS attack	Yes	No	No	No
KFS/KBS	Yes	No	No	No
Privacy-preservation	Enhanced	General	General	General
Support group authentication	Yes	No	No	No
	X-AKA [26]	Cocktail-AKA [28]	S-AKA [30]	G-AKA [31]
Type of cryptosystem	Symmetric	Symmetric	Symmetric	Symmetric
Secure against redirection attack	No	No	Yes	No
Secure against man-in-the middle attack	No	No	Yes	No
Secure against DoS attack	Partial	No	Partial	No
KFS/KBS	No	No	No	No
Privacy-preservation	No	General	No	No
Support group authentication	No	No	No	Yes

Table 6
Setting of parameters.

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

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

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

$$bw_{first} = \sum_{i=1}^{6} |Message_i| = 1250 + 544x \quad bits. \tag{3}$$

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

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

$$bw_{first} = \sum_{i=1}^{5} |Message_i| = 1220 \quad bits. \tag{4}$$

The overall bandwidth consumption for n devices is calculated as $n * 1220$.

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

$$bw_{first} = \sum_{i=1}^{3} |Message_i| = 640 + 560x \quad bits. \tag{5}$$

The overall bandwidth consumption for n devices is calculated as $n*(640 + 560x)$.

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

$$bw_{first} = \sum_{i=1}^{5} |Message_i| = 1888 \quad bits. \tag{6}$$

- $Message_1 = 2|ID| + |RN| + |MAC| = 448 \quad bits.$
- $Message_2 = Message_1.$
- $Message_3 = 2|RN| + |AMF| + |GTK| = 432 \quad bits.$
- $Message_4 = |AMF| + 3|RN| + |MAC| = 496 \quad bits.$
- $Message_5 = |MAC| = 64 \quad bits.$

$$bw_{remaining} = \sum_{i=1}^{2} | Message_i | = 880 \quad bits. \tag{7}$$

where $bw_{remaining}$ represents the bandwidth consumption of authentication of each remaining ME.

- $Message_1 = 2|ID| + |RN| = 320 \quad bits.$
- $Message_2 = |AMF| + 3|RN| + |MAC| = 496 \quad bits.$
- $Message_3 = |MAC| = 64 \quad bits.$

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

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

$$bw_{first} = \sum_{i=1}^{5} | Message_i | = 2184 \quad bits. \tag{8}$$

- $Message_1 = |ID| + |RN| + |MAC| = 320 \quad bits.$
- $Message_2 = |Message_1| + |LAI| = 360 \quad bits$
- $Message_3 = 2|RN| + |AMF| + |GTK| = 432 \quad bits$
- $Message_4 = |ID| + |MAC| + 3|RN| + |AMF| + |ECDH\ key| = 816 \quad bits$
- $Message_5 = |MAC| + |ECDH\ key| = 256 \quad bits$

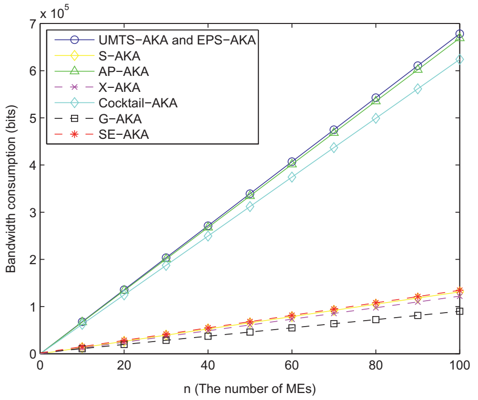
$$bw_{remaining} = \sum_{i=1}^{3} |Message_i| = 1328 \quad bits. \tag{9}$$

where $bw_{remaining}$ represents the bandwidth consumption of authentication of each remaining ME.

