Indonesian Journal of Electrical Engineering and Computer Science
Vol. 16, No. 2, November 2019, pp. 848~859
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v16.i2.pp848-859

A secure group based authentication protocol for machine to machine communications in LTE-WLAN interworking architecture

Mariya Ouaissa[1], Abdallah Rhattoy[2]
[1]Research Team: ISIC High School of Technology, LMMI Laboratory ENSAM, Moulay-Ismail University, Morocco
[2]Research Team: ISIC, Department of Computer Engineering High School of Technology, Moulay-Ismail University, Morocco

Article Info

Article history:

Received Dec 20, 2018
Revised Mar 10, 2019
Accepted May 5, 2019

Keywords:

3GPP
Authentication
EAP-AKA
ECDH
LTE-WLAN
M2M
MTC
Security

ABSTRACT

Machine to Machine (M2M) communication has been used in applications such as telemetry, industry, automation and health. Support for a large number of devices has been considered an essential requirement in M2M communications. During this time, security is the most important challenge; M2M cannot access secure networks through effective authentication, all relevant M2M applications cannot be accepted. The challenge of M2M research is authentication by the group when a large number of M2M devices simultaneously accessing the network will cause severe authentication signaling congestion. The group based model under an M2M architecture, especially when the Machine Type Communication (MTC) devices belong to the non 3rd Generation Partnership Project (3GPP) network, will face a new challenge of access authentication. In this paper, we propose a group based authentication and key agreement protocol for machine type communications combining Elliptic Curve based Diffie-Hellman (ECDH) on the Extensible Authentication Protocol (EAP). Compared to EAP-AKA and other existing authentication protocols, our solution provides increased security against various malicious activities and better performance in terms of signaling overhead, bandwidth consumption and transmission cost.

Corresponding Author:

Mariya Ouaissa,
Information and Communication Systems Engineering Research Group, High School of Technology,
Mathematical Modeling and Computer Science Laboratory, Ecole Nationale Supérieure des Arts et Métiers,
Moulay-Ismail University, Km 5, Rue d'Agouray, N6, Meknès 50000, Morocco.
Email: mariya.ouaissa@edu.umi.ac.ma

1. INTRODUCTION

Machine to Machine (M2M) communication, also called Machine Type Communication (MTC), standardized by the 3rd Generation Partnership Project (3GPP), is an emerging technology for complete mechanical automation and its rapid development will change our life styles vigorously [1]. M2M technology is attracting attention in the areas of standardization and industry, to which many forums and standardization organizations have actively participated, including the Institute of Electrical and Electronics Engineers (IEEE), the European Telecommunications Standards Institute (ETSI), the 3GPP and 3GPP2. Among these, 3GPP has been considered MTC as the promising solution facilitating M2M communications. With the development of M2M technology, mobile network operators and research groups are paying more attention to efficiency, reliability and security requirements [2-3].

Security is of paramount importance in M2M communications [4], if M2M devices cannot securely access networks through efficient authentication, all applications involving M2M cannot be widely accepted. However, recent Authentication and Key Agreement (AKA) protocols dedicated to the 3GPP for Long Term

Evolution (LTE) or Evolved Packet System (EPS), known as EPS-AKA or for non-3GPP access networks, known as Extensible Authentication Protocol (EAP-AKA), cannot provide sufficient security. In addition, to support M2M communications, the 3GPP mobile operator must adapt its network to a large number of MTC Devices (MTCDs), which can overload its network resources and introduce congestion into the network on both data and control plans [5]. In fact, congestion can occur due to simultaneous authentication signaling messages from M2M devices [6-7]. If a large number of M2M devices in a group need to access the network at the same time, traditional authentication protocols (EPS-AKA or EAP-AKA) will suffer from a high signal overhead, leading to signaling, authentication and reduction of Quality of Service (QoS) of the network [8]. The reason is that each device must perform a complete AKA authentication procedure with the home authentication server, respectively. Given the reliability, these traditional AKA protocols are not suitable for large-scale M2M communications [9].

The main idea of proposed protocol in this paper is that the first MTC device in a group, which wants to access the 3GPP core network, performs a complete AKA authentication procedure. In this process, the first MTC device obtains group authentication information and a Group Temporary Key (GTK) on behalf of other MTC devices in the same group. Then, the Authentication, Authorization and Accounting (AAA) server is allowed to perform mutual authentication with the other MTC devices in the group using the obtained group authentication information and GTK without interacting with the Home Subscriber Server (HSS). The authentication delay can be reduced as a whole and the signaling overhead between the AAA server and the HSS is greatly reduced. The proposed AKA protocol for machine type communications combining Elliptic Curve based Diffie-Hellman (ECDH) [10] on the EAP protocol.

2. SYSTEM ARCHITECTURE

As shown in Figure 1, our considered network architecture is based on the 3GPP standard and can be divided into three areas [11-12] : Access networks, including the 3GPP access network, which are composed of Evolved NodeBs (eNBs) and non 3GPP access network such as a Wireless Local Area Network (WLAN), which consists of wireless Access Points (APs). Evolved Packet Core (EPC), including the Mobile Management Entity (MME) or AAA server that performs the access authentication function, the HSS, the Gateway Serving (S-GW) and data network Gateway Packet (P-GW). In our considered network architecture, the MTC server is located outside the EPC.

