2012 Fourth International Conference on Intelligent Networking and Collaborative Systems

Dynamic Group based Authentication Protocol for Machine Type Communications

Yueyu Zhang, Jie Chen, Hui Li, Wujun Zhang, Jin Cao and Chenzhe Lai
State Key Laboratory of Integrated Service Networks
Xidian University
Xi'an, P. R. China, 710071
Email: yyzhang@xidian.edu.cn, {jchen,lihui}@mail.xidian.edu.cn, wjzhang@xidian.edu.cn, {caoj897,lcz.xidian}@gmail.com

***Abstract*—In cellular networks, Machine-Type Communication (MTC) has shown the advantages, including better coverage and lower network deployment cost, which makes it become the hotspot in industry area. However, the current cellular network is designed for human-to-human communication (H2H), and less optimal for machine-to-machine, machine-to-human or human-to-machine applications. One of the most urgent issues, which network operators are currently facing, is MTC related signalling congestion and overload. Especially, when a large co-located MTC group almost simultaneously accesses or periodically transmits, masses authentication data may cause a serious congestion in VLR/SGSN node and overload of link between serving network (SN) and home environment (HE). This paper provides a practical group based authentication and key agreement for MTC roaming scenario. In the group, each MTC device not only shares a secret key with home environment, but also a group secret key with home environment and other MTC devices belong to the same group. Then signalling data is reduced between serving network and home environment, by the MTC device using this group key as an authentication key and authenticating locally. Furthermore, a method of group key update is proposed to suit for dynamic MTC group. Finally, the analysis shows that the proposal can remarkably lower the effort of handling large MTC group, and minimize the change of operator's core network(CN).**

***Keywords*-machine-to-machine; authentication and key agreement; shared group key; congestion control**

I. Introduction

Machine-type Communication is a form of data communication which involves one or more entities that do not necessarily need human interaction. It is different to current mobile network communication services as it have some characteristic features[1]: different market scenarios, data communications, lower costs and effort, little traffic per terminal and a potentially very large number of communicating terminals with. It is known that traditional wireless terminals communicating with networks should be largely "manned" by humans, but, communications from and to MTC devices are freed from this constrain. Thus, MTC communication is considered as one of the next frontiers in wireless communications. However, the current cellular network is designed for human-to-human communication, and less optimal for machine-to-machine, machine-to-human or human-to-machine applications. As the networks will not be specially MTC-enabled, some standardization organizations devote themselves to improving current cellular networks to cope with challenges brought by MTC communication. One of the objects of Third Generation Partnership Project (3GPP) Release11[2] is reducing the impact and effort of handling large machine-type communication groups, especially for groups of MTC devices that are co-located. 3GPP also expects that a mechanism can be provided to reduce peaks in the data and signalling traffic resulting from very large numbers of MTC devices, which (almost) simultaneously attempt data or signalling interactions.

Our approach. This paper presents a dynamic group based authentication and key agreement (DGBAKA) for MTC communication. In this co-located group, each MTC device pre-shares an additional secret key with home environment and other MTC devices belong to the same group. The key will be shared for authenticating with serving network locally. Using this key, the arrive rate of authentication request message sent to home environment will be lowered, which can remarkably reduce the signaling data load between serving network and home environment. Furthermore, we deal with methods of group key generation and update for enrolment and revocation of the membership, satisfying the dynamic nature of MTC group in practical application.

Related work. MTC related congestion and overload is a primary question when MTC communication was introduced into cellular network. To avoid data congestion, network operators can pre-define or alter the time period based on some criteria, e.g. daily traffic load, and only allow MTC devices to access the network during the pre-defined time period. In the case of radio access network overload, network operator can also prevent or delay MTC devices from data transfer[3]. Nevertheless, when a large co-located MTC group is roaming or almost simultaneously accesses, masses authentication data could also cause a serious congestion in VLR/SGSN node and overload of link between serving network and home environment, which is one of the most urgent issues which network operators are currently facing, and not yet been taken into account by 3GPP.

