Wireless Pers Commun (2012) 62:965–979
DOI 10.1007/s11277-010-0104-7

Group-Based Authentication and Key Agreement

Yu-Wen Chen · Jui-Tang Wang · Kuang-Hui Chi · Chien-Chao Tseng

Published online: 1 September 2010
© Springer Science+Business Media, LLC. 2010

Abstract This paper presents an authentication and key agreement protocol to streamline communication activities for a group of mobile stations (MSs) roaming from the same home network (HN) to a serving network (SN). In such a roaming scenario, conventional schemes require the SN to interact with the HN for authenticating respective MSs, at the cost of repeated message exchanges and communication delay. Instead, in our design, when the first MS of a group visits, the SN performs full authentication with the concerned HN and thereby obtains authentication information for the MS and other members. Thus when any other MS of the same group visits, the SN can authenticate locally without subsequent involvement of the HN, so as to simplify protocol operations. We will show that our scheme does not trade performance for security and robustness to the extent that security requirements are unduly weakened. Both qualitative and quantitative discussions indicate that our proposed scheme lends itself to pragmatic settings.

Keywords Wireless network · Security · Group key · Authentication and key agreement · Roaming

This work has been supported by the National Science Council, ROC, under grants NSC 98-2220-E-009-047 and NSC 97-2221-E-009-051-MY3, and by the ministry of Economics, ROC, under the grant 9301 × S2210.

Y.-W. Chen · C.-C. Tseng (✉)
Institute of Computer Science and Engineering, National Chiao Tung University, Hsinchu, Taiwan
e-mail: cctseng@cs.nctu.edu.tw

J.-T. Wang
Information and Computer Laboratories, Industrial Technology Research Institute, Hsinchu, Taiwan

K.-H. Chi
Department of Electrical Engineering, National Yunlin University of Science and Technology, Yunlin, Taiwan

1 Introduction

Concerning wireless network security, authentication is one of the first measures that must be taken to validate users or the system. There has been active research on developing robust security mechanisms for authentication and cryptographic key management. Several authentication and key agreement (AKA) protocols like EAP-AKA [3], EAP-SIM [6], and UMTS AKA [1] are well-known examples that withstand various attacks. Other variants of AKA or applications can also be found in [4,8–11]. Though effective, current protocols incur nontrivial communication delay due to potentially prohibitive message exchanges between different network domains.

We note that mobile stations (MSs) belonging to the same Home Network (HN) may form a group that is likely to migrate somewhere together. That is, MSs of an HN may visit the same network and move along the same route, e.g., a tourist group from the same city or country traveling from one place to another, students having a field trip, or even mobile routers on a public transportation system. Such group-based movement may cause repeated invocations of costly authentication procedures within a short period of time, at the expense of signaling traffic between the serving network (SN) and the HN if a traditional AKA protocol is used to authenticate MSs separately. As shown in Fig. 1, user-perceived authentication delay and system signaling overhead grow with the involvement of more MSs.

In view of group movement behavior, this paper presents an AKA protocol with refined interactions between an SN and the HN. This is accomplished by authorizing the SN to authenticate MSs using a Group Authentication Key (GAK) generated by the HN. Only the first MS visiting the SN is required to undergo full authentication whereby the SN can acquire a Group Temporary Key (GTK) from the HN. With the GTK in place, the SN is enabled to carry out mutual authentication with remaining MSs of the group, if any, without intervention of the remote HN. In this fashion, authentication delay is trimmed as a whole. Meanwhile, the signaling overhead between the HN and the SN is considerably reduced from a factor of the number of MSs to a factor of the number of groups among MSs.

As shall be seen shortly, our proposed scheme is characterized by several strengths. First, our scheme maintains the same security level as in counterpart AKA protocols. Authentication data received by an SN include sufficient information for the SN to distinguish each MS of a group and establish a unique master key to secure a data session (each MS shares a secure communication channel with the SN). Hence, an MS cannot impersonate any other entity of the same group. Additionally, the freshness of authentication messages is ensured throughout

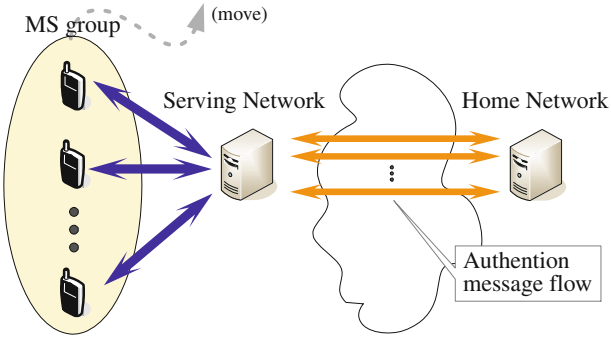

Fig. 1 Group movement causes repeated executions of AKA procedures by the same serving network and home network

because every authentication for an MS requires both a nonce and a correct number counting how many authentications the MS has performed so far. Further, our scheme can speed up handover procedure if the HN distributes authentication data to neighboring SNs the MS is likely to visit next. Moreover, our design is well applicable to any system accommodating a remote authentication sever, such as TETRA network [12], WiMAX network [2], vehicular ad hoc network [13], and security system in enterprises.

The remainder of this paper is organized as follows. Section 2 describes some background on AKA protocols. Section 3 provides the proposed mechanism in full. Next we analyze the security and signaling overhead of our proposal in Sect. 4. Lastly Sect. 5 concludes this study.