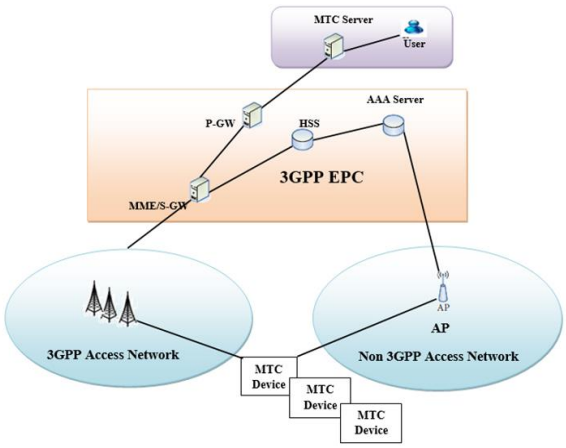

Figure 1. System architecture

3. THE PROPOSED GROUP AUTHENTICATION PROTOCOL

We present our proposed group authentication protocol for MTC [13], which has three phases: initialization phase, group authentication and key agreement phase and Group member state. Our scheme can facilitate non-3GPP MTC devices to access to 3GPP core network [14].

3.1. Group Initialization Phase

In the group initialization phase, each MTC device has an Identity (ID) such as International Mobile Subscriber Identification (IMSI), which is a private identity that identifies MTC device and should be installed in the MTC device by the supplier in order to allow the MTC device to register in a 3GPP network.

Each MTC device calculates its Temporary ID (TID) furthermore these MTC devices has preshared a secret key with HSS and the MTC devices form groups based on certain principles (e.g., belong to the same application, within the same region, etc.), then the supplier provides a Group Identity (ID_{G1}) and a Group Key (GKi) to each group for authentication. As shown in Table 1, we create an index table to manage information of MTC devices and group; the index table contains fields of group identity, MTC device ID for each MTC device.

Table 1. Temporary Index Table of G1

Group	Group ID	MTCD ID
G1	TID_{G1}	$TID_{MTCDG1-1}$
		.
		.
		$TID_{MTCDG1-n}$

3.2. Group Authentication and Key Agreement Phase

We assume that a secure communication channel between the AAA server and the HSS has already been established and can provide security services to the transmitted data. When non-3GPP MTC devices in G1 detect the AP, similarly, a group leader of MTCDs in the group ($MTCD_{LEADER}$) will be selected. Authentication and key agreement phase as follows:

Step 1: EAP request/identity

$MTCD_{G1-1}$ searches for the ID of the intended AP, when it finds the AP, the $MTCD_{G1-1}$ sends start message to start the authentication mechanism.

Step 2:

The AP requests the $MTCD_{G1-1}$ Identity.

Step 3:

Upon receiving the EAP Request/Identity message sent by AP, firstly, the $MTCD_{G1-1}$ computes $TID_{MTCDG1-1}$ et TID_{G1-1} respectively, and then $MTCD_{G1-1}$ generates $AUTH_{G1}$ as follows:

$$TID_{MTCDG1-1} = f^1_{KG1-1}(ID_{MTCDG1-1}) \quad (1)$$

$$TID_{G1-1} = f^1_{KG1-1}(ID_{G1}) \quad (2)$$

Each MTCD calculates its Message Authentication Codes (MAC) as following:

$$MAC_{MTCDG1-1} = f^2_{KG1-1}(ID_{MTCDG1-1}||ID_{G1}) \quad (3)$$

And each MTCD sends its MAC to $MTCD_{LEADER}$ who generates:

$$MAC_{G1} = MAC_{MTCDG1-1} \oplus MAC_{MTCDG1-2} \oplus MAC_{MTCDG1-3} \oplus \ldots\ldots\ldots \oplus MAC_{MTCDG1-n} \quad (4)$$

$$AUTH_{G1} = (MAC_{MTCDG1-1}||\ldots\ldots..||MAC_{MTCDG1-n}||MAC_{G1}) \quad (5)$$

Step 4: Group authentication and data request message

$MTCD_{G1-1}$ sends its parameters to the AAA server through AP, and then the AAA server finds out corresponding HSS according ID_{HSS} and forwards its parameters and its own ID_{AAA} to the HSS by authentication data request message.

Step 5: Group authentication data response

When the HSS receives authentication data request message, it verifies the identity of device and the received $MAC_{MTCDG1-1}$ and MAC_{G1}. Then HSS calculates a group temporary key GTK_{G1}.

HSS generates Group Authentication Vector (GAV) and calculate $AUTH_{HSS}$ and MAC_{HSS}:

$$GAV = (RAND||XRES||GTK||AUTHHSS) \quad (6)$$

$$GTK = f^3_{KG1}(AUTH_{HSS}) \quad (7)$$

$$AUTH_{HSS} = (R_{HSS}||ID_{HSS}||MAC_{HSS}) \quad (8)$$

$$MAC_{HSS} = f^2_{KG1}\ (ID_{HSS}||R_{HSS}) \quad (9)$$

Step 6:

The HSS sends ID_{AP}, GAV, K_{G1} to the AAA server by a pre-establish security channel.