In conventional mode, UMTS-AKA[4] and its derivates[5], [6] are used for authenticating third generation H2H terminal, just as EPS-AKA[7] for LTE or LTE-Advanced H2H terminal. Normally, these protocols were designed for authenticating between single terminal and CN. Thus the existing protocols could not be directly used

978-0-7695-4808-1/12 $26.00 © 2012 IEEE
DOI 10.1109/iNCoS.2012.86

for improving the performance of co-located MTC group. In the literature, there are few group-based authentication and key agreement protocol been proposed. Aboudagga, Quisquater and Eltoweissy[8] present a mobile group authentication protocol(mGAP) for mobile groups and individual nodes during roaming across homogeneous or heterogeneous administrative domains. As being constructed based on public key infrastructure, mGAP is not appropriate for MTC group. Ngo, Wu and Le etc.[9] propose an the authentication model using dynamic keys and group key management. They add some management entities, and adopt a very different key material comparing with current cellular network architecture. In addition, a G-AKA[10] is demonstrated for a group of mobile stations roaming from the same home network to a serving network. However, the protocol does not support dynamic group. Thus, no appropriate group-based authentication methods currently satisfy all demands of MTC communications.

Organization. The remainder of this article is organized as follows. In section II, the related background is described. In section III, a group based authentication and key agreement is proposed. Related discussion and analysis will be presented in section IV. Finally, we give our conclusions in section V.

II. Background

A. MTC security Architecture

The 3GPP system provides transport and communication services (including 3GPP bearer services, IMS and SMS) optimized for end to end application between the MTC device and the MTC server. On one hand, MTC devices can connect to the 3GPP network (UTRAN, E-UTRAN, GERAN, I-WLAN, etc) via MTCu interface that could be based on Uu, Um, or LTE-Uu interface. MTC device communicates with a MTC server or other MTC devices using the 3GPP bearer services, SMS and IMS provided by the PLMN. The MTC server is an entity which connects to the 3GPP network via MTCi/MTCsp interface and thus communicates with MTC devices. It should be noted that MTCi could be based on Gi or Sgi interface, and MTC server may be an entity outside of the operator domain or inside an operator domain. On the other hand, MTC devices and Non-MS MTC devices can also access CN through the mobile gateway of capillary networks which are made of a variety of links, either wireless or wired. The MTC security architecture described in Figure 1 is based on the system architecture (Non-Roaming Architecture) given in TR 23.888[3]. It is a potential high level security architecture for MTC Non-Roaming Architecture, in which three different areas are defined[3]:

A) Security for MTC communication between the MTC device and 3GPP network can be further divided to:
   - A1) Security for MTC communication between the MTC device and RAN.
   - A2) Security for MTC communication between the MTC device and NAS.
   - A3a) Security for MTC communication between the MTC device and MTC-IWF (for 3GPP access).
   - A3b) Security for MTC communication between the MTC device and ePDG (for non-3GPP access).

B) Security for MTC communication between the 3GPP network and an entity outside the 3GPP network can be further divided to:
   - B1) Security for MTC communication between the MTC server and 3GPP network in indirect deployment model. This can be further divided into security aspects when the MTC server is within the 3GPP network and when it is outside the 3GPP network.
   - B2) Security for MTC communication between the MTC application and 3GPP network in direct deployment model.

C) Security for MTC communication between the an entity outside the 3GPP network and MTC device can be further divided to:
   - C1) Security for MTC communication between the MTC server and MTC device in indirect deployment model.
   - C2) Security for MTC communication between the MTC application and MTC device in direct deployment model.

In the following, we will analyze the threats and security requirements based on the architecture.

B. Threats and Security Requirements

Firstly, this subsection is intended to provide an overview of the security threats for MTC communication. Main threats[11] are described as following.