2 Background

Here we outline the essence of AKA mechanisms in widespread use by mobile telecommunications networks. For a better exposition, let us take UMTS AKA [1] as a representative to exemplify their design tenet.

UMTS AKA can broadly be divided into two stages: authentication data distribution, and user authentication and key agreement (represented above and below, respectively, the dashed line in Fig. 2). The former enables the HN of an MS to distribute authentication data to the SN the MS is visiting. The latter is to establish a new pairwise session key between the MS and the SN. Overall, UMTS AKA consists of several messages exchanged in following lines.

1. ID Request: Upon detecting an access request by an MS, the SN initiates an authentication procedure by asking the MS for its identity.
2. ID Response: The MS sends its identity to the SN.
3. Authentication Data Request: The SN sends this message to acquire n Authentication Vectors AV$(1 \ldots n)$ from the HN. The operation of generating each attribute in AV is depicted in Fig. 3a. In addition to Authentication Management Field (AMF) and Sequence Number (SQN), a pre-shared key for the MS and a random number RAND are parameters taken to generate AV(i) comprising the Message Authentication Code

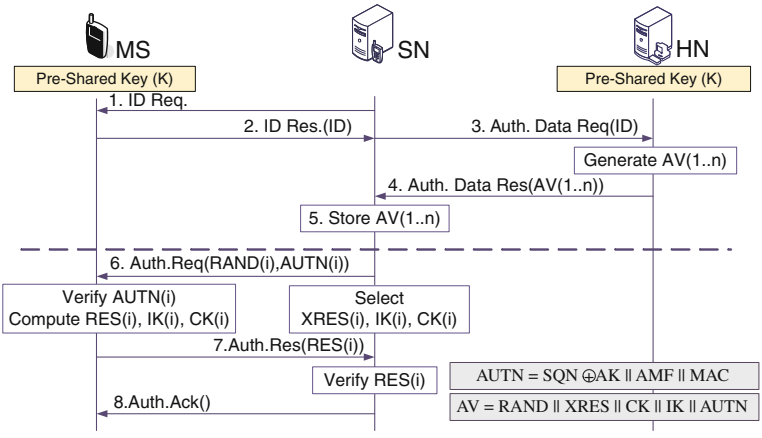

Fig. 2 UMTS AKA message flow. Operators $\oplus$ and $\|$ denote exclusive-or and concatenation, respectively, of bit strings of involved operands

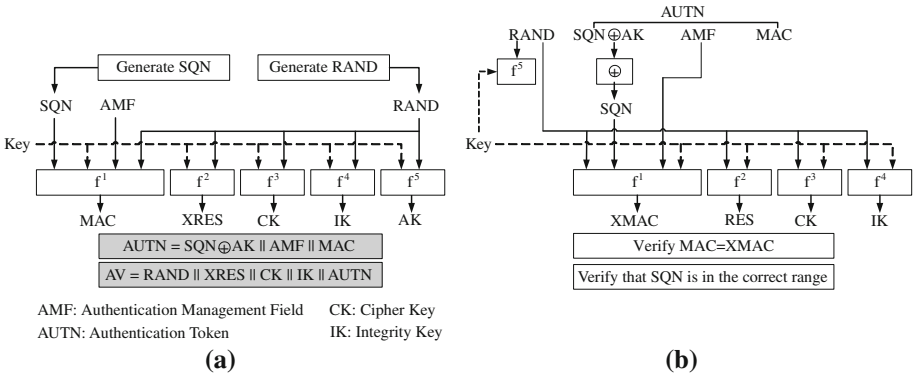

Fig. 3 UMTS AKA operations (with reference to Fig. 2). **a** Generating Authentication Vectors on the HN side, **b** Verifying a Network Authentication Token on the MS side

(MAC), eXpected Response (XRES), cipher key (CK), integrity key (IK), Anonymity Key (AK), and Network Authentication Token (AUTN).

4. Authentication Data Response: The HN sends back the generated AV (for the corresponding MS) so that the SN is authorized to authenticate the requesting MS.
5. User Authentication Request: Upon receipt of a message containing authentication vectors, the SN sends RAND(i) and AUTN(i) of the i-th selected vector to the MS, enabling the MS to verify the correctness of SQN and compute the corresponding response RES(i).
6. User Authentication Response: The MS verifies the correctness of SQN by computing MAC and comparing it with the MAC carried in AUTN(i). If matched, the MS computes and sends the corresponding response RES(i) back to the SN in a response message. The verification process is depicted in Fig. 3b.
7. Authentication Result: Once the SN receives and verifies RES(i) correctly, it chooses the corresponding CK/IK as the session key to protect its communication with the MS. In the meantime, the MS computes its CK/IK accordingly. Hence both the MS and SN reach a common session key, which terminates the UMTS AKA protocol.

Apart from UMTS AKA, other protocols such as UMTS X-AKA [7] have also been devised to reduce signaling traffic in some extent. However, these conventional mechanisms operate mostly on per-station basis. When more than one MS of a group visit an SN, current mechanisms require the SN to initiate multiple authentication processes for different MSs with the same HN, causing nontrivial overhead during handover.