Step 7: Group authentication request

The AAA server receives and stores ID_{AP}, GAV, and K_{G1} and calculates MAC_{AAA}, and generates $AUTH_{AAA}$. Also AAA generates a random value a and computes aP, and sends $AUTH_{AAA}$ and aP to device:

$$AUTH_{AAA} = (ID_{AAA}||R_{AAA}||MAC_{AAA}||MAC_{HSS}) \quad (10)$$

$$MAC_{AAA} = f^2_{KG1}(ID_{AAA}||R_{AAA}||R_{HSS}) \quad (11)$$

Step 8: Group authentication response

Firstly, $MTCD_{G1-1}$ computes:

$$GTK=f^3_{KG1}(R_{HSS}||ID_{HSS}) \quad (12)$$

Then $MTCD_{G1-1}$ verifies the received MAC_{HSS} and MAC_{AAA}:

$$MAC'_{HSS}=f^2_{KG1}(ID_{HSS}||R_{HSS}) \quad (13)$$

$$MAC'_{AAA}=f^2_{KG1}(ID_{AAA}||R_{AAA}||R_{HSS}) \quad (14)$$

If verification passes the device computes bP by generating a random value b, also computes its own Master Session Key (MSK) as EAP-AKA and RES_{G1}.

$$K_{MTCDG1-1}=f^4_{GTK}(abP) \quad (15)$$

$$RES_{G1}= f^5_{KMTCDG1-1}(ID_{G1}||ID_{MTCDG1-1}||R_{G1}) \quad (16)$$

And sends its to AAA.

Step 9: Authentication acknowledge

When the AAA server receives the group authentication response message, it compares RES_{G1} with $XRES_{G1}$. The AAA server sends ID_{AP} and MSK with EAP Success message to the AP. The AP verifies whether received ID_{AP} equals its own ID or not. If the result is incorrect, the AP drops the MSK and then terminates the execution. Otherwise the AP stores the MSK. Then AP encrypts ID_{AP} using the MSK and sends it with EAP Success message to $MTCD_{G1-1}$.

The full authentication and key agreement procedure for one MTC device is completed. The procedure is shown in Figure 2.

3.3. Group Member Joining/Leaving the Group

In our scheme, the group key can be used to authenticate HSS and MME. Therefore, when group members join or leave the group, the GK need to be updated immediately since it will influence the security of the system. Moreover, if the GK is used to encrypt group messages, the group which formed by MTC devices requires backward and forward secrecy. Backward secrecy is required that a new MTC device cannot get messages exchanged before it joined the group. Forward secrecy is required that a leaving or expelled MTC device cannot continue accessing the group's communication (if it keeps receiving the messages). When an MTC device wants to leave the group, the HSS will revoke the binding relationship between the MTC device and the group that it belongs to. Thus the MTC device cannot longer communicate with the core network as the group member. Moreover, 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.