1) **Device triggering attack**
MTC device trigger is used by MTC server to establish the communication with MTC device. It should be guaranteed to the MTC User that MTC devices can only be triggered by authorized MTC servers. When a MTC device is in detached state, the attacker can impersonate a network to send a trigger indication to it, and cause the MTC device to connect to a false network. If the IP or TCP/UDP port of the server included in trigger indication tampered, it will cause that MTC device is unable to communicate with the correct MTC server and also wastes its power consumption. Furthermore, the 3GPP network has to keep track of the location of the MTC device in order to sent the device trigger to it. Since some types of MTC device can be linked to an individual and unlike to normal UE, they ofen can not be turned off, this will leak the individual's privacy with respect to location information tracking by the network.

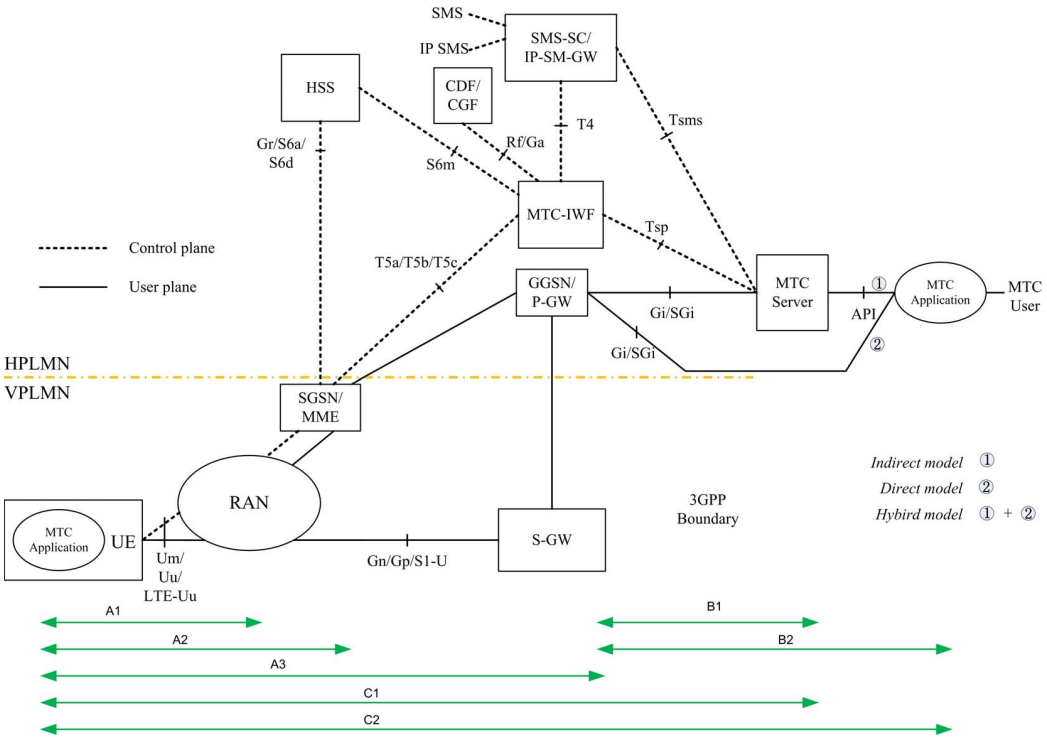

Figure 1. Security architecture for Machine Type communication

2) **Denial of service attack**
In the overload situation, it is unrealistic for SGSN/MME to perform a successful AKA with the MTC device, then perform the security mode command procedure for integrity protection and encryption. Thus the MM/GMM/EMM Reject message will be sent to the MTC device without integrity protection. Any false base station can fake the reject cause values in the reject message to launch a denial of service attack to the MTC devices and the network.

3) **Access priority indicators attack**
If the access priority indicators are sent without any protection, the attackers can tamper with it or delay tolerant access to the normal state, which will make many MTC devices connect the network during congestion control mechanism working. Vice versa, if an attacker replaces the access priority indicator in the request sent by normal UE with a fake one, the service of the UE will be maliciously degraded.

