International Journal of Electrical and Computer Engineering (IJECE)
Vol. 8, No. 2, April 2018, pp. 1169~1181
ISSN: 2088-8708, DOI: 10.11591/ijece.v8i2.pp1169-1181

A New Scheme of Group-based AKA for Machine Type Communication over LTE Networks

Mariya Ouaissa[1], A. Rhattoy[2]
[1]ISIC, High School of Technology, LMMI Laboratory, ENSAM, Moulay-Ismail University, Meknes, Morocco
[2]ISIC, Department of Computer Engineering, High School of Technology, Moulay-Ismail University, Meknes, Morocco

Article Info

Article history:

Received Jun 29, 2017
Revised Dec 16, 2017
Accepted Dec 23, 2017

Keyword:

Authentication
LTE/SAE
M2M/MTC
Networks
Security

ABSTRACT

Machine Type Communication (MTC) is considered as one of the most important approaches to the future of mobile communication has attracted more and more attention. To reach the safety of MTC, applications in networks must meet the low power consumption requirements of devices and mass transmission device. When a large number of MTC devices get connected to the network, each MTC device must implement an independent access authentication process according to the 3GPP standard, which will cause serious traffic congestion in the Long Term Evolution (LTE) network. In this article, we propose a new group access authentication scheme, by which a huge number of MTC devices can be simultaneously authenticated by the network and establish an independent session key with the network respectively. Experimental results show that the proposed scheme can achieve robust security and avoid signaling overload on LTE networks.

Corresponding Author:

Mariya Ouaissa,
ISIC,
High School of Technology,
LMMI Laboratory, ENSAM,
Moulay-Ismail University,
Meknes, Morocco.
Email: mariya.ouaissa@edu.umi.ac.ma

1. INTRODUCTION

Machine-to-Machine (M2M) enables tens of billions of machines in the world to talk to each other about their surrounding environment through wired or wireless connections. Many applications in the categories of monitoring, maintenance and safety can be considered as emerging M2M communications. The Standard 3rd Generation Partnership Project (3GPP) has become a solution to facilitate M2M communication.It has become known as Machine Type Communication (MTC) [1].

In these days, the mobile wireless communication is widely used in human communications such as voice call, messaging and Web browsing. However, these types of services and new types of services and technologies are available on request. Of all these, MTC is the most important issue in the fields of standardization and industry [2].

Many standards exist from the level components, speaking different radio interfaces, different choice routed or mesh networks, or offer a choice of identity systems. However, little effort has been made to focus on the security aspects. It can be found that such an incident in the authentication and key agreement (AKA) procedure in a fourth-generation (4G) cellular network when a device registered to the core network. The AKA procedure is required each time a device is attached to exchange one of the nearby access points networks. This attachment and change can occur at any time. When a group of devices attempts to register simultaneously, signaling traffic associated generate a significant overload of the authentication server and create congestion in the link between a server and terminals [3], [4].

To avoid this problem, we emphasize the need for the design of effective AKA procedure that reduces repetitive invocation of costly authentication signaling, especially in group situations optimization [5].

The goal of this article is to take significant steps towards a new security feature that reduces the amount of signaling traffic in the AKA phase even if the number of MTC devices is important and variable.

In this paper, we review related studies already done in this context. Then we present the LTE/MTC system architecture, and we describe the authentication protocol EPS-AKA. Furthermore, in order to secure an MTC communication, we propose a new protocol based on group authentication, then we discuss its important details (design, security and performance evaluation).

2. RELATED WORK

In the literature, few authentication protocols of group communications have been proposed. However, there are still no appropriate group authentication methods for MTC communications in 3GPP. On the other hand, several existing protocols for 3GPP networks access, like UMTS-AKA [6] and EPS-AKA [7] are not suitable for group authentication. They need to be modified to apply to the group authentication of MTC.

An alternative might be the grouping of devices and having a leader of the group represent the group to serving networks. The leader authenticates itself to the network on behalf of all the MTC devices. Once it is successful, the leader is entrusted with power over the end devices and authorized to authenticate the end devices locally without having each device access a remote authentication server.

In 2012, Chen et al. propose a security group authentication and key agreement protocol (G-AKA) for a group of mobile stations (MSs) roaming from the same home network (HN) to a serving network (SN) in this roaming scenario, because of the grouping model, this protocol can lessen communication costs on the network. However, it also cannot provide enough security and is vulnerable to redirection, man-in-the-middle attacks, etc [8].

The Dynamic Group Based and Key Agreement (DGBAKA) is a security protocol for authentication of a group of MTC devices in this roaming scenario. Because of the grouping model, this protocol optimizes the performance of authentication of group communications nevertheless, overall complexity of the system can be built to up to large as the number of MTC devices increases [9].

Cao et al. a proposed group-based access authentication scheme for MTC, that is based on an aggregate signature. An elected leader generates the aggregate signature and forwards it to the core network. The network can authenticate all the group members by verifying the aggregate signature and can establish distinct session keys with each member [10].

Secure and Efficient (SE-AKA) is a secure and efficient group authentication and key agreement protocol which can fit in with the entire group authentication scenarios in the Long Term Evolution (LTE) networks. SE-AKA can resist several existing attacks and provide enhanced user's privacy preservation and a group authentication mechanism which can efficiently authenticate devices in a group. This protocol employs Elliptic Curve Diffie-Hellman (ECDH) to realize forward secrecy and backward secrecy. It also adopts an asymmetric key cryptosystem to protect devices' privacy. ECDH and asymmetric cryptography may not be suitable for resource constraint MTC devices [11].

EAP-based Group Authentication (EG-AKA) proposed in [12], a group AKA protocol for LTE networks is for 3GPP MTC devices to access the core network over non-3GPP air interfaces. Overall delay of the current AKA for a single user takes long because of a round-trip delay to the backend of the authentication server in a core network. In order to improve this delay, EG-AKA is designed to reduce the number of accessing times to the authentication server.