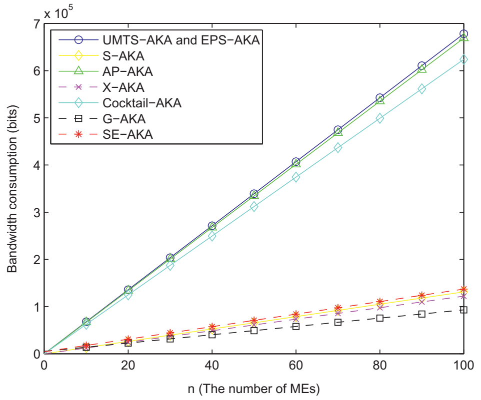
- $Message_1 = |ID| + |RN| = 256 \quad bits.$
- $Message_2 = |ID| + |MAC| + 3|RN| + |AMF| + |ECDH\ key| = 816 \quad bits$
- $Message_3 = |MAC| + |ECDH\ key| = 256 \quad bits$

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

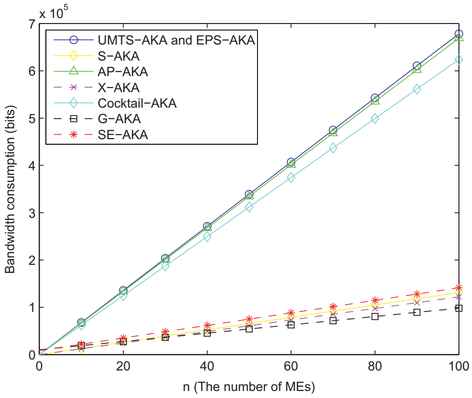
Fig. 7(a)–(f) show the bandwidth consumption of several AKA protocols, when the number of the MEs is



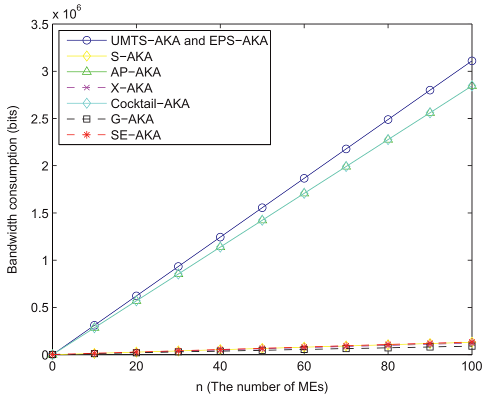
(a) *x* (the number of AVs)=10, *m* (the number of groups)=2



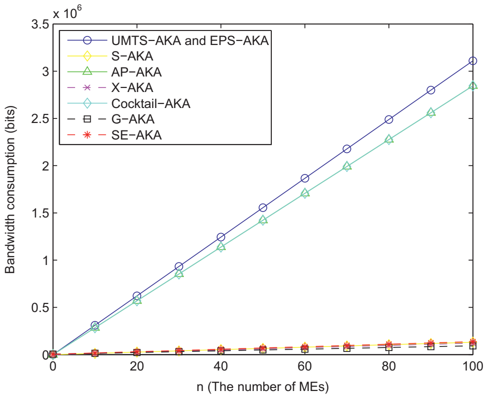
(b) *x* (the number of AVs)=10, *m* (the number of groups)=5



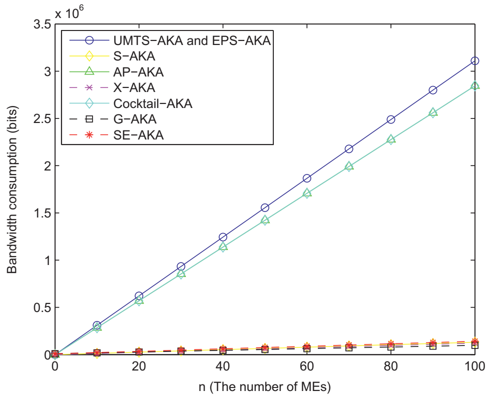
(c) *x* (the number of AVs)=10, *m* (the number of groups)=10



(d) *x* (the number of AVs)=50, *m* (the number of groups)=2



(e) *x* (the number of AVs)=50, *m* (the number of groups)=5



(f) *x* (the number of AVs)=50, *m* (the number of groups)=10

Fig. 7. Comparison of the bandwidth consumption.

Table 7
Comparison of the authentication transmission overhead.