4) **External interface attack**
The interface between MTC server and CN may be over an insecure link. Communication between the MTC server and the CN for device triggering or MTC monitoring are carried on this insecure link. Attack on the communication between MTC server and CN may cause false activities either to the MTC server, MTC device or to the 3GPP network. Moreover, some privacy sensitive information such as identities may be eavesdropped through this interface. All of these may lead to serious problems.

Based on the security architecture and the threats discussed above, MTC communication security should meet following basic requirements.

1) Although unfeasible to totally prevent an MTC device from receiving a trigger indication from a fake network, it should protect the trigger indications based on SMS, NAS signaling or user plane, for minimizing the impact of fake device triggers to the battery lifetime and unauthorized tracking of the MTC device.

2) A security mechanism is needed to prevent the DoS attack.

3) The low access priority indicator should be integrity-protected.

For adopting symmetric cryptographic, conventional cellular network should perform an AKA protocol to obtain master keys, then derive some subkeys to protect trigger indication and low access priority indicator. Only this way can we meet the above security requirements as much as possible. Hence, designing specific AKA protocol is the key of the paper for co-located dynamic MTC group.

III. Our proposal

In this section, we will present a dynamic group based optimization for MTC communication authentication. Our

proposal will be described in followings.

A. Setup

Before the AKA mechanism starts, according to the contract between MTC user and network operator, the HE maintains a table of group membership that stored only in itself. This table contains at least three columns: MTC group identity, MTC device identity and initial value(IV) for each member. Then, HE generates Group Authentication Key (GAK) shared between HE and MTC devices. The derivation of shared group key could follows key management for multicast, e.g. RFC2627[12]. It should be noted that the generated key is not a cipher key, but an authentication key. Unlike common group key agreement, key encryption keys may not be necessary for us in the procedure of key generation. A secure keying material distribution will be presented in next subsection.

It assumes that a secret key K is shared between MTC device i and HE. The shared key of MTC Group $G1$ is noted as GAK. The descriptions of message authentication functions and key generating functions used in our proposer are listed as Table I:

B. Initial distribution of GAK

There are two ways to distribute GAK initially to MTC device: one is written into Machine Communication Identity Module(MCIM) by manufacturer or network operator according to service contract, the other is configuring MCIM remotely by network operator on which is our emphasis, as shown in Figure 2. The procedure can be represented by following steps:

1) The MTC device achieves an initial attach with visited network operator by using its provisional connectivity identity, and receives a Bootstrap message from Registration Operator(RO).

2) Triggered by the Bootstrap message, MTC device sends a request of contacting Downloading Provisioning Function(DPF) with its related information, e.g. platform validation info, to the RO.

3) The RO(DPF) connects to the Selected Home Operator (SHO), and relays the MTC device information there.

4) The SHO encrypts the MCIM by using the Platform Credential(PfC) and generates the management object for MTC device, e.g. MCIMobj. Note that as a constituent part of MCIM, the new shared authentication key GAK has been generated and written into new MCIM before being encrypted.

5) The SHO delivers the encrypted MCIM within MCIMobj to the RO(DPF) and authorizes provisioning of the MCIM application to the MTC device.

6) The RO(DPF) downloads a MCIM object to the MTC device.

7) The MTC equipment provisions the downloaded MCIM into the TRusted Environment(TRE). The TRE decrypts MCIMobj by using the TRE Platform Key to obtain the MCIM.

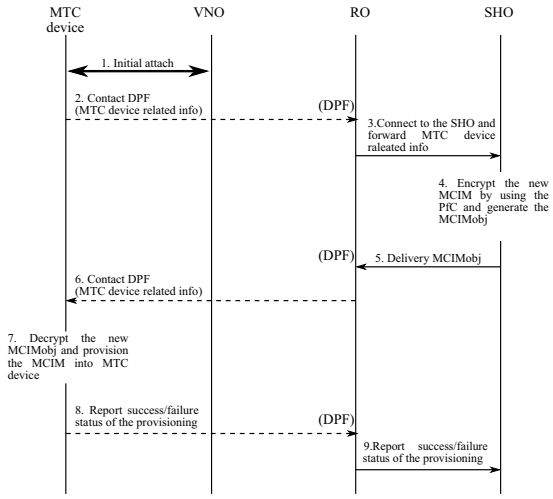

Figure 2. Remote provisioning of MCIM

8) The MTC device reports the success/failure status of the provisioning to the RO (DPF).