3 Group-Based AKA Protocol

We propose a group-based AKA (G-AKA) protocol to facilitate users with subscribership in a common HN to roam from network to network. In our G-AKA scheme, every MS provides its identity when visiting an SN. Upon reception, the SN examines whether the MS belongs to an active group of which any member has completed full authentication. If not, the SN acquires authentication data for the MS and its associated group from the concerned HN. This leads MSs of the same group to share the same authentication data, including group temporary authentication key and other necessary information. To realize, our scheme

Table 1 Index table internal to the HN

Group	Group ID	Member ID	Initial value	Other information
G1	ID_{G1}	ID_{M1-1}	IV_{M1-1}	–
		ID_{M1-2}	IV_{M1-2}	–
		–	–	–
		ID_{M1-n}	IV_{M1-n}	–
G2	ID_{G2}	ID_{M2-1}	IV_{M2-1}	–
		–	–	–

comprises three procedures: group information setup, authentication data distribution, and mutual authentication and key agreement, as shall be described in following three subsections.

3.1 Group Information Setup

In our architecture the HN sends to an SN authentication data for a group of MSs, as opposed to sending authentication data for respective MSs. Initially, the HN configures group information of MSs, including an index table and GAK. As shown in Table 1, the index table contains fields of group identity, member identity, initial value (IV_u) for each user member u, and other context information. For convenience of illustration, the group in the first entry is referred to as $G1$, the second entry as $G2$, and so forth. We let the initial value IV_u be large and unique. IV_u will behave as a sequence number for synchronization between the user u and its SN.

The HN also assigns each MS an individual key for communication confidentiality, and each group a common group key, namely GAK, for authentication purpose. The generation and distribution of GAKs along with MSs joining or leaving a group can be managed by the Authentication Center (AuC) within the home network [15–17]. Furthermore, the HN, SNs, and MSs contain MAC algorithms for authenticating messages. The inputs for MAC algorithms consist of a secret key and related information, and outputs of MAC algorithms are irreversible. Without loss of generality, we denote MAC algorithms by f^0, f^1, f^2, and f^3, respectively, for the HN to authenticate an MS, for an MS to authenticate an SN, for an SN to authenticate an MS, and for key generation.

3.2 Authentication Data Distribution

We now consider how the HN distributes authentication data for MSs of the same group migrating to an SN. Let MS_{M1-1} be the first MS initiating authentication in the roaming group $G1$. Furthermore, in what follows a parenthetical term in any message represents some specific information to be conveyed in the payload. The distribution procedure is shown in Fig. 4, where challenge-response messages can be embodied by CHAP [14], in following lines.

1. ID Request: The SN attempts to identify MS_{M1-1}.
2. ID Response ($AUTH_{G1}$): Upon receiving the ID Request message, MS_{M1-1} generates $AUTH_{G1} = (ID_{G1} \| ID_{M1-1} \| RN_{M1-1} \| MAC_{M1-1})$, where ID_{G1} denotes the group identity, ID_{M1-1} is the mobile user's identity, RN_{M1-1} represents a random number, and $MAC_{M1-1} = f^0(K_{M1-1}, RN_{M1-1})$ for the HN to authenticate MS_{M1-1} (see also Fig. 8). Here K_{M1-1} is the pre-shared secret key with the HN.

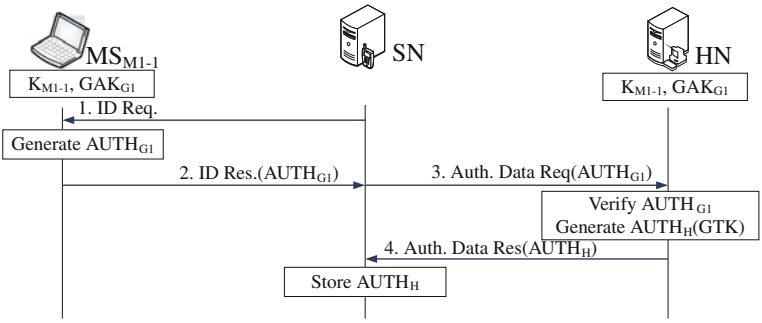

Fig. 4 Authentication data distribution

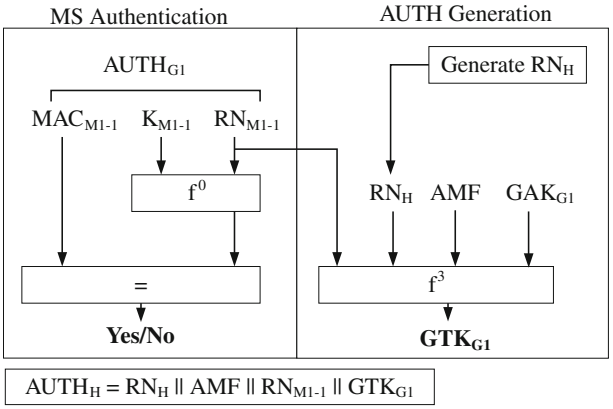

Fig. 5 The HN verifying AUTH_{G1} and generating AUTH_{H}

3. Authentication Data Request (AUTH_{G1}): Since $\text{MS}_{\text{M1}-1}$ is new, the SN without knowledge of the MS relays the foregoing message from $\text{MS}_{\text{M1}-1}$ to the HN. The HN shall authenticate the roaming group ($G1$) which $\text{MS}_{\text{M1}-1}$ belongs to.
4. Authentication Data Response (AUTH_{H}): As shown in Fig. 5, the HN verifies the received $\text{MAC}_{\text{M1}-1}$ in AUTH_{G1} using $\text{K}_{\text{M1}-1}$ (the pre-shared key with $\text{MS}_{\text{M1}-1}$). If $\text{MS}_{\text{M1}-1}$ is found authentic, the HN retrieves the corresponding group authentication key GAK_{G1} to generate a Group Transient Key $\text{GTK}_{\text{G1}} = f^3(\text{RN}_{\text{M1}-1} \| \text{RN}_{\text{H}} \| \text{AMF} || \text{GAK}_{\text{G1}})$.