Reference schemes	Authentication transmission overhead
SE-AKA	$m(6\alpha + 2\beta) + (n\text{-}m)(6\alpha) = 6n\alpha + 2m\beta$
UMTS-AKA [2]	$6n\alpha + 2n\beta$
EPS-AKA [4]	$6n\alpha + 2n\beta$
AP-AKA [24]	$5n\alpha + 2n\beta$
X-AKA [26]	$5n\alpha + 2n\beta$
Cocktail-AKA [26]	$4n\alpha + 2n\beta$
S-AKA [30]	$7n\alpha + 2n\beta$
G-AKA [31]	$m(7\alpha + 2\beta) + (n\text{-}m)(7\alpha) = 7n\alpha + 2m\beta$

different. Despite that SE-AKA is not the protocol that saves the most bandwidth, it can provide more security. The reason is that we use asymmetric cryptosystem to enhance the security, but the traditional protocols only use symmetric cryptosystem to achieve authentication in UMTS or LTE networks. Indeed, they cannot provide good security. From Table 5, the security of several protocols is weak, like X-AKA and G-AKA, they can barely resist the existing attacks. In fact, we need a hybrid cryptosystem to design authentication protocol in UMTS/LTE networks. On one hand, this can provide a higher security; on the other hand, the effectiveness of communication can also be guaranteed. As a matter of fact, even though our protocol adopts asymmetric cryptosystem, the bandwidth consumption of the protocol still does not increase rapidly. From Fig. 7(a) - (f), we can see that the bandwidth consumption of our protocol is close to that of S-AKA, G-AKA and X-AKA, and far better than that of UMTS-AKA, EPS-AKA and AP-AKA. Most importantly, our protocol can provide much better security compared to the other protocols.

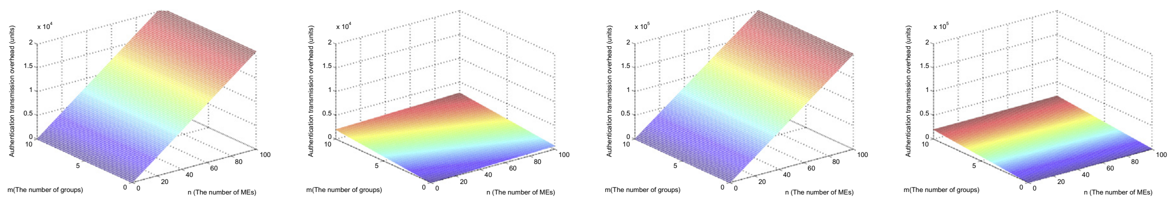
- *Authentication transmission overhead*

Let the overhead of authentication message delivery between the ME and the MME be α unit, and between the MME and the HSS be β unit, respectively. Since the MME locates the SN which is far away from the HSS, $\beta \gg \alpha$. We also assume that there are n MEs forming m groups, obviously, $n > m$. We compare the overhead of authentication message delivery of SE-AKA with that of traditional protocols as shown in Table 7.

From Table 7, we can find that the authentication transmission overhead of the existing AKA protocols are similar; therefore, we set average authentication transmission overhead of all existing AKA protocols as $5n\alpha + 2n\beta$. According to Table 7, we draw Fig. 8 when $\alpha = 1$, $\beta = 100$ and $\alpha = 1$, $\beta = 1000$, respectively. As shown in Fig. 8, we can see that the overhead of authentication message delivery of SE-AKA (Fig. 8(b) and (d)) is lower than other existing AKA protocols (Fig. 8(a) and (c)). Therefore, our protocol owns the lowest authentication transmission overhead.

6.2. Computational overhead

The time used for the primitive cryptography operations has been measured by using C/C++ OPENSSL library [40] tested on an Celeron 1.1 GHz processor as an UE and