The authors in [13] propose a new efficient security protocol for MTC using aggregation of many authentication requests into a single one . The design of this protocol is presented to be compatible with the current system by being composed of only symmetric cryptography. The security and performance of the new design are evaluated via formal verification and theoretical analysis.

In [14] authors propose a novel lightweight group authentication scheme for M2M (GLARM) under the 3GPP network architecture, which consists of two protocols that can achieve efficient and secure group authentication in the 3GPP access case and non-3GPP access case, respectively. The security analysis shows that the proposed scheme can achieve the security goals, and prevent the various security threats. Also, performance evaluation demonstrates its efficiency regarding computation complexity and communication overhead.

3. BACKGROUND

This section provides an overview of the main elements of the MTC architecture in LTE network and presents the EPS-AKA procedure used to realize mutual authentication between the user and the network.

3.1. Network Architecture

The Evolved Packet System (EPS) network supports data services, where services "circuit" migrate to "package" services. It provides IP connectivity between the (User Equipment) UE and the Packet Data Network (PDN) and provides support different radio access networks. In Figure 1, there are two types of traffic: the user traffic and traffic control (signaling).

The access network, called LTE or Evolved UTRAN (E-UTRAN), is composed of nodes Evolved NodeB (eNodeB). The EPC consists of several nodes which are: the Mobility Management Entity (MME), the Home Subscriber Server (HSS), the Serving Gateway (S-GW) and the PDN Gateway (PDN-GW). To understand the operation of the system, we give a description of each of these nodes [15].

As shown in Figure 1, an MTC user can use the services provided by one or more MTC servers to operate a large number of MTC devices. An MTC server is a server, which can communicate to the LTE network itself, and to MTC devices via the LTE network [16].

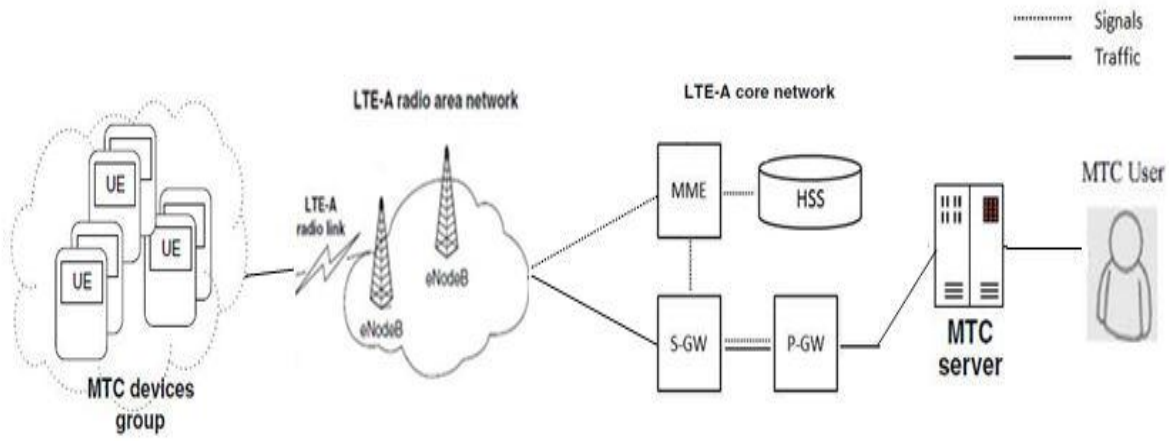


Figure 1. System architecture

3.2. LTE Authentication Protocol (EPS-AKA)

The 3GPP designed the security procedure EPS-AKA for the mutual authentication in LTE network and for securing the sharing of a cryptographic key. The EPS-AKA consists of seven messages illustrated in Figure 2.

The MME passes this first message to the HSS with the service network identity (SN id). If the IMSI is valid, the HSS generates and sends a set of several authentication vectors to the MME.A derived key (KASME) included in the authentication vector is a local key derived from K, which is a secret key shared between the UE and the HSS. The MME selects a vector authentication in the network and sends RAND [i] and AUTN [i] to MTC device to challenge the device authentication. The device authenticates the MME by checking the message authentication code. It then derives CK, AK, IK and KASME from K to respectively protect layers, Access Stratum (AS) and Non-Access Stratum (NAS) layers. Local Master Key, KASME, is valid for a period determined by the timing of the next EPS-AKA procedure. The device can choose to invoke the EPS-AKA protocol whenever the MME service changes, because of roaming to another network service [17].

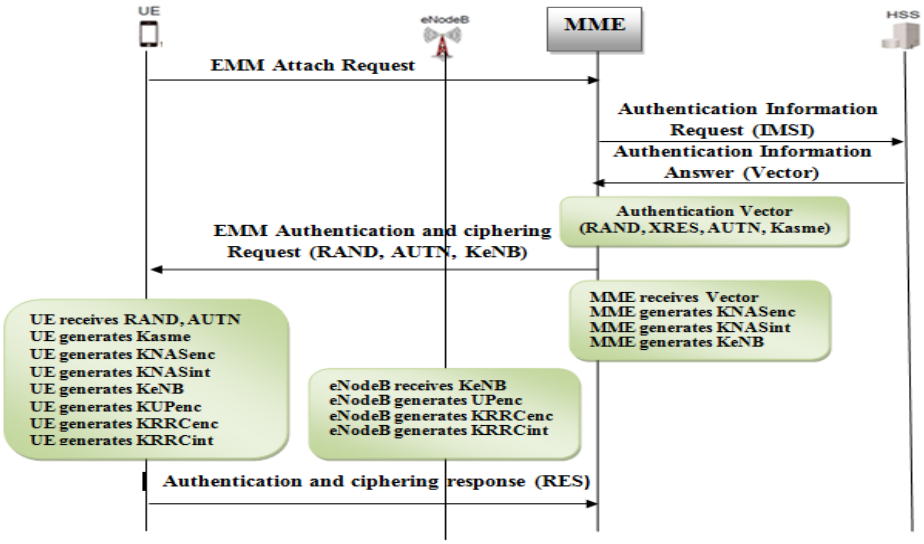


Figure 2. EPS-AKA procedure

3.3. Group Authentication Procedure

The authentication systems of existing networks are designed primarily for a single object, and they all need 3 or 4 rounds of interaction to achieve mutual authentication between a user and a server. But in practical applications, there may be a large number of users with the same properties in a network. Let's take a specific example of MTC, user terminals can form a group when they are in the same area, the same application or the same behavior. The group communication network model is illustrated in Figure 3.

