Research Article

International Journal of
Distributed
Sensor Networks

International Journal of Distributed Sensor Networks
2019, Vol. 15(4)
© The Author(s) 2019
DOI: 10.1177/1550147719841003
journals.sagepub.com/home/dsn
SAGE

Secret sharing-based authentication and key agreement protocol for machine-type communications

Ana Paula G Lopes[1], Lucas O Hilgert[1], Paulo RL Gondim[1] and Jaime Lloret[2]

Abstract

One of the main challenges for the development of the Internet of Things is the authentication of large numbers of devices/sensors, commonly served by massive machine-type communications, which jointly with long-term evolution has been considered one of the main foundations for the continued growth of Internet of Things connectivity and an important issue to be treated in the development of 5G networks. This article describes some protocols for the group-based authentication of devices/sensors in Internet of Things and presents a new group authentication protocol based on Shamir's secret and Lagrange interpolation formula. The new protocol protects privacy, avoids unauthorized access to information, and assists in the prevention of attacks, as replay, distributed denial of service, and man-in-the-middle. A security analysis and comparisons among the 3GPP evolved packet system authentication and key agreement standard protocol and other recent group authentication protocols were performed toward proving the efficiency of the proposed protocol. The comparisons regard security properties and computational and communication costs. The safety of the protocol was formally verified through simulations conducted by automated validation of internet security protocols and applications.

Keywords

Authentication and key agreement, machine-type communication, session key, Shamir secret, Lagrange interpolation

Date received: 5 June 2018; accepted: 4 March 2019

Handling Editor: Michele Amoretti

Introduction

The Internet of Things (IoT) has increased the production of daily-life devices and technological advances are leading to a type of communication defined as machine-type communication (MTC), in which at least one of the parties is a machine or sensor that requires no human intervention. MTC has been widely used for many applications related to IoT, among which the main ones are large-scale real-time applications, as

- Security (e.g. surveillance, control of physical access, home security);
- Tracking (e.g. fleet management, navigation, traffic information);
- Driverless autonomous transportation systems;
- Health (e.g. monitoring of vital signs, telemedicine, remote diagnoses);
- Metering (e.g. electric power, gas, water, heating);

[1]Department of Electrical Engineering, University of Brasilia (UnB), Brasilia, Brazil
[2]Integrated Management Coastal Research Institute, Universitat Politecnica de Valencia (UPV), Valencia, Spain

Corresponding author:
Jaime Lloret, Integrated Management Coastal Research Institute, Universitat Politecnica de Valencia (UPV), 46022 Valencia, Spain.
Email: jlloret@dcom.upv.es

- Remote maintenance/control (e.g. lighting, pumps, valves, vehicle diagnoses).

The MTC technology has been included as an important part of fifth-generation (5G) communications, based on a new radio technology able to treat massive machine-type communication (mMTC) and critical MTC (ultra reliable and low latency communication). Moreover, the addition of a MTC Server in the 3GPP architecture allows the development of applications in areas as health, transportation, environment, automation, and farming. MTC involves many categories of sensor-based applications, with billions of devices with small size data scattered worldwide and many signaling messages to be transmitted.

Support to such a massive number of MTC devices carries deep implications for the network architecture and its protocols. Among different options of access network for MTC, LTE/LTE-A (Long Term Evolution—Advanced) has been an important alternative for supporting MTC communications. It accommodates such a type of traffic, once it offers large coverage, high data rates, throughput, low latency, and mobility support. However, due to the provisioning of MTC services, a large signaling overload can occur in the network, which affects the provisioning of quality of service (QoS) for H2H (human-to-human) services.

The LTE/LTE-A radio area network is composed of mobile terminals, eNodeBs, and an LTE-A core network, named EPC (Evolved Packet Core), composed of components, such as HSS (Home Subscriber Server), MME (Mobile Management Entity), S-GW (Serving Gateway) and P-GW (Packet Data Network Gateway). In 3GPP, Releases 11, 12, and 13 of the LTE-A define the current and planned MTC features.

In an MTC architecture based on LTE/LTE-A, several MTC devices/sensors collect and send information to an MTC server, where it is analyzed. This MTC server, commonly located outside of the LTE-A network, stores the data collected by MTC devices; therefore, a new scenario involving the end-to-end connection among the MTCDs (mobile terminal communication devices), EPC, and MTC server must be treated.