9) The RO (DPF) reports the success/failure status of the provisioning back to the SHO.

C. DGBAKA for MTC communication

When a co-located MTC group moves to a new SGSN, all devices in the group should authenticate with the new SGSN. Assume the first device that initiates the authentication with the new SGSN is device $M1-1$, then the complete process can be divided into two phases: one is device $M1-1$ authenticates with its HE on behalf of the group and makes the new SGSN getting array of authentication vectors and information for overall group, the other is the rest of devices in the group authenticate individually with the new SGSN. The first phase is shown in Figure 3.

1) VLR/SGSN sends identity request to MTC device $MTCD_{M1-1}$.

2) Upon receiving identity request, the MTC device $MTCD_{M1-1}$ generates $AUTH_{G1} = (ID_{G1}||ID_{M1-1}||RAND_{M1-1}||MAC_{M1-1})$, where ID_{M1-1} is its identity, ID_{G1} is the group identity it belongs to, $RAND_{M1-1}$ is a random number selected by the device and $MAC_{M1-1} = f0(K_{M1-1}, RAND_{M1-1})$.

3) The $MTCD_{M1-1}$ sends its response to VLR/SGSN.

4) Because the $MTCD_{M1-1}$ is the first device in MTC group $G1$ and there are no authentication data stored in the new SGSN, the SGSN forwards ($AUTH_{G1}$, Membership Table Flag) to HLR. It is noted that Membership Table Flag is an options. If the SGSN finds he does not store any membership information of group $G1$, then he sends a membership flag request

Table I
DESCRIPTION OF FUNCTIONS

$f0$	message authentication functions used to compute device's MAC
$f1$	message authentication functions used to compute HE's MAC
$f2$	message authentication functions used to compute RES and $XRES$
$f2'$	message authentication functions used to compute RES' and $XRES'$
$f3$	key generating functions used to compute CK
$f3'$	key generating functions used to compute cipher key used in end-to-end applications
$f3^*$	key generating functions used to compute Group Temporary Key
$f4$	key generating functions used to compute IK
$f4'$	key generating functions used to compute integrity key used in end-to-end applications
$f5$	key generating functions used to compute AK

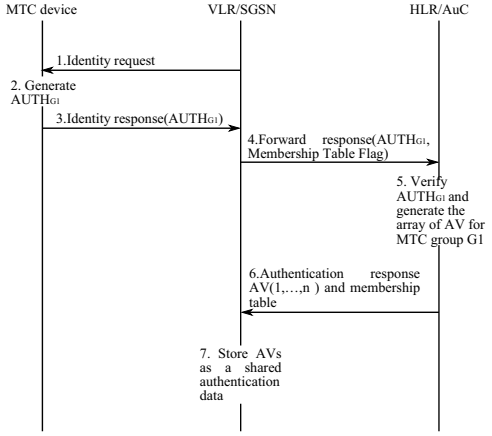

Figure 3. Distribution of group authentication data

to HLR/AuC for it.

5) The HE/AuC verifies MAC_{M1-1} in the received $AUTH_{G1}$. If the verification success, it queries GAK_{G1} and derives a Group Temporary Key for $G1(GTK_{G1})$, where $GTK_{G1} = f3^*(RAND_H, AMF, GAK_{G1})$. Then it generates an ordered array of n authentication vectors for the group. Each authentication vector consists of five components: $RAND_H$, $XRES$, CK, IK and $AUTN_H$, where $MAC_H = f1(SQN, RAND_H, AMF, GTK_{G1})$, $XRES = f2(RAND_H, GTK_{G1})$, $CK = f3(RAND, GTK_{G1})$, $IK = f4(RAND, GTK_{G1})$, $AK = f5(RAND, GTK_{G1})$ and $AUTN_H = SQN \oplus AK||AMF||MAC_H||GTK_{G1}$, $AK = f5(RAND_H, GTK_{G1})$.