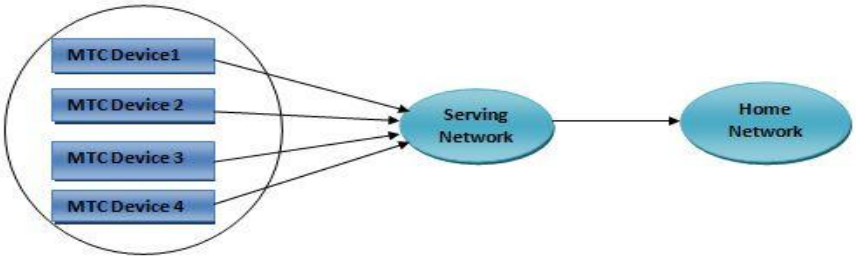


Figure 3. Network model of group communication

This paper presents a new method of group authentication based on an authentication and key agreement protocol. The main idea is the following: first, a leader MTC device is selected from a group, and a full authentication AKA procedure is performed. In this process, the leader MTC device obtains a group of authentication vectors and group authentication key (GAK) on behalf of other MTC Group devices [18].

Then the service network (MME) is allowed to perform mutual authentication with the rest of the group MTC devices using the authentication vector obtained GAK and the unattended remote core network (HSS). The authentication delay can be decreased as a whole, and the signaling overhead between the MME and the HSS is significantly reduced.

4. PROPOSED AUTHENTICATION PROTOCOL

In this section, we propose a new AKA protocol based group authentication for MTC communications in LTE networks, which overcomes the replay attack, Man in The Middle (MITM) attack, spoofing attacks and Denial of Service (DoS) attacks. This proposed solution follows strictly the framework of the 3GPP protocol EPS-AKA. Each MTC device and HSS share a secret authentication key Ks. There are three random numbers to secure the first phase of the protocol stored in MME and the UE (SRanD). It is an

accumulator number that is randomly generated. Each SRanD has a fixed size of 24 bits.The length of the parameters used in this procedure is presented in Table 1.

Our protocol consists of three phases: Initialization phase, Group authentication and key agreement phase and Group member state.

Table 1. Length of parameters

Parameters	Value (Bits)
IMSI/IMGI	128
SRanD1/SRanD2/SRanD3	24
K/Ks	128
Km	256
XRES/RES	128
AUTH/SQN	128
RAND	128
AMF/SQN	48
LAI	40
MAC	64
SNid/ID_{HSS}	20

4.1. Initialization Phase

The MTC devices form groups based on certain principles (e.g., belong to the same application, within the same region, etc.). The network service provider can group MTC devices together. Group membership may be changed at any time when the service provider adds MTC devices to groups or retires MTC devices. The service provider is responsible for determining and maintaining group memberships. Then each group for authentication has a group authentication key (GAK) and a group identity (IMGI).

A secure communication channel between the MME and the HSS has already been established (based on Diameter protocol [19]) and can provide security services to the transmitted data.

a. Each MTC device has an identity (IMSIi) that allows to the MTC device to register in a 3GPP network.
b. Each MTC device has a pre-shared secret key (Ks_i) with HSS when it is first registered in HSS.
c. HSS generates group key (GAK) for each group. Further, the service provider stores a set of parameters in the secure storage of the MTC device at the time of registration. These parameters are IMGI, GK, Ks.

The group is identified by the IMGI (International Mobile Group Identity) and a secret key Ks created by the HSS and shared between the HSS and each device.

A group of leaders is included among the MTC devices to represent the group to the core network will be selected based on the communication capability, storage status and battery status of each MTC device. These leaders are registered in the HSS and identified by their identity IMSIs. The leader itself is also an MTC device and hence needs to store the same parameters as a member. This process is described as follows.

4.2. Group Authentication and Key Agreement Phase

The session authentication and key agreement first occurs between a selected leader and the core network. Accordingly, a secure connection is established in the E-UTRAN, then all members of the group are authenticated with the core network on the secure link via the leader. Figure 4 shows the proposed MTC AKA composed of 11 messages.

M1: The MME sends a leader identity request to the MTC device

M2: The leader meets the MME with the identity of all members including IMSIi, IMGI group identity, and a random accumulator SRanD1.

M3: In response, MME registers (IMSIi, IMGI, and SRanD1) and sends another random accumulator SRanD2 to MTC device to check whether it is active or not.

M4: If the MTC device is active and is the legitimate device, it calculates SRanD3 and MAC_{MTC} as shown in (1) and (2) and delivers them to the MME besides the $IMSI_i$, IMGI, SN id and ID_{HSS}.

$$SRanD3 = f1Ks\ (SRanD1, SRanD2) \quad (1)$$

$$MAC_{MTC} = f2Ks\ (SRanD3, ID_{HSS}) \quad (2)$$

M5: HSS calculated GTK by using GAK, RAND and SN id by using (3).
On using the signature S_{MME}, the HSS should provide a level of authenticity to MME and a higher level of confidence. This signature may be considered in case the MTC device is roaming and not directly tied to its network.

$$GTK = f3GAK(RAND, SNid) \quad (3)$$

M6: HSS generates $AUTH_{HSS}$ by using the Identifier IDHSS and checks the validity of MME with consideration of SN id and calculates XAUTH to authenticate an individual device and four AV settings: RAND, AUTH, XRES, and Km, respectively using (4), (5), (6) and (7).
The master key Km is derived from the secret key Ks shared between the HSS and the leader.