Any communication in a public link can be a target for attacks, which highlights the importance of an efficient protection that imposes no inadequate bandwidth consumption (measured in number of bits sent over the communication channels). Computational resources must be carefully used and, since the process and resources involved in data collection and storage must be reliable, any decision-support process will depend on the confidence on the end-to-end network infrastructure.

This relevant scenario, based on the combination of MTCs complemented by a wireless wide area cellular network (LTE/LTE-A) and an MTC server, poses some security-related issues that must be adequately addressed. Below are some of such issues:

1. Support to a large number of MTC devices may cause signaling congestion, since the network may be overloaded with signaling from the authentication and control processes. Therefore, the repetition of costly authentication messages must be avoided.
2. An independent authentication process conducted by an MTD device will affect the radio access network (LTE) and the mobile core network (EPC) and cause high network access latency.
3. The number of bits sent on communication channels must be minimized, due to scarcity and the exponentially growing demand for voice and data traffic; moreover, cellular networks are commonly overloaded by H2H voice and data traffic.
4. An MTCD may show low processing capability and cause processing delays that might be incompatible with some applications of telemonitoring, tracking, and metering, for example. Therefore, the computational overhead imposed mainly to MTCDs must be reduced.
5. The current standardized AKA (Authentication and Key Agreement) protocol, known as EPS-AKA (Evolved Packet System Authentication and Key Agreement),[1] works in an individual basis and no group management scheme is provided. A full EPS-AKA[1] authentication procedure conducted for each MTCD imposes computational overhead and an authentication delay that hamper its practical use when a large number of devices requires authentication.
6. EPS-AKA[1] is vulnerable to several known attacks (e.g. man-in-the-middle (MITM) and denial of service (DoS)) and suffers from disclosure of user's identity in the first access to the network.
7. Users and network infrastructure may suffer from other typical threats and attacks (e.g. network impersonation, redirection, and replay attacks), which require security-related countermeasures for protection of integrity of data and preservation of MTCDs privacy.

The literature reports some protocols that enable group authentication, avoid congestion, and address safety toward circumventing such problems. However, they involve some security issues and their performance requires improvements.

Our solution involves a security-robust protocol that shows high performance for MTC in LTE/LTE-A network for the circumvention of the above-stated problems. A new authentication and key agreement protocol for congestion avoidance and better security has been designed; it consumes less bandwidth and fewer computational resources than other recent proposals. It is characterized by a mutual authentication and key agreement protocol, based on devices grouping, according to criteria, as same application type, localization, same MTC server, among others. Instead of authenticating each device separately, the network authenticates all in the MTC group simultaneously, reducing the signaling traffic. A leader has specific tasks for each group, which reduces the bandwidth consumption.

The protocol for MTC groups is based on Shamir's[2] secret and a binary tree group management, which guarantees security protection and improvements in the performance. It can also resist many attacks at low bandwidth consumption. It assumes a KGC (Key Generation Center) integrated with the HSS for avoiding the creation of a new component for key management. A session key is established between each MTCD and MME and two phases, namely registration (which uses asymmetric cryptography) and mutual authentication and key agreement (which uses symmetric cryptography) are considered.

The contributions of this article are as follows:

- The proposal of a group authentication protocol to avoid the disadvantages of EPS-AKA[1] protocol (standardized by 3GPP) that authenticates each device independently, generating high computational and communication costs and security issues.
- Computational cost reduction, due to the use of symmetric cryptography, when compared to group-based authentication and key agreement (GR-AKA),[3] also based on Shamir[2] and Harn,[4] and to other group authentication protocols as Lai et al.[5] and Choi et al.[6] Consequently, the main operations performed (hash, module, multiplication, and Lagrange component) have low cost, which reduces the processing time of the operations performed.
- Communication cost reduction, due to the use of symmetric cryptography, when compared to the above-mentioned protocols. Causing a diminishing in the size of exchanged parameters during the authentication procedure. Basically, only identities and hash are sent. Moreover, the amount of parameters exchanged is reduced.
- Protection against attacks, such as replay, DoS, MITM, redirection, and impersonation;
- Assurance of security properties, such as confidentiality, integrity, anonymity, forward, and backward secrecy;
- Formal validation of the protocol, using some Automated Validation of Internet Security Protocols and Applications (AVISPA)[7,8] backends, and a graphic simulation tool, which provides the visualization messages exchanged with or without the presence of an intruder.