Group authentication data sent to the SN contains $\text{AUTH}_{\text{H}} = (RN_{\text{H}} \| \text{AMF} \| \text{RN}_{\text{M1}-1} \| \text{GTK}_{\text{G1}})$, where RN_{H} is a newly selected random number by the HN, AMF denotes contents of the Authentication Management Field, and $\text{RN}_{\text{M1}-1}$ is the random number chosen *a priori* for $\text{MS}_{\text{M1}-1}$. The group information for $G1$ (entire record associated with $G1$ drawn from the HN's index table) is piggy-backed on Message 4 above. The SN will keep the received information, particularly AUTH_{H}, in local storage for future use.

3.3 Mutual Authentication and Key Agreement

Upon receipt of group authentication data, the SN proceeds to mutual authentication and key agreement with $\text{MS}_{\text{M1}-1}$. This procedure validates the authenticity of both sides and

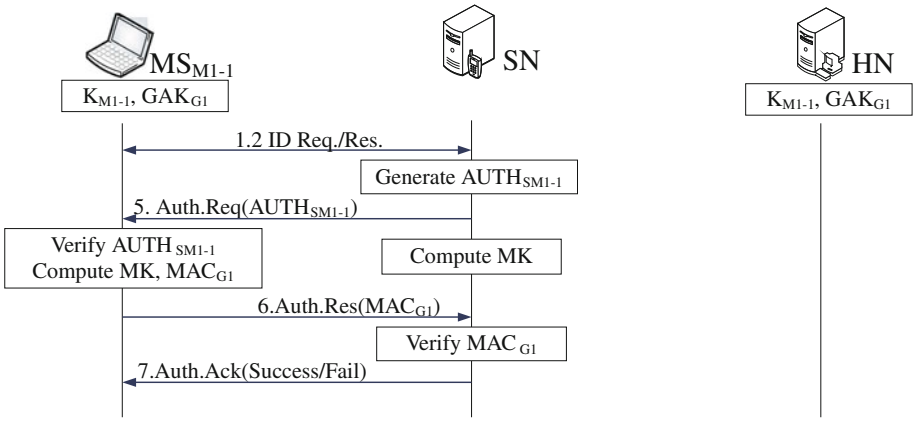

Fig. 6 Mutual authentication and key agreement (subsequent to the message flow of Fig. 4)

establishes the pairwise session key for message encryption between the MS and the SN. As a result of computing different responses of challenge messages with different arguments, both the MS and the SN authenticate each other by verifying the correctness of responses. After both sides have been successfully authenticated, the pairwise session key is generated to protect the traffic in between. This procedure is depicted in Fig. 6, starting with Message 5.

5. Authentication Request ($\text{AUTH}_{\text{SM1}-1}$): After acquiring AUTH_{H} for group $G1$, the SN conducts instantly the i-th run of mutual authentication with $\text{MS}_{\text{M1}-1}$ by generating $\text{AUTH}_{\text{SM1}-1} = (\text{AMF}\|\text{RN}_{\text{H}}\|\text{RN}_{\text{M1}-1}\|\text{MAC}_{\text{S}}\|\text{RN}_{\text{SM1}-1})$, where the first three parameters are meant for $\text{MS}_{\text{M1}-1}$ to generate GTK_{G1}, $\text{MAC}_{\text{S}} = f^1(\text{GTK}_{\text{G1}}\|\text{RN}_{\text{M1}-1}\|\text{IV}_{\text{M1}-1}+i)$, and $\text{RN}_{\text{SM1}-1}$ is a nonce chosen by the SN to challenge $MS_{\text{M1}-1}$. While waiting for a response from $\text{MS}_{\text{M1}-1}$, the SN computes the master key $\text{MK} = f^3(\text{GTK}_{\text{G1}}\|\text{IV}_{\text{M1}-1} + i\|\text{RN}_{\text{M1}-1}\|\text{RN}_{\text{SM1}-1})$ for subsequent sessions with $\text{MS}_{\text{M1}-1}$ in advance. (See Fig. 7).
6. Authentication Response (MAC_{G1}): To acknowledge Authentication Request ($\text{AUTH}_{\text{SM1}-1}$), $\text{MS}_{\text{M1}-1}$ computes GTK_{G1} using the first three arguments in $\text{AUTH}_{\text{SM1}-1}$ and GAK_{G1} stored in each MS of the same group. $\text{MS}_{\text{M1}-1}$ then authenticates the SN by computing and comparing the corresponding result with MAC_{S}. After successfully authenticating the SN, $\text{MS}_{\text{M1}-1}$ calculates the master key MK with respect to the SN and generates a message back to the SN containing $\text{MAC}_{\text{G1}} = f^2(\text{GTK}_{\text{G1}}\|\text{RN}_{\text{SM1}-1}\|\text{IV}_{\text{M1}-1} + i)$. Such operations are diagrammed in Fig. 8.
7. Authentication Result (Success/Failure): Upon receiving an Authentication Response message carrying MAC_{G1}, the SN checks whether $\text{MS}_{\text{M1}-1}$ has produced the correct response using operations as in Fig. 7. Then a message with a status code indicating either success or failure for mutual authentication is sent to $\text{MS}_{\text{M1}-1}$, whence our key agreement procedure is completed.