$$AUTH_{HSS} = (ID_{HSS}, GTK) \quad (4)$$

$$AUTH = (SQN, AMF, MAC_{MME}, GAK) \quad (5)$$

$$XRES = f4Km\ (RAND) \quad (6)$$

$$Km = f5Ks\ (GTK, RAND) \quad (7)$$

M7: The MME performs mutual authentication with the leader MTC by generating $AUTH_{MME}$ as follows in (8) and (9).
MME sends the message to the leader, the members should have the right to listen to all previous messages. The leader checks if IMGI is the real group identity and authenticates the MME by checking MAC_{MME}. The same for the other MTC devices.

$$AUTH_{MME} = (RAND, MAC_{MME}) \quad (8)$$

$$MAC_{MME} = f2GTK\ (SQN, RAND) \quad (9)$$

M8: Then the leader calculates Km and prepares the RES response value and sends it to the MME.
The NAS security is responsible for the security of communication between leader and MME
M9: MME authenticates the leader by the validity of the equivalence of RES and XRES and sends a request for authentication of members to all MTC devices group.
M10: Each MTC device calculates its parameters K_{MTCi}, $AUTH_{MTCi}$, and MAC_{MTCi} as shown in (10), (11) and (12) respectively.

$$K_{MTCi} = KDF\ (Ks_i \oplus Km) \quad (10)$$

$$AUTH_{MTCi} = f6Ks_i\ (Rand_i) \quad (11)$$

$$MAC_{MTCi} = f2K_i\ (AUTH_{MTCi}) \quad (12)$$

M11: The leader sends the authentication response to the MME and this latter authenticates the MTC device by comparing $AUTH_{MTCi}$ and XAUTH.
The AKA procedure is successful, and the MME is ready to join the group.

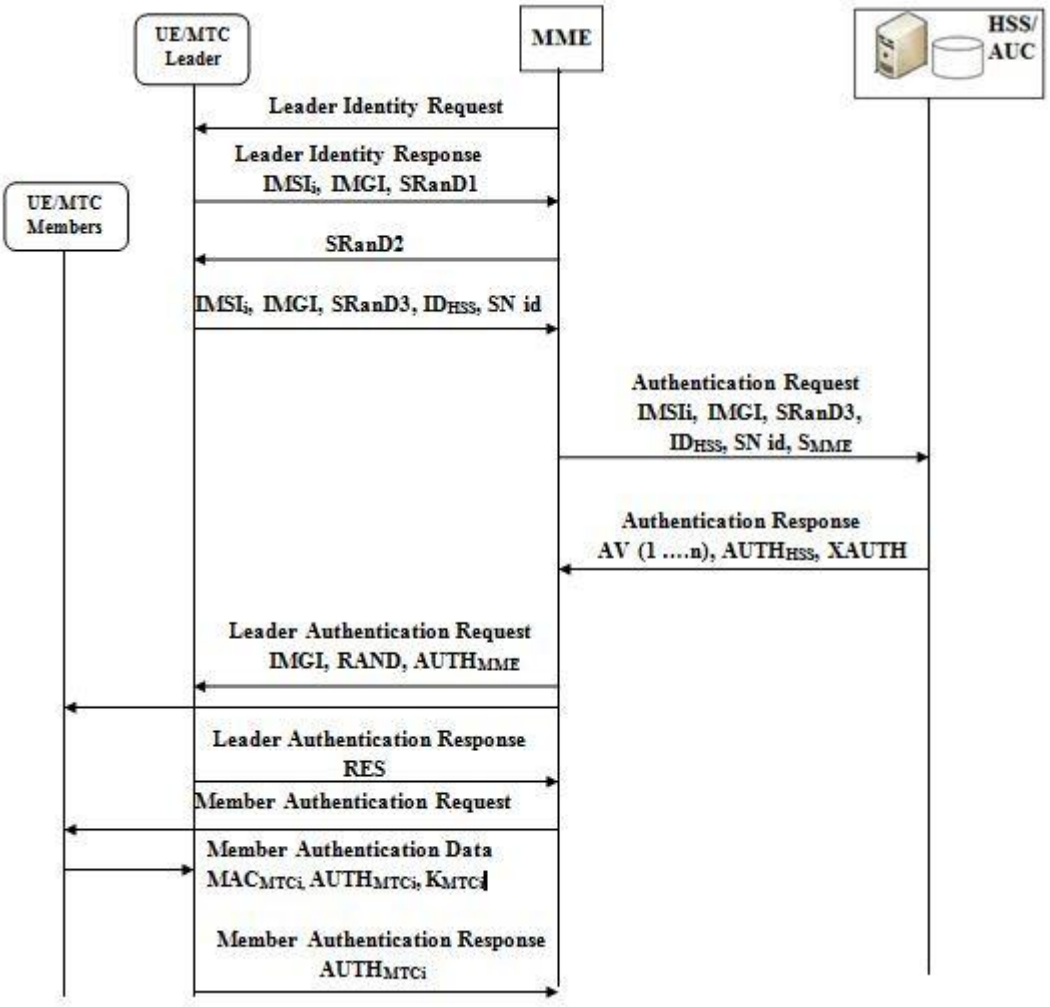


Figure 4. Proposed protocol for MTC

4.3. Group Member State

In our scheme, the group authentication key (GAK) can be used to authenticate HSS and MME. Therefore, when group members join or leave the group, the GAK need to be updated immediately since it will influence the security of the system. Moreover, if the GAK is used to encrypt group messages, the group which formed by MTC devices requires backward and forward secrecy. Backward secrecy is required that a new MTC device cannot get messages exchanged before it joined the group. Forward secrecy is required that a leaving or expelled MTC device cannot continue accessing the group's communication (if it keeps receiving the messages). When an MTC device wants to leave the group, the HSS will revoke the binding relationship between the MTC device and the group that it belongs to.Thus the MTC device cannot longer communicate with the core network as the group member. Moreover, to prevent the old MTC device to decrypt the new packets of the group which it was able to sniff, the group key must be updated when the old MTC device leaves the group. After the old MTC device leaves the group, all members of the group should share a new group key. Similarly, when an MTC device wants to join the group, an access control of the group is necessary for it, and it needs to perform a full AKA authentication procedure with the HSS. Meanwhile, the group key must be updated when the new MTC device wants to join a group. After the new MTC device joins the group, all members of the group should share a new group key. In that case, the new MTC device cannot decrypt the old packets of the group before it joins in

4.4. The Hierarchy of Keys in the MTC System

After successful authentication, each MTC device and SN (MME) shared a key Km as an essential tool for the derivation of the following keys. The hierarchy of keys in the MTC communication system is shown in Figure 5.