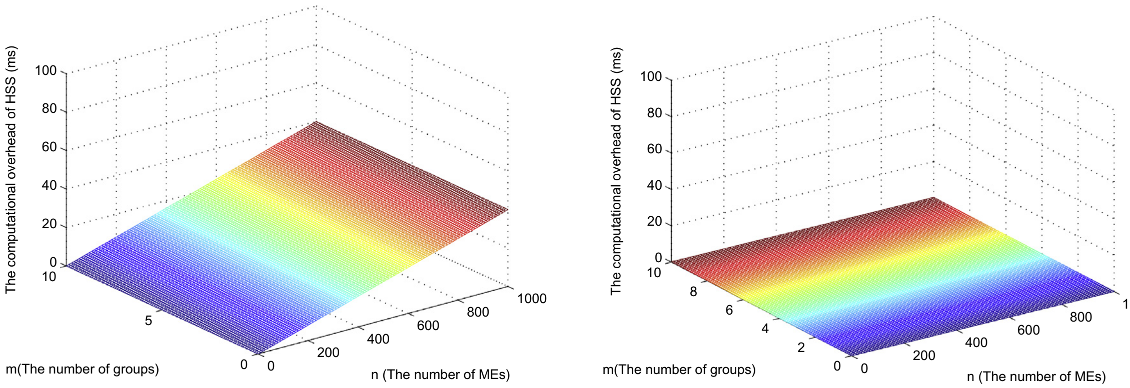


(a) The existing AKA protocols without group authentication mechanism ($\alpha = 1$, $\beta = 100$)
(b) SE-AKA ($\alpha = 1$, $\beta = 100$)
(c) The existing AKA protocols without group authentication mechanism ($\alpha = 1$, $\beta = 1000$)
(d) SE-AKA ($\alpha = 1$, $\beta = 1000$)

Fig. 8. Comparison of the authentication transmission overhead.

Table 8
Comparisons of computational overhead of each entity among the 3GPP AKA protocols.

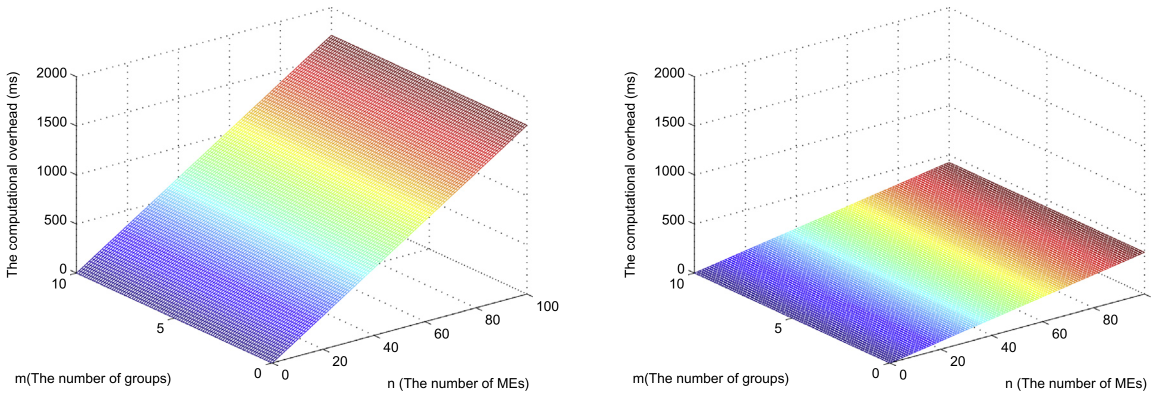
ms	SE-AKA	UMTS-AKA [2]	EPS-AKA [4]	AP-AKA [24]
The first ME	$4T_H + 2T_{PM} = 3.2964$	$5T_H = 0.178$	0.178	$3T_H = 0.1068$
Remaining MEs	$(n-1)(3T_H + 2T_{PM}) = 3.2608(n-1)$	$0.178(n-1)$	$0.178(n-1)$	$0.0168(n-1)$
MME	$n(3T_H + 2T_{PM}) = 0.9863n$	0	0	0
HSS	$m(2T_H) = 0.0242m$	$n(5T_H) = 0.0605n$	$0.0605n$	$n(4T_H) = 0.048n$
Total	$4.2471n + 0.0242m$	$0.2385n$	$0.2385n$	$0.1548n$
	X-AKA [26]	Cocktail-AKA [28]	S-AKA [30]	G-AKA [31]
The first ME	$5T_H = 0.178$	$8T_H = 0.2848$	$6T_H = 0.2136$	$4T_H = 0.1424$
Remaining MEs	$0.178(n-1)$	$0.2848(n-1)$	$0.2136(n-1)$	$0.1424(n-1)$
MME	0	$nT_H = 0.0121n$	$n(2T_H) = 0.0242n$	$n(3T_H) = 0.0363n$
HSS	$n(5T_H) = 0.0605n$	$n(5T_H) = 0.0605n$	$n(2T_H) = 0.0242n$	$m(2T_H) = 0.0242m$
Total	$0.2385n$	$0.3574n$	$0.262n$	$0.1787n + 0.0242m$



(a) The existing AKA protocols without group authentication mechanism

(b) SE-AKA

Fig. 9. Comparison of the computational overhead of HSS.