The remainder of the paper is organized as follows: section "Related work" addresses some related and relevant studies; section "Proposed protocol" presents the protocol, which involves a registration phase and mutual authentication and key agreement; section "Security analysis of the protocol" reports on some security analyses and comparisons to other protocols; section "Performance evaluation" describes the performance evaluation that considered computation and communication costs; finally, conclusions and suggestions for future works are provided in section "Conclusion."

Related work

Security in group-based communication that considers sensor networks has been previously addressed, with proposals that lead to performance improvement.[9–11] In this study, we consider an extended scenario, where sensors (MTCD's), organized in groups, are connected to an MTC server by an LTE/LTE-A network for a broad range of applications, including e-health, smart metering, online school, and environment monitoring.

The development of group authentication has generated complex and robust protocols for MTC in LTE/LTE-A with higher security protection and better performance, which has brought innovations in the field.

A first contribution was provided by Harn,[4] who used Shamir's[2] Secret Sharing Scheme, a scheme based on polynomial and Lagrange interpolating formula. The protocol enables a group manager to generate a secret token, based on random polynomial, for each member of a group, where all tokens have a secret value in common. Therefore, all members can authenticate each other, reconstructing the secret value through the Lagrange interpolating formula. Only if all of them are legitimate, that is, all have legitimate tokens, the right secret will be reconstructed. Despite being an efficient group authentication protocol, it was not designed to be used for MTC in LTE/LTE-A; therefore, it does not consider the network architecture, security properties, and the higher performance required by MTC development.

Li et al.[3] developed a group authentication protocol based on Shamir's[2] secret and Harn's[4] group authentication scheme, called GR-AKA. Its architecture is similar to that of 3GPP EPS-AKA[1] and the difference is the MTC Server can be located inside or outside the

LTE architecture. Despite its key management efficiency, Li et al.[3] do not guarantee the anonymity of the MTC group and privacy in the device's identities from other devices in the same group. The group's identity is sent in plaintext and enables the attacker to track and identify the groups involved in the authentication procedure. The proposal uses asymmetric cryptography in the authentication phase, which requires higher consumption of computational resources in comparison to symmetric cryptography.

Lai et al.[5] proposed a protocol, called GLARM (group-based lightweight authentication scheme for resource constrained machine-to-machine communications), which is totally based on symmetric keys and hash functions and provides mutual and fast group authentication and key agreement. It consists of two phases, namely Initialization and Group Authentication and Key Agreement and its differential is the use of location area identification (LAI) of the base station involved in the authentication procedure to prevent attacks originated from intruder base stations. LAI identifies base stations in a unique way. The architecture is similar to that of 3GPP, as shown in Figure 1. Although it provides a fast group authentication, it requires high a consumption of communication resources, not desired in the development of MTC.

The protocol designed by Choi et al.[6] is based on symmetric cryptography and manages a group of devices through a binary tree, where each node is associated with a secret value derived from its parents. The tree provides an efficient and secure structure for the management of groups of devices, enables each device to be authenticated simultaneously with the group leader, and establishes different session keys between the MME and each device. The session key is based on the secret values of the common tree nodes between each device and the MME and on a random number generated by the HSS in the authentication procedure. However, regarding security, the protocol does not guarantee the anonymity of the MTC group.

The protocol created by Fu et al.[12] (privacy-AKA) is a privacy-preserving group authentication protocol based on ECDH (elliptic-curve Diffie–Hellman) key agreement. It performs secure and efficient mutual authentication and key agreement among groups of devices and a MME (Mobility Management Entity). The work preserves the privacy and anonymity of the devices by defining a set of pseudo identities, consequently protecting their permanent identities. Privacy-AKA is composed of two phases, namely initialization and mutual authentication.

Lai et al.[13] developed a group authentication protocol based on ECDH to perform the mutual authentication among groups of devices and a MME. The authentication phase is divided in two parts, one to authenticate the first MTCD to arrive in the server network and another to authenticate the rest of devices in the group. In the first part, it is necessary to involve the HSS in the authentication. The second part just involves MTCDs and the MME. The scheme does not select a group leader. Consequently, the first device to arrive in the server network might not be able to perform the important task of representing its group if its resources are limited.