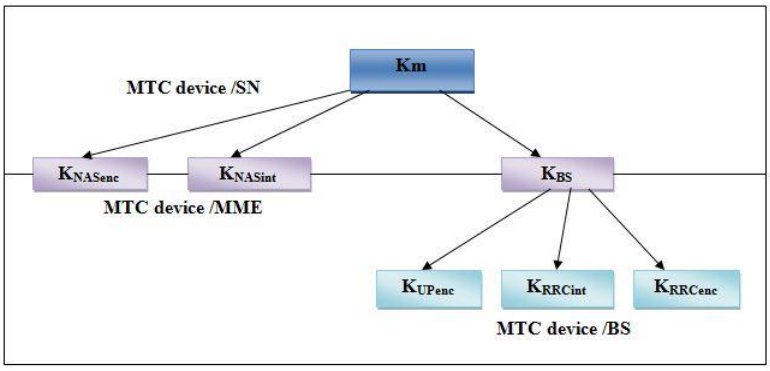


Figure 5. The hierarchy of keys

The function of each key is described as follows:
Km: primary key generated during the session LTE / EPC
K_{NAS}: NAS signaling security keys
K_{RRC}: Security key signaling radio RRC
K_{UP}: encryption key session data leaves key K_{NASenc}, K_{NASint}, K_{RRCenc}, K_{RRCint} and K_{UPenc} that serve as key encryption algorithms or integrity to protect the NAS signaling, AS and the user plane (data).

5. RESULTS AND DISCUSSION

In this section, we submit the protocol to a security analysis to show that it supports all security requirements required by M2M and explain how that solution is resistant to various security attacks such as MITM, DoS, redirection…, then we presente the validation of our protocol using AVISPA tool and compares the performance of our model with others research in literature. In addition, we evaluate the performance of the proposed group based AKA scheme regarding communication overhead and computational overhead.

5.1. Security Analysis

We analyze the security properties of the proposed scheme; both security analysis and formal verification are conducted to demonstrate that our approach can achieve all the security goals and requirements.

5.1.1. Mutual Authentication

The message authentication code MAC_{MME} = f2GTK (SQN, RAND) sent in M7 is created with the temporary key group, GTK. Because GTK is shared only among the members of HSS and group members, including the leader, can authenticate the MME if MAC_{MME} proves true.

For authentication in the other direction, the MME on behalf of HSS authenticates the leader by comparing the value of the RES with XRES. The leader can demonstrate his knowledge of the master key Km and also GTK by presenting the correct value of RES. The MME can authenticate each device MTC if AUTH matches XAUTH.

5.1.2. Resistance to Attacks

5.1.2.1. Replay Attack

Firstly, the protocol is free from this type of attack by sending random values (SRanD1, SRanD2, SRanD3) and by using a random timeout during the message transmission on the network In addition, wireless communication links between members, MTC device, and the MME can submit to this type of attack because the leader and the MME reach an agreement key on the session after the M8, an opponent could hear the four significant messages M2, M4, M7, M8 as shown in Figure 4. With this possession of these four messages, an adversary may attempt authentication pretending to be a leader and replay M8 to MME. However, this attempt would fail immediately because the value of RES recorded by listening differs from that of MME because of a new value of RAND in XRES =f4Km (RAND).

5.2.1.2. DoS Attack

The attacker MTC0 device floods the victim MME with the authentication request by usurping the IMSI and the SRanD1, and then an SRanD2 from MME is returned to the spoofed source MTC0. Therefore,

the MME will not get final information to complete authentication requests. This leads to a half-opened authentication requests to the MME. There is a waiting period for each MTC device to maintain the state of half-opened authentication requests. If the attacker MTC0 can create an overflow to the victim MME with half-opened authentication requests, the MME cannot accept new incoming authentication requests.

5.2.1.3. Man in the Middle Attack

An MITM attack can happen when a device attempts to connect to an eNodeB. For payload encryption in the proposed solution, a new key Km is a converse between UE and MME. This new key was introduced to make the most reliable communication between the UE and MME, this encryption key is introduced to overcome the MITM attack, and that is consulted by the UE and the MME in the last message of the protocol.

5.1.3. Formal Verification

This solution was checked by the security protocol verification tool, Automated Validation of Internet Security Protocols and Applications (AVISPA) [20], which indicated that it is a very secure level. The main advantage of this tool is the ability to use different verification techniques on the same protocol specification.

The protocol designer interacts with the tool by specifying a security problem in the High Level Protocol Specification Language (HLPSL). The HLPSL is an expressive, modular, role-based, formal language that is used to specify control-flow patterns, data-structures, alternative intruder models and complex security properties, as well as different cryptographic primitives and their algebraic properties [21].

The primary goal of our proposed protocol is to provide mutual AKA services between the MTC devices and the MME. We only need to verify that the proposed protocol can provide a successful mutual authentication between the MTC devices and the serving network.

In our proposed scheme described in High Level Protocol Specifications Language, the MME and MTC device represent the two participants in basic roles.

We need to verify that the proposed protocol can provide a successful mutual authentication between the MTC devices and the MME by using back-end servers.