(a) Cao's scheme

(b) SE-AKA

Fig. 10. Comparison of the total computation overhead between scheme [21] and SE-AKA.

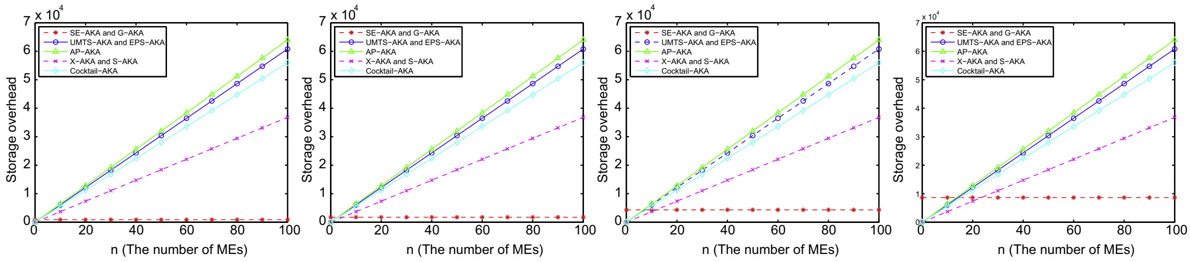
Dual-Core 2.6 GHz as an MME and an HSS in reference [41]: $Time_H^{ME} = 0.0356$ ms, $T_H^{MME} = T_H^{HSS} = 0.0121$ ms. $T_{PM}^{ME} = 1.537$ ms, $T_{PM}^{MME} = T_{PM}^{HSS} = 0.475$ ms. T_H and T_{PM} represent time cost of hash and time cost of point multiplication, respectively. Moreover, n represents the number of MEs, m stands for the number of groups.

Comparisons of computational overhead of each entity among the 3GPP AKA protocols are shown in Table 8. From Table 8, we can find that the computational overhead of the existing AKA protocols are similar; therefore, we first set average computational overhead of HSS in all existing AKA protocols as $0.04n$. According to Table 8, we plot the computational overhead of HSS in terms of ME numbers n and group numbers m, as shown in Fig. 9. It can be seen that the proposed SE-AKA protocol always achieves lower computational overhead of HSS compared to other existing AKA protocols. Therefore, the computational overhead of HSS in our SE-AKA is the lowest in all protocols. This is because SE-AKA shifts some computational overhead in the HSS to the MME. This can make the HSS and the MME bear computational overhead together and reduce the overload of the HSS to some extent.

Furthermore, the computational overhead of all entities in SE-AKA are lager than that of other traditional protocols

Table 9
Comparison of storage overhead in the SN.

Reference schemes	Storage overhead (bits)
SE-AKA	432m
UMTS-AKA [2]	608n
EPS-AKA [4]	608n
AP-AKA [24]	640n
X-AKA [26]	368n
Cocktail-AKA [26]	560n
S-AKA [30]	368n
G-AKA [31]	432m



(a) m=2 (b) m=4 (c) m=10 (d) m=20

Fig. 11. Comparison of the storage overhead.

except for the HSS. This is because ECDH is adopted to solve KFS/KBS in SE-AKA, while other traditional protocols only use symmetric cryptography. Despite SE-AKA is not the protocol that has the lowest computational overhead, compared with the scheme that is completely based on asymmetric cryptosystem, e.g., scheme [21], it costs about $17.2n + 57.3$ ms, while the SE-AKA costs $4.2471n + 0.0242$ m ms. To compare the total computation overhead between scheme [21] and the SE-AKA, we plot the total computation overhead in terms of ME numbers n and group numbers m, as shown in Fig. 10. It can be seen that the proposed SE-AKA protocol achieves lower total computation overhead compared to scheme [21]. Therefore, the proposed SE-AKA can provide good security with acceptable computation overhead.