The scheme of Gupta et al.[14] proposes a dynamic group authentication and key agreement protocol for MTC in LTE/LTE-A (group-based secure authentication and key agreement (GBS-AKA)), based on symmetric cryptography composed of four phases. The group organization is based on binary tree and a group leader is elected. The protocol calculates temporary identities preserve the privacy of each MTC device. To maintain the forward and backward security, the group key is updated each time a device joins or leaves the group. GBS-AKA has proven to be secure against

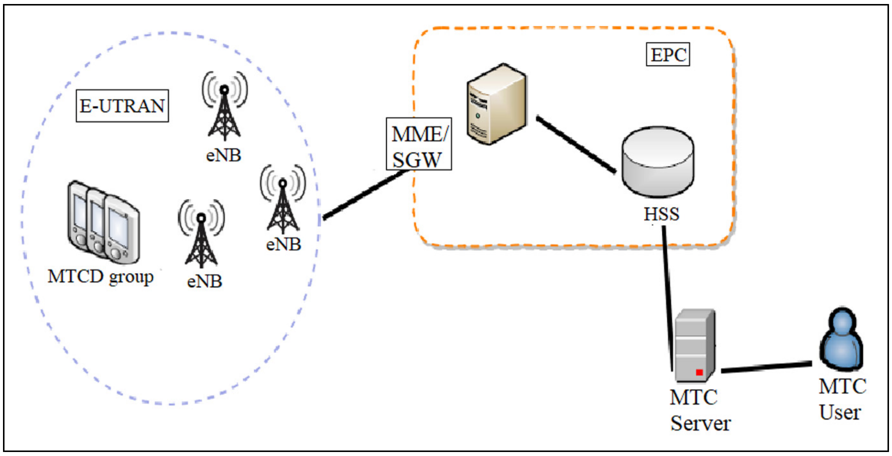


Figure 1. Network architecture of the proposed protocol.

several attacks; nevertheless, it presents high communication cost.

Parne et al.[15] proposed security enhanced group based authentication and key agreement (SEGB-AKA), a protocol for M2M communication in an IoT-enabled LTE/LTE-A network. The protocol is divided in four phases and is based on symmetric cryptography. Unique key identifiers are used to preserve the privacy of devices. A group leader is selected based on characteristics as battery life, storage capacity, and communication capability. The group management is based on binary tree. However, although it provides efficient and secure mutual authentication, the proposed protocol presents high communication and computational costs.

Asymmetric cryptography frequently imposes a higher cost than symmetric cryptography. Such an aspect was considered in our proposal toward reducing computational costs. Table 1 shows comparisons among the protocols regarding structure and techniques.

Proposed protocol

This section presents a new group authentication protocol based on symmetric cryptography, Shamir's secret sharing scheme and Lagrange's interpolating formula, that aims at secure and efficient authentication and key agreement for large groups of devices with good performance of authentication protocols. The Dolev–Yao model is adopted as the basis for the attack (adversary) model.

The network architecture, shown in Figure 1, is derived from 3GPP[1] standards. The following basic assumptions related to the entities involved were considered:

1. KGC is a trustful authority integrated with the HSS;
2. The channel between MME and HSS is secure;
3. The MTC server is located outside the EPC.

The group organization and management of MTC devices are based on the use of a binary tree for a group, which facilitates the group management and control of members.[6]

The protocol uses the Asynchronous (t; m; n) group authentication scheme (GAS) designed by Harn[4] to perform group authentication using Shamir's[2] scheme. The (t, m, n) GAS guarantees group authentication for m devices of a group with n members and is resistant to (t – 1) compromised tokens. The values of m and n are the same, that is, all members in a group are authenticated. Harn's scheme is suitable to our proposal because it quickly obtains one-time authentication for

Table 1. Comparison of authentication protocols.