6) The HLR delivers array of n authentication vectors to the SGSN. In addition, a membership table of MTC group $G1$ is also sent to SGSN.

7) The SGSN stores the array of authentication vectors as shared authentication data for the group.

Figure 4 describes the second phase in following steps:

8) After getting AVs, the SGSN selects an authentication vector from the ordered array and sends $(RAND_H, AUTN'_H)$ to the MTC device, where $AUTN'_H = SQN \oplus AK||AMF||MAC_H$. It also computes $XRES' = f2'(RAND_{M1-1}, XRES, IV_{M1-1} + i, GTK_{G1})$ and stores locally.

9) The MTC device firstly deduces $GTK_{G1} = f3(RAND_H, AMF, GAK_{G1})$, then computes $AK = f5(RAND_H, GTK_{G1})$, $SQN = (SQN \oplus AK) \oplus AK$, and $XMAC_H = f1(SQN, RAND_H, AMF, GTK_{G1})$, where $SQN \oplus AK$, AMF and $RAND_H$ are got from $AUTH_H$. The MTC device checks whether $AUTN'_H$ can be accepted by comparing $XMAC_H$ and MAC_H.

10) And, if so, produces a response $RES = f2(RAND_H, GTK_{G1})$, $RES' = f2'(RAND_{M1-1}, RES, IV_{M1-1}+i, GTK_{G1})$ which is sent back to the SGSN. The MTC device also computes $CK = f3(RAND_H, GTK_{G1})$, $IK = f4(RAND_H, GTK_{G1})$, $CK_{M1-1} = f3'(RAND_{M1-1}, CK, IV_{M1-1} + i)$ and $IK = f4'(RAND_{M1-1}, IK, IV_{M1-1} + i)$, where CK_{M1-1} and IK_{M1-1} are the cipher key and the integrity key respectively.

11) The SGSN compares the received RES' with $XRES'$. If they match the SGSN deduces CK_{M1-1} and IK_{M1-1} and considers the authentication and key agreement to be successfully completed.

12) The SGSN sends a message to the MTC device indicating authentication result.

For other devices in MTC group $G1$, e.g. M1-2, a full authentication procedure includes step 1,2,8-12, which means no signaling traffic between serving network and home enviroment.

D. Update on the shared group key

The shared group key between MTC group and HE may be updated for changes of membership. The notification of update is sent to the MTC group by HLR rather than SGSN. The procedure of GAK update can be illustrated in Figure 5. It can be divided into four steps:

1) The HE sends a GAK update notification for the MTC group to the SGSN.

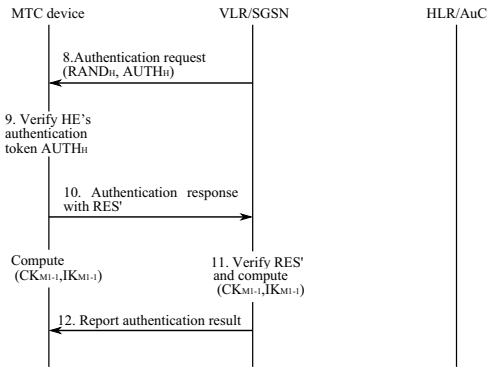

Figure 4. Mutual authentication and key agreement

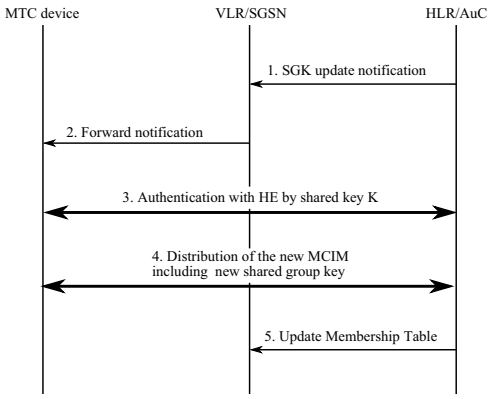

Figure 5. Update on SGK

2) The SGSN broadcasts this notification to all devices in the group.
3) Each device starts an authentication with HE by the key K shared between them.
4) After above steps, MTC device and HE performs a procedure of distribution of new MCIM that includes an updated group key.
5) HE also sends an incremental of membership table to the SGSN.