6.3. Storage overhead

In this section, we analyze the storage overhead of several AKA protocols. In addition, we consider that there are n MEs that are formed into m groups, $n > m$.

For UMTS-AKA and EPS-AKA, each ME requires its SN to store a set of authentication vectors (AVs), the length of AV is 608 bits, therefore occupied storage space for authenticating n MEs is 608n bits. For X-AKA, $n \times (Temporal\ Key\ (TK) + AUTH)$ bits space is occupied, where $|TK| = 128$ bits and $|AUTH| = 240$ bits. As to S-AKA, it will occupy $n \times ((DK) + AUTN)$ bits storage space, where $|DK| = 128$ bits and $|AUTN| = 240$ bits. For AP-AKA, each ME requires its SN to store a set of authentication vectors (AVs), the length of AV is 640 bits, therefore occupied storage space for authenticating n ME is 640n bits. For Cocktail-AKA, it will occupy $n \times |PAV|$ bits storage space, where $|PAV| = 560$ bits. G-AKA utilizes group authentication data, instead of maintaining each ME's authentication information, thus for m groups of MEs, the SN only uses $m \times \ (GTK + RN_H + RN_{M1-1} + AMF + Index\ \ table\ \ entry)$ bits storage space, where $|GTK| = 128$ bits, $|RN_H| = |RN_{M1-1}| = 128$ bits and $|AMF| = 48$ bits, $|Index\ table\ entry|$ can be negligible. While SE-AKA also utilizes group authentication data, instead of maintaining each ME authentication information, thus its storage overhead is basically the same as G-AKA's. Comparison of storage overhead on several AKA protocols is presented in Table 9.

Fig. 11(a)–(d) compare the storage overhand of several AKA protocols, varying with the number of MEs. From the figures, we can see both the SE-AKA and G-AKA protocols have smaller storage costs than others. The reason that the storage overhead of the SE-AKA protocol does not change with n is that SE-AKA shifts the impact of the number of MEs to the impact of that of the number of ME groups.

7. Conclusion and future work

In this paper, we have proposed a secure and efficient AKA protocol SE-AKA to fit in the LTE networks with all of the group authentication scenarios. Compared with other authentication protocols, SE-AKA cannot only provide strong security properties including privacy-preservation and KFS/KBS, but also provide a group authentication mechanism which can effectively authenticate group devices. Extensive security analysis and formal verification by using ***proverif*** have shown that the proposed SE-AKA is secure against various malicious attacks. The elaborate performance evaluations in terms of communication, computational and storage overhead have been conducted, which demonstrate that the transmission overhead of the whole authentication is considerably reduced, the computational overhead of the HSS and the storage overhead in the serving network can also be decreased, and the bandwidth consumption is close to that of S-AKA, G-AKA and X-AKA, and far better than that of UMTS-AKA, EPS-AKA and AP-AKA.

In the group-based communication, group devices will face new challenges in authentication when they are moving. A long delay and large computational overhead may occur during handover or roaming. Therefore, the security research of group-based communication in the duration of handover or roaming will be further exploited in our future work.

Acknowledgments

This work is supported by China Scholarship Council, the National Natural Science Foundation of China Grant 61272457, 61102056, and the National Science and Technology Major Projects (No. 2012ZX03002003).

Appendix A

The basic cryptographic primitives used by the SE-AKA protocol are formalized as follows (see Tables A.1, A.2 and A.3).

Table A.1
Symmetric encryption.

1. **Type** key.
2. **Fun** senc (bitstring, key): bitstring.
3. **Reduc forall** m: bitstring, k: key; sdec (senc (m, k), k) = m.

Table A.2
Asymmetric encryption.

1. **Type** skey.
2. **Type** pkey.
3. **Fun** pk (skey): pkey.
4. **Fun** aenc (bitstring, pkey): bitstring.
5. **Reduc forall** m: bitstring, sk: skey; adec (aenc (m, pk (sk)), sk) = m.

Table A.3
Diffie–Hellman key agreement.

1. **Type** G.
2. **Type** exponent.
3. **Const** g: G [**data**].
4. **Fun** exp (G, exponent): G.
5. **Equation forall** x: exponent, y: exponent; exp (exp (g, x), y) = exp (exp (g, y), x).