In this paper, we only present the authentication analysis of one MTC device, basic roles of the MME and MTC device and the authentication goals are shown in Figure 6, Figure 7 and Figure 8, respectively.

The output of the model checking results are shown in Figures 9, we can conclude that the proposed scheme can accomplish the goal of mutual authentication and also can resist those malicious attacks, such as replay attacks, MITM attacks and secrecy attacks under the test of AVISPA.

```
role mme(MME,MTCD : agent,
         SND, RCV: channel(dy),
         GAK: symmetric_key,
         IMSI,IMGI,Sqn,SNID,AMF : text,
         F1,F2,F3,F4,F5: function)

  played_by MME
  def=

  local State : nat,
  SRanD1,SRanD2,SRanD3,RAND,IDhss : text
  const  mme_mtcd, mtcd_mme: protocol_id


  init  State := 1

  transition
    1.  State   = 1 /\ RCV(SRanD1'.IMSI.IMGI)
        =|>
        State' := 2 /\ SRanD1' := new()
                    /\ SRanD2' := new()
                    /\ SND(SRanD2')
                    /\ secret(IMSI,IMGI,SRanD1',SRanD2')
                    /\ witness(MME,MTCD,mme_mtcd,SRanD1',SRanD2')
    2.  State = 2   /\ RCV(IMSI.IMGI.SNID.IDhss'.F2(SRanD3'.IDhss').F1(SRanD1'.SRanD2'))
        =|>
        State' := 3 /\ secret(IMSI,IMGI,IDhss',SNID,SRanD3')
                    /\ witness(MME,MTCD,mme_mtcd,SRanD3')

    3.  State = 3   /\ SND(IMGI.RAND'.Sqn.AMF.F5(GAK.RAND').F3(GAK.Sqn.RAND'))
        =|>
        State' := 4
                    /\ RCV(F4(RAND').F5(GAK.RAND'))
                    /\ request(MME,MTCD,mtcd_mme,RAND)

end role
```

Figure 6. Role of MME

The back-end On-the-fly-Model-Checker (OFMC) will be used to verify that the proposed scheme maintains its security objectives even under various attacks. We run the Security Protocol Animator (SPAN) for AVISPA in OFMC mode to validate the above goals. The output of the model checking results is shown in Figure 9. According to this Figure, we can conclude that our scheme can achieve the security goals and withstand various attacks including MITM attacks, impersonation attacks, DoS and replay attacks under the test of AVISPA and SPAN using the OFMC back-end with a bounded number of sessions.

```
role mtcd (MTCD,MME : agent,
           SND, RCV: channel(dy),
           GAK: symmetric_key,
           IMSI,IMGI,Sqn,SNID,AMF : text,
           F1,F2,F3,F4,F5: function)

  played_by MTCD
  def=

  local State : nat,
  SRanD1,SRanD2,SRanD3,RAND,IDhss : text
  const
  mme_mtcd, mtcd_mme: protocol_id

  init  State := 1

  transition

    1.  State = 1 /\ RCV(start)
        =|>
        State':=2           /\ SRanD1' := new()
                            /\ SND(SRanD1'.IMSI.IMGI)
    2.  State = 2           /\ RCV(SRanD2)
        =|>
        State' := 3         /\ SRanD3' := new()
                            /\ IDhss' := new()
                            /\ SND(IMSI.IMGI.SNID.IDhss'.F2(SRanD3'.IDhss').F1(SRanD1'.SRanD2'))
    3.  State = 3           /\ RCV(IMGI.RAND'.Sqn.AMF.F5(GAK.RAND').F3(GAK.Sqn.RAND'))
         =|>
        State' := 4         /\ SND(F4(RAND').F5(GAK.RAND'))
                            /\ witness(MTCD,MME,mtcd_mme,RAND')
                            /\ request(MTCD,MME,mme_mtcd,RAND')
end role
```

Figure 7. Role of MTC device

```
goal

  authentication_on mme_mtcd
  authentication_on mtcd_mme

end goal
```

Figure 8. Analysis goals of our scheme

```
% OFMC
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/MTC_AKA.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.01s
  visitedNodes: 7 nodes
  depth: 3 plies
```

Figure 9. Results reported by the OFMC back-end in SPAN

5.1.4. Comparison of Security Protocols

Through Table 2 we have compared the security protocols performance of existing AKA protocols with those of our protocol, and to check the security level of the proposed solution, we have demonstrated that our protocol can provide the most comprehensive security performance by using the modeling of this protocol using AVISPA.

Table 2. Comparisons of security protocols

Vulnerability	EPS-AKA	Choi's Protocol	GLARM-1 GLARM-2	Cao-AKA	Proposed
Support group authentication	No	Yes	Yes	Yes	Yes
Type of cryptosystem	Symmetric	Symmetric	Symmetric	Asymmetric	Symmetric
Ensure confidentiality of IMSI	No	Yes	Yes	Yes	Yes
Resistance against redirection attack	No	Yes	Yes	No	Yes
Resistance against the DoS attack	No	No	Yes	No	Yes
Resistance against the blocking of services by an MITM	No	Yes	Yes	Yes	Yes
Resistance against attacks on the responses of authentication data	No	Yes	Yes	Yes	Yes
Resistance against the usurpation of identity of MME	No	Yes	Yes	Yes	Yes

5.2. Performance Evaluation

We evaluate the performance of the proposed scheme in terms of communication overhead and computational overhead.

5.2.1. Communication Overhead

The cost of communication is by definition the number of bits to complete many repetitive AKAs t for a number n of MTC devices. It is a function of n, t and the sum of the size of messages in an AKA procedure. In this part, we compare the communication overhead of the following protocols: EPS-AKA, Choi's Protocol, Cao-AKA and our proposed protocol for MTC.The measurments are based on the lenght of the parameters in Table 1.

In EPS-AKA, each device exchanges five messages to complete the proposed procedure. It needs this time $t \times n \times 5$ messages to n devices. The sum of the size of messages to a single procedure is 1364 bits. So the cost of communication is $1364 \times n \times t$.