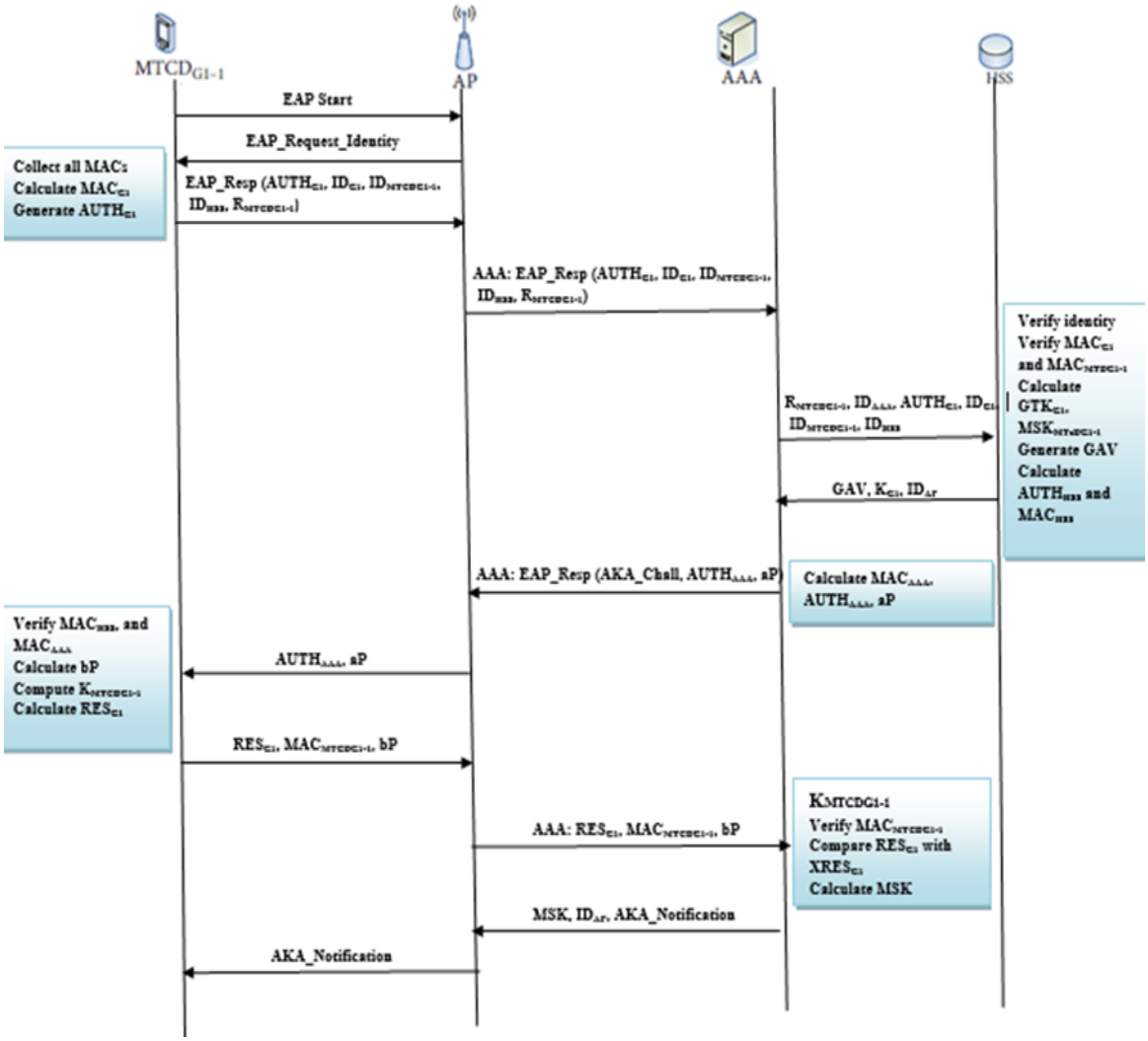

Figure 2. Authentication procedure of our proposed

4. SECURITY ANALYSIS

We analyze both formal verification implemented by the security protocol verification tool AVISPA and security analysis properties of the proposed scheme, to show that the proposed can achieves all the security goals [15].

4.1. Mutual Authentication and Key Agreement

Firstly, for the mutual authentication we can verify that the proposed protocol can provide a successful mutual authentication between MTC devices and the 3GPP Core Network (CN) by formal verification described as follows: all the MTC devices in the G1 first calculate their $MAC_{MTCDG1-1}$ and send them to the MTC leader. Then, the MTC leader collects all MAC, $MAC_{MTCDG1-1}$ and calculates MAC_{G1}. By verifying MAC_{G1}, the HSS can identify all MTC devices and the group. Then, the HSS calculates MAC_{HSS}, generates $AUTH_{HSS}$ and $XRES_{G1}$, and generates GAV for all MTC devices in G1. The HSS sends GAV containing $AUTH_{HSS}$ and $XRES_{G1}$ to the AAA; then, the AAA calculates MAC_{AAA} and generates $AUTH_{AAA}$, and sends them to all MTC devices in G1. By verifying $AUTH_{HSS}$/$AUTH_{AAA}$, all MTC devices can trust the HSS and the AAA. Also by verifying RES_{G1}, the AAA can authenticate all MTC devices in G1 [16].

For key agreement because all keys used among entities are computed without being transmitted over any insecure communication channels, the key agreement procedure is secure whether between the MTC device and the AAA server where can achieve through ECDH with symmetric key, and the MTC device and the AAA server can share a secret key $K_{MTCDG1-1}$ or between the MTC device and the AP where is the same as EAP-AKA and the MTC device and AP can securely communicate with other by the MSK.

4.2. Resistance to Attack

In this part we discribe various resistance methods to limit attacks in our model.

4.2.1 Resistance to Replay Attack

In our protocol, random numbers R_{HSS} generated by the HSS and R_{AAA} generated by the AAA server are temporarily used in generating challenge messages toward the opposite side, respectively. Since these random numbers used in each authentication procedure are different, even if an attacker acquires a random number in an authentication procedure, he still cannot fake challenge messages by reusing the random number in a new authentication procedure.

4.2.2 Resistance to Impersonate Attack

In our proposed, all the MTC devices of a group share a common GTK, the 3GPP CN can easily distinguish one MTC device from another even though all MTC devices use the same GTK, one MTC device cannot generate a correct parameters for another device MTC to perform a successful authentication with the HSS and cannot decrypt traffic between any other MTC device and the 3GPP CN.

4.2.3 Resistance to Denial of Service (DoS) attack

During the authentication for our protocol, a malicious MTC device may launch a DoS attack either to the HSS or to the AAA by sending an invalid MAC. However, the HSS can detect the invalid MAC and quickly re-authenticate the legitimate MTC devices in the group.

4.2.4 Resistance to Man-in-the Middle (MITM) Attack

In our proposed protocol, only the MTC devices and HSS can obtain real ID information of the devices and the group from encrypted temporary ID information. An attacker cannot derive and modify this information. The AP receives the EAP Success message with ID_{AP} and MSK sent by the AAA server. After that, the AP can verify whether its own ID equal to the received ID or not. If not the procedure of authentication and key agreement will fail. Our protocol can resist against several types of man-in-the middle attack.