References

[1] J.A. Audestad, Network aspects of the GSM system, EUROCON 88 (1988).
[2] 3GPP TS 21.133 V4.1.0, 3G Security; Security Threats and Requirements, 2001.
[3] http://www.3gpp.org/LTE.
[4] 3GPP TS 33.401 V12.5.0, 3GPP System Architecture Evolution (SAE); Security architecture, September 2012.
[5] M. Zhang, Provably-secure enhancement on 3GPP authentication and key agreement protocol, Verizon Commun., Cryptology ePrint Archive Rep. 2003/092, 2003.
[6] U. Meyer, S. Wetzel, A man-in-the-middle attack on UMTS, in: Proc. 3rd ACM WiSe, New York, 2004, pp. 90–97.
[7] H.J. Zhu, X.D. Lin, M.H. Shi, P.H. Ho, X.M. Shen, PPAB: a privacy-preserving authentication and billing architecture for metropolitan area sharing networks, IEEE Transactions on Vehicular Technology 58 (5) (2009) 2529–2543.
[8] X.H. Liang, X. Li, R.X. Lu, X.D. Lin, X.M. Shen, Enabling pervasive healthcare with privacy preservation in smart community, in: 2012 IEEE International Conference on Communications (ICC), 10–15 June 2012, pp. 3451–3455.
[9] X.H. Liang, R.X. Lu, L. Chen, X.D. Lin, X.M. Shen, PEC: a privacy-preserving emergency call scheme for mobile healthcare social networks, Journal of Communications and Networks 13 (2) (2011) 102–112.
[10] M. Arapinis, L. Mancini, E. Ritter, M. Ryan, N. Golde, K. Redon, R. Borgaonkar, New privacy issues in mobile telephony: fix and verification, in: Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS '12), ACM, New York, NY, USA, 2012, pp. 205–216.
[11] 3GPP TR 23.888 V11.0.0, System Improvements for Machine-Type Communications, September 2012.
[12] R.X. Lu, X. Li, X.H. Liang, X.M. Shen, X.D. Lin, GRS: the green, reliability, and security of emerging machine to machine communications, IEEE Communications Magazine 49 (4) (2011) 28–35.
[13] C.Z. Lai, H. Li, Y. Y Zhang, J. Cao, Security issues on machine to machine communications, KSII Transaction on Internet and Information Systems 6 (2) (2012) 498–514.
[14] A. Wasef, X.M. Shen, PPGCV: privacy preserving group communications protocol for vehicular ad hoc networks, in: IEEE International Conference on Communications, 2008. ICC '08, 19–23 May 2008, pp. 1458–1463.
[15] D. Niyato, L. Xiao, P. Wang, Machine-to-machine communications for home energy management system in smart grid, IEEE Communications Magazine 49 (4) (2011) 53–59.
[16] Y. Zhang, R. Yu, M. Nekovee, Y. Liu, S.l. Xie, S. Gjessing, Cognitive machine-to-machine communications: visions and potentials for the smart grid, IEEE Network 26 (3) (2012) 6–13.
[17] K. Lee, J.S. Shin, Y.W. Cho, K.S. Ko, D.K. Sung, H.S. Shin, A group-based communication scheme based on the location information of MTC devices in cellular networks, in: Communications (ICC), 2012 IEEE International Conference on, 10–15 June 2012. pp. 4899–4903.
[18] H.H. Ngo, X.P. Wu, P.D. Le, B. Srinivasan, An individual and group authentication model for wireless network services, JCIT: Journal of Convergence Information Technology 5 (1) (2010) 82–94.
[19] N. Aboudagga, J.J. Quisquater, M. Eltoweissy, Group authentication protocol for mobile networks, in: WIMOB '07 Proceedings of the Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications IEEE Computer Society Washington, DC, USA, 2007.
[20] A. Fu, S. Lan, B. Huang, Z. Zhu, Y. Zhang, A novel group-based handover authentication scheme with privacy preservation for mobile WiMAX networks, IEEE Communications Letters 16 (11) (2012) 1744–1747.
[21] J. Cao, M. Ma, H. Li, A group-based authentication and key agreement for MTC in LTE networks, in: Proc. of IEEE Globecom, 2012.
[22] B. Blanchet, Proverif: Cryptographic Protocol Verifier in the Formal Model. <http://www.proverif.ens.fr/>.
[23] L. Harn, W.J. Hsin, On the security of wireless network access with enhancements, in: Proceedings of the 2003 ACM Workshop on Wireless Security, San Diego, CA, USA, pp. 88–95.