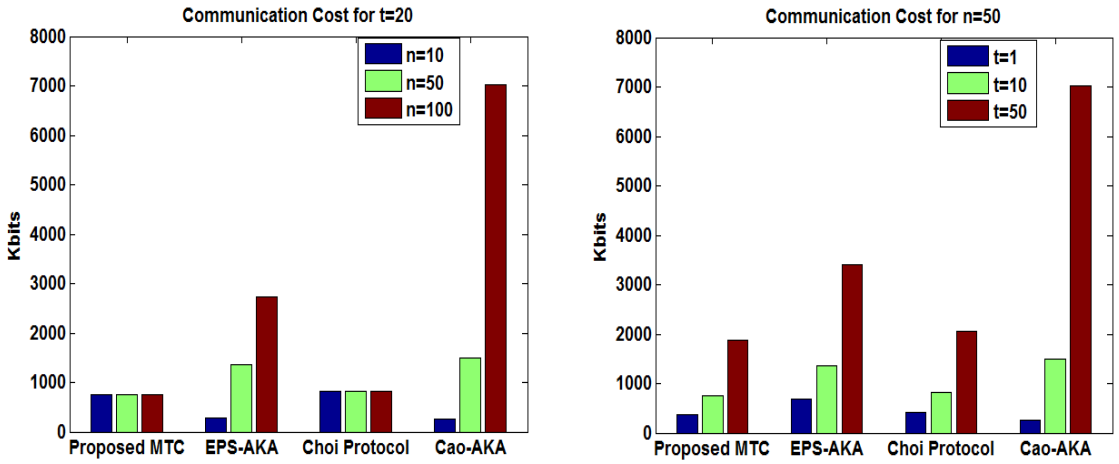
For Choi's Protocol, six messages are needed for the device to the MME and three messages in reverse to complete a simple exchange AKA, for the rest of (t - 1), an MTC device sending 6(t - 1) messages and receive 4(t - 1) messages for a total of message $n + (10 \times t) - 1$.

In the beginning, each MTC device in the Cao-AKA exchanges two messages with the key center and then executes the EPS-AKA. It takes 9n messages for n devices. In the rest period of (t – 1) times of the AKA, a group leader accepts a single message from n - 1 devices and exchanges another two messages with the core network. As many as $9n + (n + 1)(t - 1)$ messages are required to complete t times of the AKA for n devices.

The group leader in our proposed MTC-AKA exchanges the same number of seven messages as the EPS-AKA. The leader collects and processes the reply messages of (n - 1) devices and forwards them to the MME. When the MME can authenticate the leader and (n – 1) devices, the HSS creates an admission message and sends it to the group members. The total number of messages added to $n + 11 + 7 (t - 1) + 5 (t - 1) = n + (12 \times t) - 1$.

Figure 10(a) compares the communication cost of four protocols for a number of MTC devices, n = 1, 10 and 50 where the number of repetition is fixed at 20 (t = 20). t = 20. If the number of MTC devices is n=10 of the communication costs, the proposed AKA is the most expensive according to EPS-AKA and Cao-AKA protocols. As the number of devices increases, our approach demands less cost by benefiting of grouped requests. In this comparison, we can see that the proposed protocol based group authentication and key agreement is more efficient compared to other protocols.

Figure 10(b) shows a comparison of the communication cost for different numbers of repetitive AKAs, t = 1, 10 and 50, where the number of MTC devices is fixed at 50 (n = 50). The communication cost of the proposed is improved over Cao-AKA at t = 1. Note that the communication cost of the Cao-AKA is greatest after the first round of the AKA. After 50 repetitions of the AKA, the performance of Choi Protocol is second to the proposed AKA. In this comparison, the proposed AKA is the most efficient.



(a) (b)

Figure 10. (a) Comparison of communication costs for AKAs protocols when n = 1, 10, and 50, (b) Comparison of communication costs for AKAs protocols when t = 1, 10, and 50

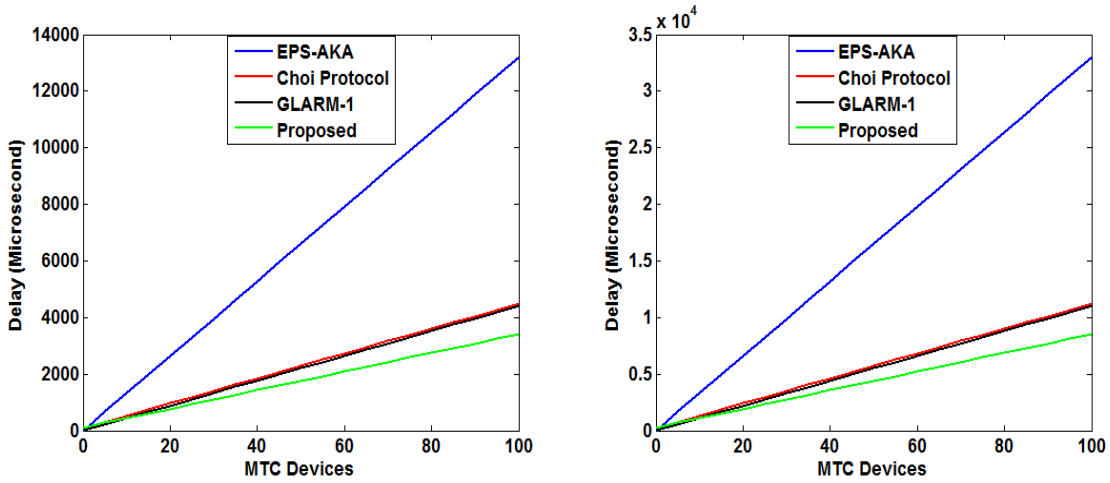
5.2.2. Computational Overhead

The computation overhead of proposed protocol is evaluated and compared to similar schemes. Associated calculations of the delays of many cryptographic operations in the process of message generation are due to calculations made at each of three network components: the device, MME and HSS. According to 3GPP, the functions f0, f1, f2, f3, f4 and f5 are HMAC-SHA256. For digital signatures, we will study the overhead associated with Digital Signature Algorithm (DSA), and for symmetric encryption we will use the Advanced Encryption Standard (AES) algorithm. Also, we analyzed these cryptographic operations in each message and summed the time of the operations for all messages that consist of the AKA as a way to measure and compare the computational delays of different protocols.