4.3. Security Property

From our analysis and comparison, we derive the properties of our proposed protocol with others AKA protocols AKA protocols: EAP-AKA [17], Mun's protocol [18], EG-AKA [19] and GLARM-2 [20] protocol and show the results in Table 2. The comparison results demonstrate that our protocol can provide the most comprehensive security performance compared to the other AKA protocols. Providing group access authentication and heterogeneous network access are the two main advantages of our protocol. In particularly, our proposed protocol meets the following security properties.

Table 2. Comparisons of Properties among the Authentication and Key Agreement Protocols

Vulnerability	EAP-AKA	Mun's protocol	GLARM-2	EG-AKA	Proposed
Support group authentication	No	No	Yes	Yes	Yes
Type of cryptosystem	Symmetric	Symmetric and ECDH	Symmetric	Symmetric and ECDH	Symmetric and ECDH
Ensure confidentiality of user identity	No	Yes	Yes	Yes	Yes
Resistance against replay attack	Yes	Yes	Yes	Yes	Yes
Resistance against the DoS attack	No	No	Yes	Yes	Yes
Resistance against the blocking of services by an MITM	Yes	Yes	Yes	Yes	Yes

4.4. Formal Verification

This solution was checked by the security protocol verification tool AVISPA 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 AAA. We only need to verify that the proposed protocol can provide a successful mutual authentication between the MTC devices and the AAA server. We need to verify that the proposed protocol can provide a successful mutual authentication between the MTC devices and the AAA by using back-end

servers.In this paper, we only present the authentication analysis of one MTC device, basic roles of the AAA and MTC device and the authentication goals are shown in Figures 3, 4 and 5, respectively.

```
role mtcd (MTCD, AAA: agent, GK: symmetric_key,
F1,F2,F3,F4 : hash_func,
SND, RCV: channel (dy))
played_by MTCD
def=
local
State: nat,
Rmtcd, Rhss, Raaa, IDhss, Key: text,
Kma : message
const

   mtcd_aaa, aaa_mtcd  : protocol_id
init
State := 0
transition
1. State = 0
/\RCV (start)
=|>
State' := 2
/\Rmtcd' := new()
/\SND (Rmtcd')
2.  State = 2
/\RCV ({F2(Raaa'. Rhss'.Rmtcd)}_({F3(IDhss'. Rhss')}_GK).Raaa'. Rhss')
/\witness (MTCD, AAA, aaa_mtcd, Raaa', Rhss')
=|>
State' := 4
/\Key' := new()
/\ Kma':= {F4(Key')}_({F3(IDhss'. Rhss')}_GK)
/\ secret (Kma', {AAA, MTCD})
3.   State = 4
/\SND ({F1(Raaa. Key')}_Kma)
/\wrequest (MTCD, AAA, mtcd_aaa, Raaa)
=|>
State' := 6
end role
```

Figure 3. Role of MTCD

```
role aaa (MTCD, AAA: agent, GK: symmetric_key, F1,F2,F3,F4 : hash_func,
SND, RCV: channel (dy))
played_by AAA
def=
local
State: nat,
Rmtcd, Rhss, Raaa, IDhss, Key: text,
Kma : message
const
   mtcd_aaa, aaa_mtcd : protocol_id
init
State := 1
transition
1. State = 1
/\RCV (Rmtcd')
=|>
State' := 3
/\Raaa' := new ()
/\IDhss' := new()
/\Rhss' := new()
/\SND ({F2(Raaa'. Rhss'. Rmtcd')}_({F3(IDhss'. Rhss')}_GK). Raaa'.Rhss')
/\wrequest (AAA, MTCD, aaa_mtcd, Raaa',Rhss')
2. State = 3
/\RCV({F1(Raaa. Key')}_Kma')
/\witness (AAA, MTCD, mtcd_aaa, Raaa)
=|>
State' := 5
/\IDhss' := new()
/\Rhss' := new()
/\Kma':= {F4(Key')}_({F3(IDhss'. Rhss')}_GK)
/\secret (Kma',{AAA, MTCD})
end role
```

Figure 4. Role of AAA

```
goal

authentication_on mtcd_aaa
authentication_on aaa_mtcd

end goal
```

Figure 5. Analysis goals of the model.

We run the Security Protocol Animator (SPAN) for AVISPA in On-the-Fly-Model-Checker (OFMC) and SAT-based-Model-Checker (SATMC) modes to validate the above goals. The output of the model checking results is shown in Figures 6 and 7. According to this Figures, 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 and SATMC back-ends with a bounded number of sessions.

Figure 6. Results reported by the OFMC back-end

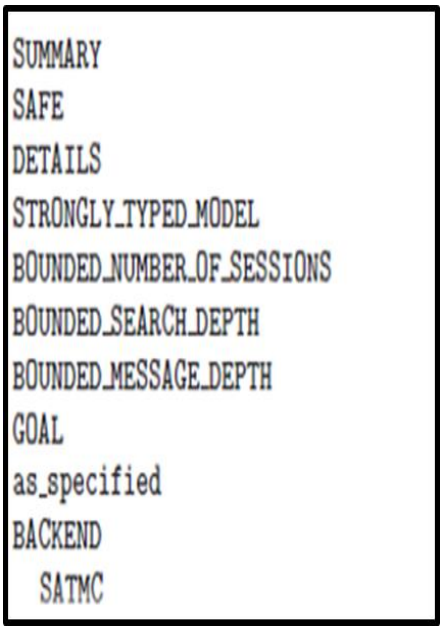

Figure 7. Results reported by the SATMC back-end

5. RESULTS AND DISCUSSION

In this section, we evaluate the performance of the proposed scheme in terms of signaling overhead, bandwidth consumption and transmission cost.