After full authentication, both $\text{MS}_{\text{M1}-1}$ and its SN share a common MK that shall become essential material for subsequent key derivations.

When a second member, say $\text{MS}_{\text{M1}-2}$, arrives, the SN starts on mutual authentication and key agreement with $\text{MS}_{\text{M1}-2}$ locally using the existing GTK_{G1}. More specifically, Steps 3 and 4 in the prescribed authentication data distribution procedure can be bypassed, leaving out signaling traffic between SN and HN. In this regard, however, the SN needs to generate a new random number $\text{RN}_{\text{SM1}-2}$ to create a new challenge message for $\text{MS}_{\text{M1}-2}$ (similar to Step 5

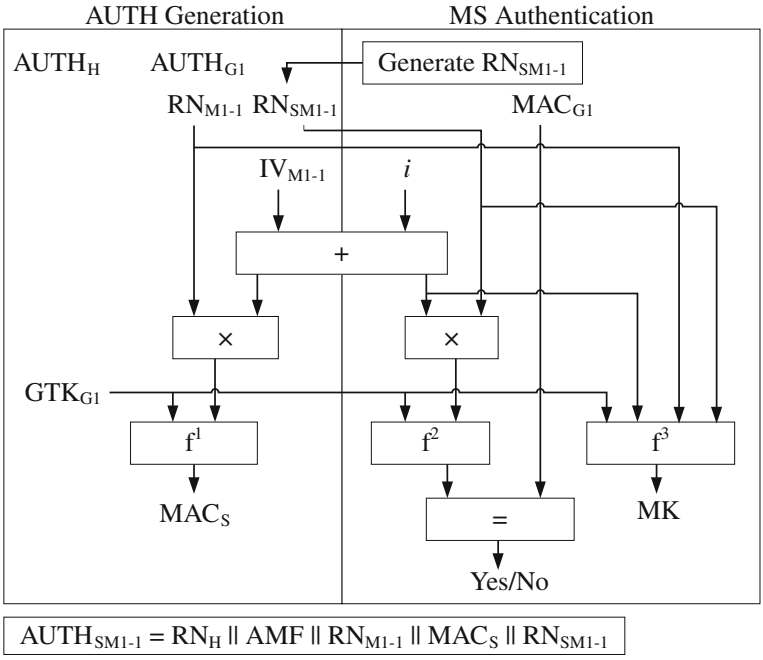

Fig. 7 An SN generating $\text{AUTH}_{\text{SM1}-1}$ and MK and verifying MAC_{G1}

in Fig. 6). Using distinct arguments from those for $\text{MS}_{\text{M1}-1}$, such as $\text{RN}_{\text{SM1}-2}$, $\text{RN}_{\text{M1}-2}$ and $\text{IV}_{\text{M1}-2}$, our scheme ensures not only the freshness of challenge-response messages but also the uniqueness of master keys for respective MSs.

To conclude this section, we remark that, when a group of mobile subscribers visits a network, our protocol saves both the HN and the SN from repeated message exchanges by providing group authentication data and GTK. The latter is used in place of GAK to prevent GAK from being divulged to eavesdroppers. Observe that a GTK allows of periodic or aperiodic updates whenever new random numbers are available. Such a design strengthens the robustness and sustainability of the protocol.

4 Discussions

This section covers security analysis and evaluation of the proposed G-AKA protocol. We will show that G-AKA maintains its due security level and quantify its outperformance to other schemes in terms of storage and message complexity.

4.1 Security Analysis

Here we reason that G-AKA can achieve the comparable security level as other contemporary AKA protocols, without compromising overall security of the system. Our reasoning is as follows.

Mutual Authentication First, each MS achieves mutual authentication with its HN by Steps 2 and 4 of Sect. 3.2 and Step 6 of Sect. 3.3. To see this, in Step 2 an MS, say MS_{M1}, provides $AUTH_{G1}$ to prove its authenticity. Step 4 enables the HN to validate the MS's proof and to generate a GTK as part of $AUTH_H$ addressed back to the SN over a secure communication channel. Step 6 allows MS_{M1} to authenticate the HN implicitly, based on the premise that a correct GTK available to the SN can only be sourced from the genuine HN.
Besides, Steps 6 and 7 of Sect. 3.3 ensure mutual authentication between MS_{M1} and its SN. This is because the MS authenticates its SN by comparing its computed MAC_S with that in $AUTH_{SM1}$ (Step 6). This step requires the SN to be legitimate for acquiring a correct GTK from the HN and thereby the SN is assured of authenticity (not a rogue entity). On the other hand, the SN checks whether the returned MAC_{G1} from the MS is correct (Step 7). By doing so, the MS is authenticated by the SN, and *vice versa*.
Secure Key Derivation In our G-AKA protocol, the pairwise master key for an MS and its SN are derived on either peer side directly, without being delivered over any communication channels. Therefore, the keying material is prevented from being disclosed, attacked, or intercepted by adversaries.
Replay Attack Resistance Our protocol operates free from potentially ill effects caused by an attacker intercepting and re-transmitting authentication messages to spoof or get admitted to the system. In our protocol, whenever any MS, e.g., MS_{M1} requests to be authenticated, new random numbers RN_{M1} and RN_{SM1} are produced on the MS and its SN side, respectively, for temporary use in generating challenge messages toward the opposite side. Meanwhile, these two sites maintain an identical initial value IV_{M1-1} to keep themselves synchronized throughout AKA processing. An out-of-sync initialization value will lead to authentication failure. Thus a node without the required random numbers and initial value cannot perform a replay attack on our system.
Fraud Control Although all the members of a group share a common GTK, our protocol is not exposed to masquerade attack by the same group. To see this, suppose that MS_{M1-1} attempts to impersonate another member, say MS_{M1-2}, by eavesdropping traffic between MS_{M1-2} and the SN. However, MS_{M1-1} is unable to generate a correct Authentication Response message carrying MAC_{G1}, because MAC_{G1} can result only from GTK, RN_{M1-2}, and unique IV_{M1-2} associated with MS_{M1-2}. For the same reason, MS_{M1-1} cannot learn the master key used by MS_{M1-2} and the SN. By application of Steps 6 and 7 of Sect. 3.3, the SN aware of the incorrect response message will block the fraud straightway. So, the SN can easily tell one member from another even though all members use the same GTK. Neither can MS_{M1-1} not decrypt traffic between MS_{M1-2} and the SN, since the traffic has been protected with a master key which is unknown to MS_{M1-1}. A central treatment is that we employ distinct random numbers and initial values for different MSs, so as to guarantee the freshness of challenge-response messages and the uniqueness of master keys for respective MSs.