	Schemes								
	EPS-AKA[1]	Lai et al.[5]	Li et al.[3]	Choi et al.[6]	Harn[4]	Fu et al.[12]	Lai et al.[13]	Gupta et al.[14]	Parne et al.[15]
Group authentication	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Type of cryptography	Symmetric	Symmetric	Asymmetric	Symmetric	–	Asymmetric	Asymmetric	Symmetric	Symmetric
MTC server	–	Inside EPC	Both	Outside EPC	–	Outside EPC	Inside EPC	Outside EPC	Both
Leader election	–	Yes	Yes	Yes	No	Yes	No	Yes	Yes
Structure for group management	–	Table	Not mentioned	Binary tree	Not mentioned	Not mentioned	Table	Tree	Tree
Shamir's secret	No	No	Yes	No	Yes	No	No	No	No
Location area identification (use of LAI)	No	Yes	No	No	No	No	Yes	Yes	Yes

EPS-AKA: evolved packet system authentication and key agreement; MTC: machine-type communication; EPC: evolved packet core; LAI: location area identification.

Table 2. Main entities involved in the architecture of the protocol.

Abbreviation	Entity
MTCD$_{i-j}$	Mobile terminal communication device *j* of group *i*
MTCD$_{leader}$	Mobile terminal communication device's group leader
HSS	Home subscriber server
MME	Mobile management entity
eNB	Evolved node B

MTC group without the presence of a managing entity, as HSS.

Table 2 shows the main entities involved in the architecture of the protocol and the notations and corresponding definitions are provided in Table 3. The design and operation of the protocol is composed of two phases, namely registration and mutual authentication and key agreement.

Registration phase

The registration phase establishes and configures all parameters necessary for MTCD groups to be authenticated by the network. It is divided into subphases (a–e).

Group definition and leader election. This phase begins with an initialization procedure that considers a scenario with n MTCDs arranged into m groups, each group with n/m members. The MTCDs form a group based on common characteristics and a group leader is elected. Some of the device's characteristics used for the group definition may be localization, type of application, and management by the same MTC server. The criteria used for the selection of the group leader may be higher storage capacity, longer battery, higher computational power, and higher communication capacity. The literature reports some processes for leader election,[16] which is outside the scope of this article. The phase occurs over a secure channel.

Creation of a binary tree. The HSS creates a binary tree, as described in Choi et al.,[6] for organizing each MTC group registered in the network. An identifier ID_{MTCDi} is assigned for each device and a set of temporary identifiers $TID_{MTCDi-j}$ is obtained in the sequence. Each device is placed in an empty leaf and each node of the tree has a secret defined by HSS. The devices know all the secrets, except those that form a path between the device and the root of the tree. The HSS defines all nodes' secrets and sends the tree to each member in the group with the secrets each one can know.

Table 3. Notations used in the protocol.

Notation	Definition
R_z	Random number z
Z_p	Prime field of order *p*
x	A secret value of HSS/KGC
ID_a, TID_a	Identity and temporary identity of entity *a*
LAI	Location area identification
n	Number of devices
m	Number of groups
$(t - 1)$	Number of compromised tokens the system is resistant
G_i	Group *i*, *i* = 1,2,3…
G	Random number *g*
P	Random prime number *p*
GK_i, GTK_i	Group key/group temporary key
GF	Finite field
MAC_a	Message authentication code of entity *a*
r_a	Random number generated by entity *a*
LC_a	Lagrange component of entity *a*
S	Shamir's secret between devices and MME
$f(x)$	Random polynomial function of degree $t - 1$
SEK_{i-j}	Secret key shared between MTCD$_{i-j}$ and HSS
SECy	Secret value of node *y*
$h_1(.)$	Secure hash function
$h_2(.)$	Message authentication hash function
$h_3(.)$	Key generation hash function
$h_4(.)$	Session key hash function
$H(.)$	Secure hash function
||	Concatenation operation
⊕	XOR operation
→	Secure channel
⇢	Insecure channel

HSS: home subscriber server; KGC: key generation center; MME: mobile management entity.

Generation of temporary identities of devices. HSS selects four hash functions h_1, h_2, h_3, and h_4, generates **z** random numbers, $\boldsymbol{R_z} \in \mathbf{Z}_\mathbf{p}^*$, $(z = 1, 2, \ldots, i)$, and calculates a set of temporary identities $TID_{MTCDi-j}$ for each $MTCD_{i-j}$, as follows

$$TID_z = h_1(ID_{MTDCDi}||R_z * x) \quad (1)$$

where *x* is a secret value only known by HSS/KGC.

The devices store each TID_z related to their respective R_z. A different TID is used whenever an authentication and key agreement procedure is conducted. In addition, the leader of the selected group receives the LAI of the base station that covers the group, which is important information for the authentication procedure.