To avoiding frequently update group key, we should not regular change membership of MTC group. It implies the MCIM status of MTC device that will not accept service with other devices should not be retired or deleted as far as possible. We could temporarily de-activate MCIM, that is to say, set its status to blocks and make it unavailable for use.

IV. Analysis of DGBAKA

The protocol we proposed could solve the problem of congestion and overload caused by co-located MTC group roaming. In the scheme, all of the devices belong to the same group share a key with HE for authentication. Thus, these devices could use a same array of authentication vectors and only one in these devices needs a full AKA interaction with HE firstly, while others locally authenticate with SGSN. It has some advantages for the group based authentication for MTC communication than that with no optimization:

1) Only one full AKA with HE for a MTC group decreases the congestion probability of HLR/AuC due to roam and authentication.
2) Other devices authenticate with SGSN locally, which makes the connections between the SGSN and the HLR drastically reduced, thereby, solves the overload of link in core network.
3) Local authentication brings less delay for most devices in MTC group.

Table III
comparision of protocol's characters

Protocol	dynamics	compatibility	flexibility
G-AKA	poor	poor	good
DGBAKA	better	good	better

A. Performance Analysis

In the following, we will compare our proposal with some known protocol, i.e. UMTS-AKA, X-AKA and G-AKA. Supposing that n MTC devices form g groups and each device initiates m registrations or (re)authentications. The comparing result is illustrated by Table II. As we know, the signaling traffic of UMTS-AKA is linear with the total device number in the group, i.e. signaling message amount is $7m$. While X-AKA, G-AKA and ours reduce message complexity by authenticating the other $m-1$ devices locally, only 5 messages are required in their respective authentication process. Hence, a total of $7+5(m-1)$ messages are expended. When $m = 1$ and $n > 1$, the number of costed messages is proportion to device number, that is to say, $7n$. Considering G-AKA and our proposal, the first device in each group carries out a full authentication at the cost of 7 message. 5 messages are needed for each device in the rest, namely $7g + 5(n - g)$. Furthermore, when $m > 1$, each device should perform another $(m - 1)$ times authentication locally, which means $7g + 5(n - g) + 5n(m - 1)$ messages totally. From Table II, it can be concluded that G-AKA and our DGBAKA have a similar showing in signaling message complexity, and they are superior to UMTS-AKA and X-AKA remarkablely. However, our DGBAKA also has some characters that G-AKA does not have, refer to Table III:

1) Our proposal supports the enrolling and revocation of members while G-AKA is designed for static group, which makes the DGBAKA more practical.
2) For maximizing the use of current authentication facility, DGBAKA brings less impact to core network than G-AKA, since most changes are performed on VLR/SGSN.
3) An array of authentication vectors are transferred during authentication data delivery. Thus the refresh

Table II
SIGNALING TRAFFIC COMPLEXITY COMPARISION

Protocol	$m = 1, n = 1$	$m > 1, n = 1$	$m = 1, n > 1$	$m > 1, n > 1$
UMTS-AKA	7	$7m$	$7n$	$7mn$
X-AKA	7	$7 + 5(m-1)$	$7n$	$n(7 + 5(m-1))$
G-AKA	7	$7 + 5(m-1)$	$7g + 5(n-g)$	$7g + 5(n-g) + 5n(m-1)$
DGBAKA	7	$7 + 5(m-1)$	$7 + 5(m-1)$	$7g + 5(n-g) + 5n(m-1)$

of cipher key and integrity key could be flexible controlled by key lifetime or authentication vectors.