5.1. Signaling Overhead

In order to evaluate the signaling overhead, we assume that the number of MTC device is n, the number of group is m and the number of (re)authentications is x.

For EAP-AKA, authentication procedures performed by an MTC device require the total number of signaling messages, there are 12 signaling messages for one complete authentication procedure. Thus, the number of signaling message of a MTC device is $12x$ and the total number of signal message is $12nx$.

Mun's protocol is a new authentication and key agreement protocol based on EAP-AKA designed for 3G-WLAN interworking. This protocol combines ECDH with symmetric key cryptosystem to overcome several vulnerabilities. In addition, their protocol provides Perfect Forward Secrecy (PFS) to guarantee stronger security, mutual authentication, and resistance to replay attack. In this protocol, the MTC device runs a full authentication using 8 messages at one time, a total of $8x$ messages is required. Similarly, when n MTC devices belonging to m group perform authentication, so a total of Mun's protocol is $8nx$ messages.

Concerning EG-AKA, 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 first MTC device initiating authentication in the group complete the whole procedure of authentication and the number of signaling message is 8. The rest devices of the group only need 6 signaling messages For our proposed protocol, to complete a group authentication procedure the first MTC needs 7 message of signalization. The rest devices of the group only need 5 signaling messages. In this scenario, the number of the rest devices is $n - m$ and the total number of signaling message is $7m + 5(n - m)$. If each device executes another $x - 1$ re-authentications, then the total number of signaling message is $7m+ 5(n-m) + 5(x - 1)$.

Figure 8 illustrates the comparison of the number of signaling messages with the change of the number of MTC devices. According to this Figures it can be seen that our proposed protocol is much less than that of other existing schemes and outperforms this authentication protocols this is because our protocol shifts the impact of the number of MTC devices on network to the impact of that of the number of MTC device groups on network. The proposed scheme can largely reduce the authentication signaling overhead and relieve the charge of eNBs/APs and the MME/AAA server. Thus, our scheme can ensure QoS for MTC devices without restriction on the access requests.

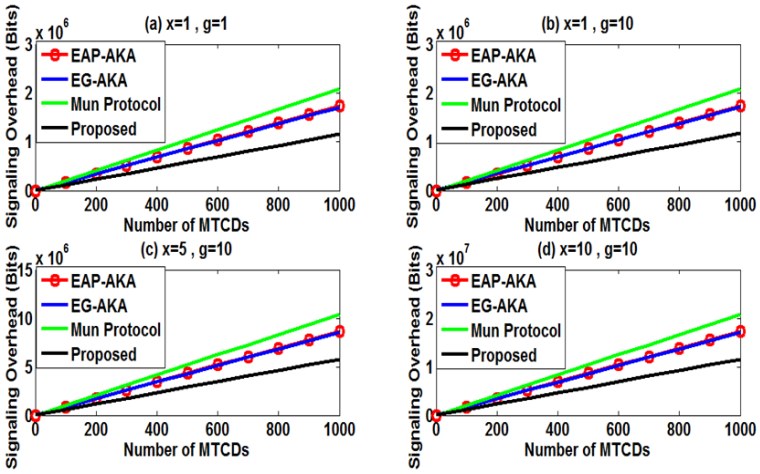

Figure 8. Comparison of signaling overhead

5.2. Bandwidth Consumption

In order to analyze the bandwidth consumption, we assume that x AVs are transmitted every time the HSS successfully authenticates one MTC device, and there are n MTC devices forming m groups. Table 3 shows the setting of parameters for evaluating bandwidth consumption.

Table 3. Setting of Parameters

Parameters	Value (bits)
TID/ID	128
GTK	128
XRES/RES	128
RAND/R	128
ECDH key	192
MAC	64

The bandwidth consumption of AKA protocols are as follows, where $bwfirst$ represents the bandwidth consumption of the authentication of the first MTCD. Bandwidth analysis of EAP-AKA: the sizes of authentication messages are calculated as follows:

$$bw_{first} = \sum_{i=1}^{5} |Message_i| = 704 + 608x \; bits \quad (17)$$

The overall bandwidth consumption for n devices is calculated as n × (704 + 608x). Bandwidth analysis of Mun's protocol: the sizes of authentication messages are calculated as follows:

$$bw_{first} = \sum_{i=1}^{7} |Message_i| = 2432 \; bits \quad (18)$$

The overall bandwidth consumption for n devices is calculated as n × 2432. Bandwidth analysis of EG-AKA: the sizes of authentication messages are calculated as follows:

$$bw_{first} = \sum_{i=1}^{6} |Message_i| = 2688 \; bits \quad (19)$$

$$bw_{remaining} = \sum_{i=1}^{3} |Message_i| = 1024 \; bits \quad (20)$$

Where bwremaining represents the bandwidth consumption of authentication of each remaining MTC devices.