There are several additional security measures in our design. We avail ourselves of random numbers RN_H and RN_{M1-1} generated by the HN and the first MS of a group, respectively, to ensure the freshness of GTK, as shown in Figs. 7 and 8. These two nonces are incorporated jointly to protect GTK. Moreover, our scheme uses some parameter to record how many authentications have been performed by an MS so far. The parameter is embedded in our authentication processes (Step 5 of Sect. 3.3) to consolidate the freshness of each authentication message.

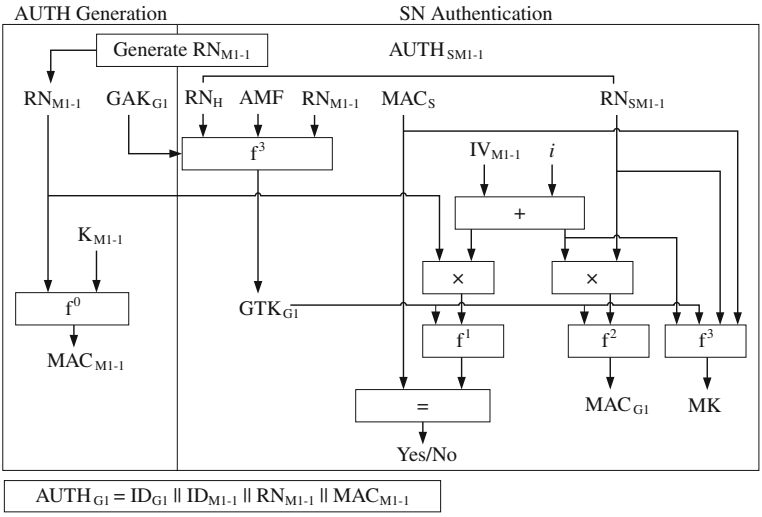

Fig. 8 An MS generating $AUTH_{G1}$, verifying MAC_S, and generating MAC_{G1}

While our scheme reduces the overhead of the HN, the security focused on SN should be investigated because the SN's responsibility is increased. If both the SN and HN are under common ownership of an operator, the communication channel between these two sites must be secured when the network was deployed. Any well-known robust security protocols can be leveraged here to enforce confidential information transfers in between. In case that the SN and the HN belong to different operators, it is noted that more and more network service providers have interoperation or roaming agreements. In this case, cross-realm authentication involves the use of AAA (authentication, authorization, and accounting) entities situated in respective network domains. In practice, the AAA authority of the SN is charged with establishing a security association with the AAA entity of the HN for exchanging MS' credentials over a secure channel [5]. Both AAA entities are configured with sufficient security relationships and access controls, so they can negotiate the authorization that enables MSs to have access to requested resources. Therefore, the analysis above ensures that the SN in our architecture does not lead to security vulnerability, nor does our G-AKA weaken overall system integrity.

4.2 Performance Results

Let us now compare the proposed G-AKA with UMTS AKA and UMTS X-AKA [7], two probably best known counterpart protocols for mobile telecommunications networks. Note that both our G-AKA and UMTS X-AKA authorize an SN to authenticate an MS locally after the HN has authenticated the MS. Yet our G-AKA takes a step further by introducing the use of a group authentication key to enable the SN to authenticate other MSs of the same group. Therefore our G-AKA can reduce both authentication delay and signaling overhead within the core network.

Considering that n MSs form g groups and that each MS initiates m (re)authentications, Table 2 compares the three AKA protocols in terms of how many signaling messages shall be

Table 2 Message complexity of AKA protocols

Protocol	Signaling message count			
	1 MS		n MSs	
	$m = 1$	$m > 1$	$m = 1$	$m > 1$
UMTS AKA	7	$7m$	$7n$	$7mn$
UMTS X-AKA	7	$7 + 5(m - 1)$	$7n$	$n(7 + 5(m - 1))$
G-AKA	7	$7 + 5(m - 1)$	$7g + 5(n - g)$	$7g + 5(n - g) + 5n(m - 1)$

generated. For an MS, the total number of signaling messages required by UMTS AKA grows linearly with m, thus amounting to m times the message count for running a single round of the protocol, namely $7m$. In comparison, UMTS X-AKA reduces message complexity if m authentication requests are initiated by the same MS that undergoes full authentication at the cost of 7 messages, with provision of a transient key to the following $(m-1)$ local AKA operations. Since each such local operation costs an MS 5 messages [7], a total of $7+5(m-1)$ messages is required. Nevertheless, UMTS X-AKA operates on per-MS basis and generates authentication data for each MS. Accordingly, signaling overhead increases in proportion to n when there are n MSs in the system performing X-AKA procedures. As a result, signaling traffic between the SN and HN may place a nontrivial processing burden on all the network entities along the routing path.