Generation of group identities. The HSS defines a group identity ID_{Gi} and temporary group identity TID_{Gi}, generates a random number **g**, and calculates the group key, *GK*

$$GK_i = h_3\left(SEC_{i-1} \oplus SEC_{i-2} \oplus \cdots \oplus SEC_{i-j} \oplus g{*}x\right) \quad (2)$$

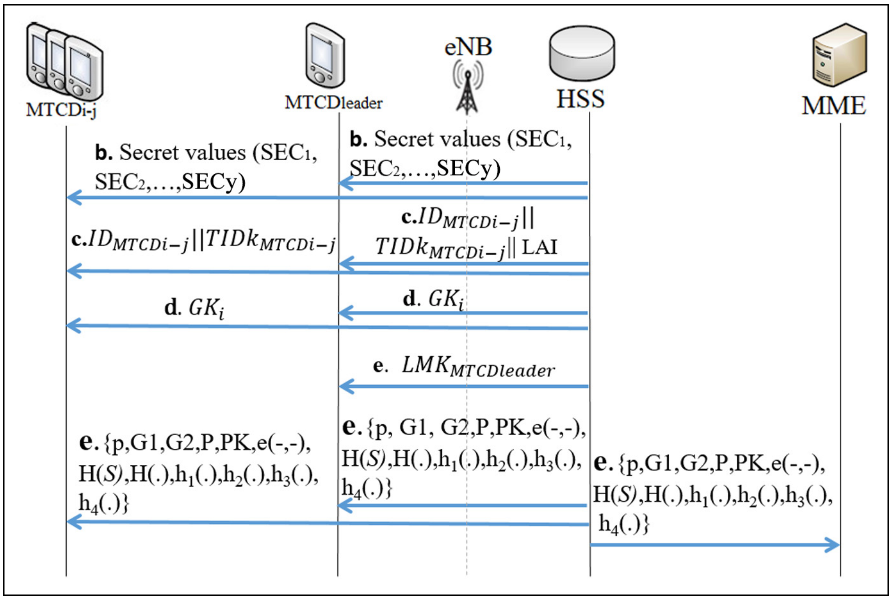

MTCDi-j
MTCDleader
eNB
HSS
MME
b. Secret values (SEC1, SEC2,…,SECy)
b. Secret values (SEC1, SEC2,…,SECy)
c.$ID_{MTCDi-j}||TIDk_{MTCDi-j}$
c.$ID_{MTCDi-j}||$ $TIDk_{MTCDi-j}||$ LAI
d. GK_i
d. GK_i
e. $LMK_{MTCDleader}$
e.{p,G1,G2,P,PK,e(-,-), H(*S*),H(.),h1(.),h2(.),h3(.), h4(.)}
e.{p, G1, G2,P,PK,e(-,-), H(*S*),H(.),h1(.),h2(.),h3(.), h4(.)}
e.{p,G1,G2,P,PK,e(-,-), H(*S*),H(.),h1(.),h2(.),h3(.), h4(.)}

Figure 2. Registration phase. Letters b, c, d, and e indicate the respective subphase in which the message is exchanged.

Generation of tokens and secret S. Below is the description of the generation of k tokens and secret S to be used in the authentication phase. KGC chooses a random prime number **p**, defines a finite field $GF(\mathbf{p})$, generates an authentication message $\boldsymbol{S}$, which is a secret parameter essential for group authentication, and selects a random polynomial function $f(x)$ of degree $t-1$ for each group, where $t \leqslant n$, representing the number of tokens necessary to recover secret S. The polynomial function is described as follows

$$f(x) = \sum_{i=0}^{t-1} a_i x^i \, mod\, p \tag{3}$$

and secret S is

$$S = f(0) = a_0 \tag{4}$$

$$S = \sum_{c=1}^{n} f(x_k) \prod_{q=1;q\neq c}^{n} \frac{-x_q}{x_c - x_q} mod\, p \tag{5}$$

All coefficients a_i are in the finite field $GF(p)$. KGC guarantees the condition is achieved and then generates **k** tokens $f(TID_{lMTCDi})$ for each device, where $l = 1, 2, \ldots, k$, and one token for each TID specific for a given device. The devices store their **k** tokens with the respective TIDs. The tokens must remain secret to any device that is outside the group and will be used in the authentication of the devices in the next phase.