The overall bandwidth consumption for n devices is calculated as m × 2688 + (n − m) × 1024. Bandwidth analysis of the proposed protocol: the sizes of authentication messages are calculated as follows:

$$bw_{first} = \sum_{i=1}^{6} |Message_i| = 2816 \; bits \quad (21)$$

Message1 = 3|ID| + |R| + |MAC| = 576 bits.
Message2 = 4|ID| + |R| + |MAC| = 704 bits.
Message3 = 2|R| + |XRES| + |GTK| + |ID| = 640 bits.
Message4 = |R| + |ECDH key| = 320 bits.
Message5 =|RES| + |ECDH key| = 320 bits.
Message6 = |ID| + |MSK| = 256 bits.

$$bw_{remaining} = \sum_{i=1}^{3} |Message_i| = 960\ bits \quad (22)$$

Where bwremaining represents the bandwidth consumption of authentication of each remaining MTC devices.
Message1 = |R| + |MAC| + |ECDH key| = 384 bits.
Message2 = |RES| + |ECDH key| = 320 bits.
Message3 = |ID| + |MSK| = 256 bits.
The overall bandwidth consumption for n devices is calculated as 2816 × m +960 $(n - m)$ bits.

We present the bandwidth consumption comparison of the proposed scheme and other relative AKA protocols. The comparison of bandwidth consumption is shown in Figure 9. From Figures, we can see that the bandwidth consumption of our proposed protocol is better than that of existing schemes.

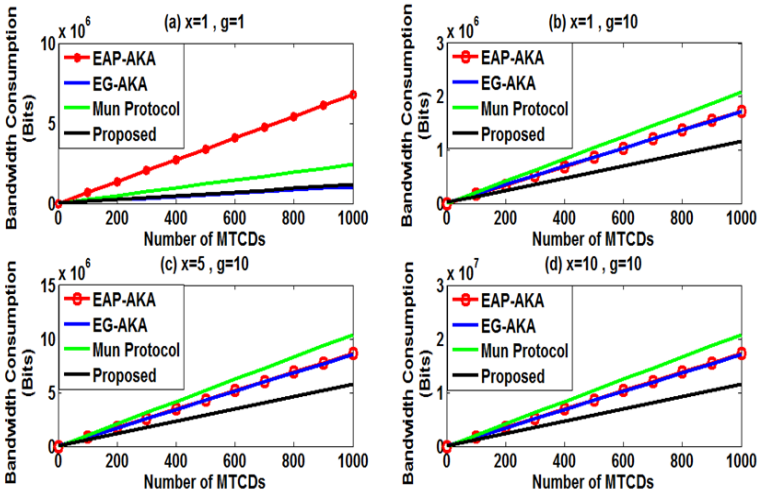

Figure 9. Comparison of bandwidth consumption

5.3. Computational Delays

We mainly consider the cost of the following operations cryptographic according to [22] including a point multiplication T_{mul} , a pairing operation T_{pair} and a map to point hash operation T_{mtp} and the hash operation T_{hash}. The cost of XOR can be negligible. Table 4 demonstrates the average elapsed time of five cryptographic operations.

Table 4. Time Cost of Cryptographic Operations used in Comparing Computational Delays

Operation	Symbol	Time (μs)
HMAC-SHA-256	T_{hash}	67
Multiplication over elliptic curve	T_{mul}	612
Addition over elliptic curve	T_{add}	125
MaptoPoint hash function	T_{mtp}	525
Pairing	T_{pair}	4514

The AAA and the MME in the core network and the devices execute many cryptographic operations in the process of message generation. We analyzed these cryptographic operations in each message and summed the elapsed time of the operations for all messages that consist of the AKA as a way to measure and compare the computational delays of the standard protocol EAP-AKA and our proposed.

Moreover, n represents the number of MTCDs in a group; m represents the number of groups. We present the computational delays comparison of the proposed scheme and other relative AKA protocols.

From the computational delays demanded by the total network, based on the following computations, we elaborately evaluate the computational delay for our proposed where the network has to compute all keys and authentication parameters, which takes $(8T_{hash}) \times n + 2T_{hash} \times m + 4\ T_{mul} \times n$. For EAP-AKA, the delay takes $12T_{hash} \times n \times m$.

We compared the computational delays of the proposed scheme and EAP-AKA protocol in Figure 10 with different values of m. From the figures, we can clearly see that the computational delays of our proposed protocol is much less than EAP-AKA when the number of groups take value bigger than 2. However, when the number of groups equal to 2 the computational delays of our solution is higher than EAP-AKA this is due to the addition of concept of group authentication in the proposed scheme.