In our architecture, the first MS of a group carries out the proposed procedure at the expense of 7 messages, whereas the procedure for each of the remaining members entails 5 messages. The latter message complexity applies to every MS invoking a subsequent re-authentication process. Given n MSs forming g groups, the first MS per group still experiences 7 message exchanges. However, for any of other $(n-g)$ members, 5 message exchanges suffice to complete our protocol. So, signaling traffic reaches $7g+5(n-g)$ in number. When each MS needs to perform our protocol another $(m-1)$ times, every MS requires 5 additional messages each time. Therefore overall signaling takes on the sum of $7g + 5(n - g) + 5n(m - 1)$ messages.

From Table 2, it can be seen that our G-AKA outperforms counterpart protocols if $n > g$. Our design will gain marked improvement when $n \gg g$, which is a common occurrence because MSs typically outnumber groups composed of these MSs. The outperformance is attributed to our reduction from a factor of the number of MSs to a factor of the number of MS groups. In this manner, not only signaling traffic in the core network but also authentication delays (part of handover delays) are saved by an appreciable amount. These advantages will become multiplicative when more than one group of MSs need to run re-authentication and key agreement processes repeatedly. Figure 9 plots the number of signaling messages incurred in different combinations of n MSs, each invoking an AKA protocol m times, and g groups.

We now address storage space complexity at an SN site. For UMTS AKA, each MS requires its SN to keep a set of Authentication Vectors (AVs) in storage, so n MSs occupy an order of $n \times$ AV storage space. As to UMTS X-AKA, $n \times$(Temporal Key (TK)+AUTH) space is occupied because a piece of authentication data (TK+AUTH) is meant for a single MS. Unlike UMTS AKA and X-AKA, however, our G-AKA utilizes group authentication data, i.e., a Group Authentication Key and an Index Table entry, instead of maintaining per-user information. Provided g groups of MSs, the SN spends $g \times$(GTK+Index Table entry) storing authentication data. By comparison, our G-AKA brings the SN g pieces of group data, while counterpart protocols necessitate n distinct copies.

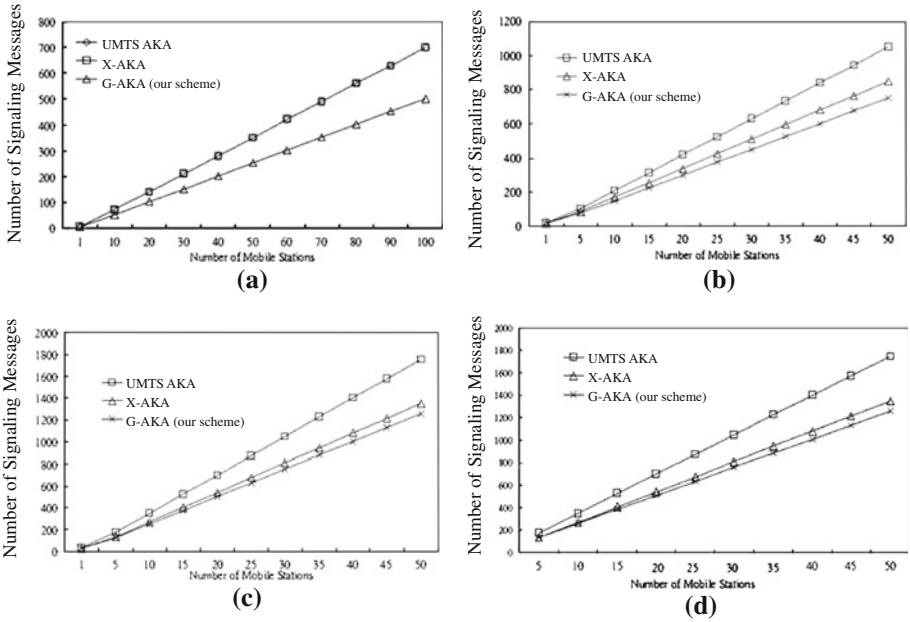

Fig. 9 Comparison of signaling message counts of AKA protocols. **a** $m = 1, g = 1$, **b** $m = 3, g = 1$, **c** $m = 5, g = 1$, **d** $m = 5, g = 5$

5 Conclusion

The proposed protocol operates under a notion of group authentication, producing by far fewer message exchanges than those required by counterpart protocols when a group of mobile subscribers request for authentication and key agreement. Unlike conventional protocols that generate different data for respective MSs, our protocol uses a shared GTK and data structure (Index Table) to authenticate these MSs. Protocol operations are thus refined greatly, which allows of better scalability when there is a growing number of MSs in the system.