[24] M.X. Zhang, Y.G. Fang, Security analysis and enhancements of 3GPP authentication and key agreement protocol, IEEE Transactions on Wireless Communications 4 (2) (2005) 734–742.
[25] C.C. Lee, C.L. Chen, H.H. Ou, L.A. Chen, Extension of an efficient 3GPP authentication and key agreement protocol, Wireless Personal Communications (2011) 1–12.
[26] C.M. Huang, J.W. Li, Authentication and key agreement protocol for UMTS with low bandwidth consumption, in: Proceedings of 19th IEEE international conference on advanced information networking and applications (AINA), 2005, pp. 392–397.
[27] J. Al-Saraireh, S. Yousef, A new authentication protocol for UMTS mobile networks, EURASIP Journal of Wireless Communications and Networking 2006 (2) (2006).
[28] H.H. Ou, M.S. Hwang, J.K. Jan, A cocktail protocol with the authentication and key agreement on the UMTS, Journal of Systems and Software 83 (2) (2010) 316–325.
[29] S. Wu, Y. Zhu, Q. Pu, Security analysis of a cocktail protocol with the authentication and key agreement on the UMTS, Communications Letters 14 (4) (2010) 366–368.
[30] Y.L. Huang, C.Y. Shen, S.W. Shieh, S-AKA: a provable and secure authentication key agreement protocol for UMTS networks, IEEE Transactions on Vehicular Technology 60 (9) (2011) 4509–4519.
[31] Y.W. Chen, J.T. Wang, K.H. Chi, C.C. Tseng, Group-based authentication and key agreement, Wireless Personal Communications (2010).
[32] NIST, Special Publication 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, 2006.
[33] H. Krawczyk. SIGMA: the'SIGn-and-MAc' Approach Authnticated Diffie-Hellman and its Use in the IKE Protocols, in: Advances in Cryptology – CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA. vol. 2729, August 17–21, 2003, pp. 400–425.
[34] L. Lee, D. Kim, B. Chung, H. Yoon, Adaptive hysteresis using mobility correlation for fast handover, IEEE Communications Letters 12 (2) (2008) 152–154.
[35] H.H. Choi, J.B. Lim, H. Hwang, K. Jang, Optimal handover decision algorithm for throughput enhancement in cooperative cellular networks, in: Proc. 2010 IEEE VTC – Fall, pp. 1–5.
[36] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, J. Arkko, Diameter Base Protocol. <http://www.ietf.org/rfc/rfc3588.txt>.
[37] M.J. Moyer, J.R. Rao, P. Rohatgi, A survey of security issues in multicast communications, IEEE Network 13 (6) (1999) 12–23.

[38] S. Rafaeli, D. Hutchison, A survey of key management for secure group communication, ACM Computing Surveys 35 (3) (2003) 309–329.
[39] C.Z. Lai, H. Li, X.Q. Li, J. Cao, A novel group access authentication and key agreement protocol for machine-type communication, Transactions on Emerging Telecommunications Technologies (2013).
[40] OPENSSL, <http://www.openssl.org/>.
[41] J. Cao, H. Li, M.D. Ma, Y.Y. Zhang, C.Z. Lai, A simple and robust handover authentication between HeNB and eNB in LTE networks, Computer Networks 56 (8) (2012) 2119–2131.

Chengzhe Lai received the B.S. degree from Xi'an Institute of Posts and Telecommunications. He is currently working toward the Ph.D. degree in Cryptography, Xidian University, China. He is currently a visiting Ph.D. student with the Broadband Communications Research (BBCR) Group, University of Waterloo. His research interests include wireless network security, LTE networks and M2M communication security.

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.

Rongxing Lu received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China in 2006 and the 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 Division of Communication Engineering, School of Electrical and Electronics Engineering, Nanyang Technological University. His research interests include wireless network security, applied cryptography, and trusted computing.

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.