B. Security Analysis

Our protocol has several security properties, which can achieve comparable security level with [4], [5], [6]et. al. All of the optimization satisfies the requirements of 3GPP for cellular communication, as described below:

Mutual Authentication In our scheme, an MTC device first uses his $AUTH_{Gi}$ to get AVs and GTK_{Gi} for group i from HLR and performs a mutual authentication with HN. At the same time, MTC device $Mi - j$ can authenticate the HN by the unique GTK_{Gi}. Moreover, a mutual authentication between MTC device and its SN is out of our consideration. As UMTS-AKA protocol follows 'good enough security' principle, representing the balance established among cost-efficiency, security and usability. Despite this, our proposal archives mutual authentication, such as Section 3.3.

Security of Key In key generation process, the keys are computed by an MTC device and its SN respectively, and without being transmitted over any communication channels. On the other hand, the freshness of dedicated keys is assured by its key lifetime or change of AV. The devices in same group have different dedicated keys, for the random number is participate in key generation algorithm respectively.

Replay Attack Resistance Two types of random numbers are used in our protocol, generated by MTC device and by home enviroment. Even if a attacker gets a random number in a authentication procedure, it is infeasible to fake a challenge messages by reusing the random number. Moreover, MTC device and HE maintain an identical initial value to keep them synchronized during authentication process. An unsynchronized situation will lead to authentication failure. Thus, DGBAKA could prevent replay attacks in MTC communication.

Security against Fraud Attack In our protocol, all the MTC devices of a group share a GTK. Suppose that one MTC device intends to impersonate another device in the same group, even existing eavesdropping traffic between the two, the one can not generate a correct MAC_{G1} to impersonate another one, for lacking of IV and RAND of another one.

V. CONCLUSION

In summary, we provide a group based authentication and key agreement for MTC communication in roaming scenario. It avoids congestion and overload caused by masses of co-located MTC devices. And we present methods of group key generation and update for dynamic MTC group. Our construction has a good compatibility with current 3GPP system, thus it is more practical. For future work, we plan to further study the proposed protocol to support dynamic selection of group header and adaptive of group size, and reduce the complexity of MTC group update.

VI. ACKNOWLEDGMENT

This work is partially supported by Natural Science Foundation of China (61102056), the 111 Project of China (B08038).

REFERENCES

[1] 3GPP TS 22.368, *Service requirements for Machine-Type Communications (MTC)*; Stage 1(Release 11), v11.4.0. 3GPP, 2012.

[2] *Overview of 3GPP Release 11* V0.1.0. 3GPP, 2012.

[3] 3GPP TR 23.888, *System improvements for Machine-Type Communications (MTC)* , v1.6.1. 3GPP, 2012.

[4] 3GPP TR 33.102, *3G security; Security architecture*, v11.2.0. 3GPP, 2012.

[5] C. M. Huang and 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), pp. 392-397, 2005.

[6] K. K. Oh, T. Y. Lee, C. S. Nam and D. R. Shin, *Strong Authentication and Key Agreement Protocol in UMTS*, Fifth International Joint Conference on INC, IMS and IDC. pp. 917-920, 2009.

[7] 3GPP TS 33.401, *3GPP System Architecture Evolution (SAE); Security architecture*, v11.3.0. 3GPP, 2012.

[8] N. Aboudagga, J. J. Quisquater and M. Eltoweissy, *Group Authentication Protocol for Mobile Networks*, In Proceedings of the Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, pp.28, 2007.

[9] H. H. Ngo, X. P. Wu, P. D. Le and B. Srinivasan, *An Individual and Group Authentication Model for Wireless Network Services*, Journal of Convergence Information Technology, Vol. 5, No.1, pp. 82-94, 2010.

[10] Y. W. Chen, J. T. Wang, K. H. Chi and C. C Tseng, *Group-Based Authentication and Key Agreement*, Wireless Personal Communications, Vol. 62, No. 4, pp. 965-979, 2012.

[11] 3GPP TR 33.868, *Security aspects of Machine-Type Communications*, v0.7.0. 3GPP, 2012.

[12] RFC 2627, *Key Management for Multicast: Issues and Architectures*, IETF, 1999.