To summarize, this study identified group movement behavior of MSs roaming from a home network and then developed a mutual authentication and key agreement protocol whereby these MSs can gain secure, fast access to a common visited network. The visited network is authorized to authenticate a group of MSs locally, after the first visiting member has authenticated itself to the network. This results from authentication with the first MS bringing the visited network some security context, which is of utility to authenticate other MSs of the group. So significant authentication delays and signaling overhead between the serving network and home network are reduced. Such a design expedites fast reauthentications as well. Concerning the effectiveness of our development, Sect. 4.1 has reasoned that the protocol is an efficient means to system-wide access control, without trading performance for security and robustness to the extent that security requirements are unduly weakened. Although group-based authentication is done in our architecture, personal security is still honored, as reasoned in fraud control. Section 4.2 has also quantified how our protocol outperforms counterpart schemes. Both qualitative and quantitative comparisons indicate the usefulness of our design.

In closing, we stress that mutual authentication and key agreement are preliminary to any secure handover procedure. Our development serves to refine handover processes by which mobile users, upon visiting a new network, can resume secure communication with the network by completing handover sooner than would otherwise required. As another variant, our design further speeds up handover procedures if security context information associated with MSs can be distributed to neighbor networks these MSs shall visit next. In a broad sense, our design is well applicable to any system with a remote authentication sever, for example, for network access control or entrance security control.

References

1. 3rd Generation Partnership Project. (2001). Security architecture, 3GPP TS 21.133.
2. IEEE 802.16-2004. (2004). IEEE standard for local and metropolitan area networks part 16: Air interface for fixed broadband wireless access systems.
3. Arkko, J., & Haverinen, H. (2006). Extensible authentication protocol method for 3rd generation authentication and key agreement (EAP-AKA). (RFC 4187). IETF Network Working Group.
4. Bargh, M. S., Hulsebosch, R. J., Eertink, E. H., Laganier, J., Zugenmaier, A., & Prasad, A. R. (2007). UMTS-AKA and EAP-AKA inter-working for fast handovers in all-IP networks. In *IEEE Globecom* (pp. 1–6).
5. Glass, S., Hiller, T., Jacobs, S., & Perkins, C. (2000). *Mobile IP authentication, authorization, and accounting requirements*. (RFC 2977). IETF Network Working Group.
6. Haverinen, H., & Salowey, J. (2006). *Extensible authentication protocol method for global system for mobile communications (GSM) subscriber identity modules (EAP-SIM)*. (RFC 4186). IETF Network Working Group.
7. Huang, C. M., & Li, J. W. (2005). 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)* (pp. 392–397).
8. Kasera, S., & Narang, N. (2005). *3G mobile networks: Architecture, protocols, and procedures*. New Delhi: McGraw-Hill.
9. Mitchell, C. J. (2003). *Security for mobility*. London, IN: Institution of Electrical Engineers.
10. Niemi, V., & Nyberg, K. (2004). *UMTS Security*. Chichester: Wiely.
11. Ou, H.-H., Hwang, M.-S., & Jan, J.-K. The UMTS-AKA protocols for intelligent transportation systems. *EURASIP Journal on Wireless Communications and Networking*, http://www.hindawi.com/journals/wcn/aip.267283.pdf.
12. TETRA Association. http://www.tetramou.com/.
13. Raya, M., & Hubaux, J. P. (2005). The security of VANETs. In *Proceedings of the 2nd ACM international workshop on vehicular Ad Hoc networks* (pp. 93–94).
14. Simpson, W. (1996). *PPP challenge handshake authentication protocol (CHAP)*. (RFC 1994). IETF Network Working Group.
15. Tan, C. H., & Teo, J. C. M. (2005). An authenticated group key agreement for wireless networks. In *Proceedings of the IEEE wireless communication and networking conference (WCNC)*.
16. Wallner, D., Harder, F., & Agee, R. (1999). *Key management for multicast: Issues and architectures*. (RFC 2626). IETF Network Working Group.
17. Wong, C. K., Gouda, M., & Lam, S. S. (2000). Secure group communications using key graph. *IEEE/ACM Transactions on Networking, 8*(1), 78–85.

Author Biographies

Yu-Wen Chen received her B.S. degree in Information Management from National Cheng-Chih University in 2006 and M.S. degree in Computer Science and Information Engineering from National Chiao Tung University, Taiwan, in 2008. Her current research interests include wireless Internet protocols and wireless Internet applications.

Jui-Tang Wang received his Ph.D. degree in computer science and information engineering from National Chiao Tung University, Taiwan, in 2008. He is with the Information and Communications Laboratories, Industrial Technology Research Institute, ROC. His current research interests include WiMAX network and Long-Term Evolution technologies.

Kuang-Hui Chi received the B.S. degree in computer science and engineering from Tatung University in 1991 and the M.S. and Ph.D. degrees in computer science and information engineering in 1993 and 2001, respectively, from National Chiao Tung University, Taiwan. He is Associate Professor at the Department of Electrical Engineering, National Yunlin University of Science and Technology, Taiwan. His current research interests include beyond wireless Internet and protocol verification. Dr. Chi is a member of the ACM and the IEEE.

Chien-Chao Tseng is currently Professor and Director of the Institute of Network Engineering at National Chiao Tung University, Hsinchu, Taiwan, ROC. He received his B.S. degree in Industrial Engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1981, and M.S. and Ph.D. degrees in Computer Science from the Southern Methodist University, Dallas, Texas, U.S.A., in 1986 and 1989, respectively. His research interests include wireless Internet infrastructure and protocols, and wireless Internet applications.