Finally, KGC calculates the hash of secret S, $H(S)$, and hash function $H()$ to be used in the verification of the validity of all devices in the group. It also publishes the following parameters: $\{p, GF(p), P, H(S), H(.), h_1(.), h_2(.), h_3(.), h_4(.)\}$. The registration phase procedure is summarized in Figure 2.

Mutual authentication and key agreement phase

Once the registration phase has been successfully accomplished, the protocol proceeds as shown in Figure 3, with the following sequence.

Step 1.

$$MTCD_{i-j} \xrightarrow{(TID_{MTCDi-j})} MTCD_{i-j}$$

Each device chooses a non-used $TID_{MTCDi-j}$ with its respective associated token $f\left(TID_{MTCDi-j}\right)$ and broadcasts its own $TID_{MTCDi-j}$ to the other devices in the group, so that they can calculate their Lagrange component LC_{i-j}.

Step 2.

$$MTCD_{i-j} \xrightarrow{(LC_{MTCDi-j})} MTCD_{i-j}$$

Each $MTCD_{i-j}$ computes a Lagrange component, LC_{i-j}, using the selected token $f\left(TID_{MTCDi-j}\right)$ received from the KGC through the Lagrange interpolating formula

$$LC_{MTCDi-j} = f\left(TID_{MTCDi-j}\right) \prod_{q=1;q\neq i}^{\frac{n}{m}} \frac{-TID_{MTCDi-q}}{TID_{MTCDi-j} - TID_{MTCDi-q}} mod\, p \tag{6}$$

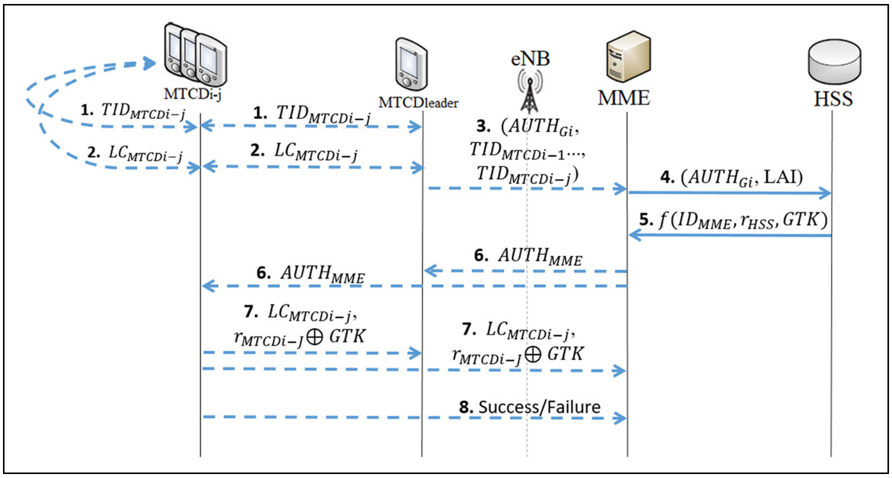

MTCDi-j
MTCDleader
eNB
MME
HSS
1. $TID_{MTCDi-j}$
1. $TID_{MTCDi-j}$
2. $LC_{MTCDi-j}$
2. $LC_{MTCDi-j}$
3. $(AUTH_{Gi}, TID_{MTCDi-1}, \ldots, TID_{MTCDi-j})$
4. $(AUTH_{Gi}$, LAI)
5. $f(ID_{MME}, r_{HSS}, GTK)$
6. $AUTH_{MME}$
6. $AUTH_{MME}$
7. $LC_{MTCDi-j}$, $r_{MTCDi-J} \oplus GTK$
7. $LC_{MTCDi-j}$, $r_{MTCDi-J} \oplus GTK$
8. Success/Failure

Figure 3. Authentication and key agreement phase of the protocol.

Each MTCD uses $TID_{MTCDi-j}$ received from the other devices in the group to generate a valid Lagrange component and broadcasts the respective $LC_{MTCDi-j}$ to all group members that authenticate themselves. After receiving the Lagrange components from the other group members, MTCDs check their legitimacy. If all of them are considered legitimate devices by each $MTCD_{i-j}$, the group is also legitimate. The verification is performed through the calculation of a secret S' and $H(S')$ and comparison of the value found with the value published by the KGC in the registration phase, $H(S)$

$$S' = \sum_{j=1}^{\frac{n}{m}} LC_{MTCDi-j} \, mod \, p \tag{7}$$

If $H(S') = H(S)$, all devices are validated and considered legitimate. If the verification fails, the group has one or more intruders and the process of authentication fails. The process continues only if all devices are legitimate and have been verified.