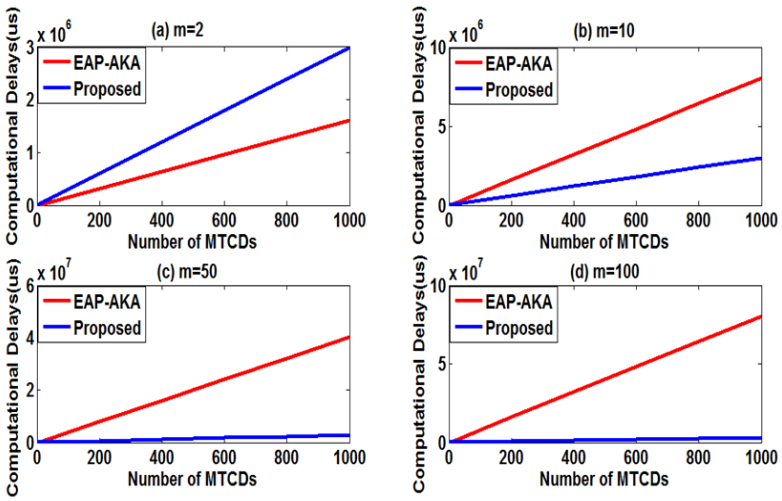

Figure 10. Computational delays of EAP-AKA and proposed protocols

6. CONCLUSION

In this paper, we have proposed a new group authentication and key agreement scheme for MTC communications under the EAP framework that supports the group authentication where the AAA server can authenticate a massive group of MTC devices in 3GPP network in the same group simultaneously. Formal verification and security analysis show that the proposed protocol can provide robust security and fulfill its design goals. In addition, the performance evaluation shows that our proposition is more efficient and achieves better performance than the existing schemes in terms of signaling overhead, bandwidth consumption and computational delay.

REFERENCES

[1] M. Ouaissa, A. Rhattoy, "A Secure Model for Machine to Machine Device Domain Based Group in a Smart City Architecture," *International Journal of Intelligent Engineering and Systems (IJIES)*, vol.12, no.1, 2019.

[2] A. Biral, *et al.*, "The challenges of M2M massive access in wireless cellular networks," *Digital Communications and Networks*, vol. 1, no. 1, pp. 1-19, 2015.

[3] M. Ouaissa, *et al.*, "New Method to Control Congestion for Machine to Machine Applications in Long Term Evolution System," *International Journal on Communications Antenna and Propagation (I.Re.C.A.P.)*, vol. 8, no. 4, 2018.

[4] A. Khanum and R. V, "An enhanced security alert system for smart home using IOT," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 13, no. 1, pp. 27-34, 2019.

[5] G. K. Sodhi, *et al.*, "Preserving Authenticity and Integrity of Distributed etworks through Novel Message Authentication Code," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 12, no. 3,pp. 1297-1304, 2018.

[6] 3GPP TS 36.300. V15.2.0, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (Release 15)," 2018.

[7] P. K. Verma, *et al.*, "Machine-to-Machine (M2M) communications : A survey," *Journal of Network and Computer Applications*, vol. 66, pp. 83-105, 2016.

[8] M. Ouaissa and A. Rhattoy, "New Method Based on Priority of Heterogeneous Traffic for Scheduling Techniques in M2M Communications over LTE Networks," *International Journal of Intelligent Engineering and Systems*, vol.11, no.6, 2018.
[9] D. Astely, *et al.*, "LTE release 12 and beyond," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 154-160, 2013.
[10] V. S.Miller, "*Use of elliptic curves in cryptography*," in Proceedings of the Advances in Cryptology (CRYPTO '85), Springer, pp. 417-426, 1986.
[11] 3GPP TS 33.401. V12.5.0, "3GPP System Architecture Evolution (SAE); Security architecture," 2018.
[12] M. Ouaissa, and A. Rhattoy, "QoS Hybrid Uplink Scheduler Based on Service Type for M2M Communications in LTE Networks," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 14, no. 3, 2019.
[13] M. Ouaissa, *et al.*, "*Group Access Authentication of Machine to Machine ommunications in LTE Networks*," in ICC '17 Proceedings of the Second International Conference on Internet of things and Cloud Computing, 2017.
[14] K. Ahmavaara, *et al.*, "Interworking architecture between 3GPP and WLAN systems," *IEEE Commun. Mag.*, vol. 41, no. 11, pp. 74-81, 2003.
[15] M. Zhang , and Y. Fang , "Security analysis and enhancements of 3GPP authentication and key agreement protocol," *IEEE Trans. Wirel. Commun*., vol. 4, no. 2, pp. 734-742, 2005.
[16] M. Ouaissa, and A. Rhattoy, "A New Scheme of Group-based AKA for Machine Type Communication over LTE Networks," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 2, pp. 1169-1181, 2018.
[17] RFC 4187, "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)," 2006.
[18] H. Mun, *et al.*, "*3G-WLAN interworking: security analysis and new authentication and key agreement based on EAP-AKA*," in Proceedings of the Wireless Telecommunications, 2009.
[19] R. Jiang, et al., "EAP-based group authentication and key agreement protocol for machine-type communications," *International Journal of Distributed Sensor Networks (Hindawi)*, 2013.
[20] C. Lai, *et al.*, "GLARM: Group-based lightweight authentication scheme for resource-constrained machine to machine communications," *Computer Networks*, pp. 66-81,2016.
[21] T. A. Team, "AVISPA v1. 1 User Manual 2006," [Online]. Available: avispaproject, http://avispaproject.org/.
[22] D. Choi, et al., "A group-based security protocol for machine-type communications in LTE-advanced," *Wireless Networks*, vol. 21, no. 2, pp. 405-419, 2015.