Delay values, available in [22], were obtained by measurements running on an AMD Opteron 8354 2.2 GHz processor under Linux.

We mainly consider the cost of the following operations, including a hash operation T_{hash} and an encryption operation T_{aes}. According to [22], T_{aes} takes 0.411 microseconds (μs) and T_{hash} takes 0.55 μs.

Table 3 displays equations of the computational delays demanded by the device and the core network. Based on these equations, we compared the computational delays of the following protocols in Figures 11(a) and 11(b) with different values of t.



(a) (b)

Figure 11. (a) Computations delays with t=20, (b) Computations delays wih t=50

Table 3. Time cost of cryptography operations

Protocol	Cost of Total Network (μs)
EPS-AKA	$12ntT_{hash}$
Choi's Protocol	$(4tn+8t)T_{hash}+tT_{aes}$
GLARM-1	$6tT_{hash}+4ntT_{hash}$
Proposed	$10tT_{hach} + 3ntT_{hash}$

6. CONCLUSION

In this article, we have presented the MTC architecture in LTE network and reviewed the main security protocols used in MTC networks to protect users from different types of attacks.

The 3GPP considers MTC as a significant sector in the LTE network for fourth-generation mobile communications. This paper has proposed a new protocol based group authentication appropriate for securing MTC communication. Extensive security analysis and formal verification have shown that the proposed MTC-AKA is secured against diverse malicious attacks. Thorough analysis and comprehensive evaluations with respect to communication overhead and computations delays confirm that the proposed MTC-AKA outperforms the existing AKA solution namely Choi's Protocol and other protocols.

REFERENCES

[1] J. Kim, *et al.*, "M2M service platforms: survey, issues, and enabling technologies," *IEEE Communications Surveys & Tutorials*, vol/issue: 16(1), pp. 61-76, 2014.

[2] A. Rghioui and A. Oumnad, " Internet of Things: Surveys for Measuring Human Activities from Everywhere", *International Journal of Electrical and Computer Engineering (IJECE)*, vol/issue 7(5), pp. 2474-2482, 2017.

[3] A. Kunz, *et al.*, "Machine Type Communications in 3GPP: From Release 10 to Release 12," *GLOBECOM workshops (GC Wkshps)*, pp. 1747-1752, 2012.

[4] V. Suryani, et al, "Trust-Based Privacy for Internet of Things", *International Journal of Electrical and Computer Engineering (IJECE)*, vol/issue 6(5), pp. 2396-2402, 2016.

[5] R. Shaik, et al, "Sufficient Authentication for Energy Consumption in Wireless Sensor Networks", *International Journal of Electrical and Computer Engineering (IJECE)*, vol/issue: 6(2), pp. 735-742, 2016.

[6] 3GPP TS 33.102 ver.11.5.1, "3G security: security architecture (release 11)." 2009.

[7] C. K. Han and H. K. Choi, "Security analysis of handover key management in 4G LTE/SAE networks," *IEEE Transactions on Mobile Computing*, vol/issue: 13(2), pp. 457-468, 2014

[8] Y. W Chen., *et al*., "Group-based authentication and key agreement," *Wireless Personal Communications*, vol/issue: 62(4), pp. 965-979, 2012.

[9] Y. Zhang, et al., *"Dynamic group based authentication protocol for machine type communications,"* IEEE International Conference on Intelligent Networking and Collaborative Systems (InCoS), 2012.

[10] J. Cao, et al., *"A group-based authentication and key agreement for MTC in LTE networks,"* Global Communications Conference (GLOBECOM),pp. 1017-1022, 2012.

[11] C. Lai, *et al*., "SE-AKA: A secure and efficient group authentication and key agreement protocol for LTE networks," *Computer Networks*, vol/issue: 57(17), pp. 3492-3510, 2013.

[12] R. Jiang, *et al.,* "EAP-based group authentication and key agreement protocol for machine-type communications," *International Journal of Distributed Sensor Networks*, 2013.

[13] D. Choi, *et al*., "A group-based security protocol for machine-type communications in LTE-advanced," *Wireless Networks*, vol/issue: 21(2), pp. 405-419, 2015.

[14] C. Lai, *et al*., "GLARM: Group-based lightweight authentication scheme for resource-constrained machine to machine communications,", *Computer Networks*, pp. 66–81, 2016.

[15] G. Fritze, "SAE–The Core Network for LTE," *Ericsson, Technical white paper*, 2008.

[16] T. Taleb and A. Kunz, "Machine type communications in 3GPP networks: potential, challenges, and solutions," *IEEE Communications Magazine*, vol/issue: 50(3), pp. 178-184, 2012.

[17] 3GPP TS 35.206 V11.0.0, "Technical Specification; 3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*," Document 2: Algorithm Specification (Release 11), 2012

[18] C. Lai, *et al*, "Security issues on machine to machine communications," *KSII Transactions on Internet and Information Systems (TIIS)*, vol/issue: 6(2), pp. 498-514, 2012.

[19] DIAMETER et ses Applications Principes, Architecture et Services , EFFORT http://www.efort.com

[20] AVISPA Project: http://www.avispa-project.org/

[21] Y. Ben Slimane and K. Ben Ahmed, " Efficient End-to-End Secure Key Management Protocol for Internet of Things", *International Journal of Electrical and Computer Engineering (IJECE)"* vol./issue 7(6), pp. 3622-3631, 2017.

[22] W. Dai, Crypto++ 5.6.0 Benchmarks, http://www.cryptopp.com/benchmarks.html,2009.