Step 3.

$$MTCD_{leader}\left(AUTH_{Gi}, TID_{MTCDi-1}, \ldots, TID_{MTCDi-j}\right)MME$$

$MTCD_{leader}$ generates the group's MAC_{Gi} and $AUTH_{Gi}$

$$\dashrightarrow$$

$$MAC_{Gi} = h_2(GK||ID_{Gi}||LAI||S') \tag{8}$$

$$AUTH_{Gi} = (TID_{Gi}||MAC_{Gi}) \tag{9}$$

MAC_{Gi} is based on GK and ID_{Gi}, which are parameters known only by valid members of the group, and on group secret S', which proves the group's legitimacy if S' is equal to the original secret S generated by the KGC in the registration phase. It is also based on LAI, which is an identifier related to the group's legit base station. $MTCD_{leader}$ sends $(AUTH_{Gi}||TID_{MTCDi-1}|| \ldots ||TID_{MTCDi-n})$ to MME.

Step 4.

$$MME \xrightarrow{\quad (AUTH_{Gi}, LAI) \quad} HSS$$

MME knows the LAI associated with the group and adds it to the message, so that HSS can verify if the LAI provided by the group leader is legit. MME stores each device's $TID_{MTCDi-j}$ for future use and sends $AUTH_{Gi}||LAI'$ to HSS.

Step 5.

$$HSS \xrightarrow{\quad (f(ID_{MME}), r_{HSS}, GTK) \quad} MME$$

After receiving the message from MME, HSS associates the group temporary identity, TID_{Gi} with its permanent identity, ID_{Gi}, and group key GK. It uses GK, ID_{Gi}, with LAI and S' received from MME to calculate MAC'_{Gi}

$$MAC'_{Gi} = h_2(GK||ID_{Gi}||LAI||S') \tag{10}$$

If MAC'_{Gi} calculated is equal to MAC_{Gi} received from MME, the MTCD group is authenticated by HSS. Otherwise, a failure message is sent to the $MTCD_{leader}$.

HSS chooses a random number r_{HSS} and generates temporary group key GTK

$$GTK_{Gi} = h_3(\mathrm{GK}||r_{HSS}) \tag{11}$$

It then calculates a token for MME, $f(ID_{MME})$ using MME's identity, ID_{MME}. The token will enable the devices to further authenticate MME.

Finally, HSS sends $f(ID_{MME})||GTK_{Gi}||r_{HSS}$ to MME.

Step 6.

$$\xrightarrow{MME \qquad (AUTH_{MME}) \qquad MTCD_{i-j}}$$

After receiving the message from HSS, MME generates a random number, r_{MME}, and conducts an XOR's operation of r_{MME} and GTK. Therefore, only the one that knows GTK will recover r_{MME}, that is, only a legitimate device can recover r_{MME}. Next, MME calculates its own Lagrange component and $AUTH_{MME}$

$$AUTH_{MME} = (LC_{MME}||r_{MME} \oplus GTK||r_{HSS}||ID_{MME}) \quad (12)$$

$$LC_{MME} = f(ID_{MME}) \prod_{q=1}^{\frac{n}{m}} \frac{-TID_{MTCDi-q}}{ID_{MME} - TID_{MTCDi-q}} mod\, p \quad (13)$$

Step 7.

$$\xrightarrow{MTCD_{i-j}(LC_{MTCDi-j}, r_{MTCDi-j} \oplus GTK)MME}$$

When each device has received the message from MME, they first update their Lagrange component with the MME's identity, ID_{MME}, as follows

$$LCnew_{MTCDi-j} = LC_{MTCDi-j} * \left(\frac{-ID_{MME}}{TID_{MTCDi-j} - ID_{MME}} \right) \quad (14)$$

Next, each device obtains r_{HSS} and calculates GTK

$$GTK_{Gi} = h_3(GK||r_{HSS}) \quad (15)$$

A new group temporary key is generated at each session. After updating the Lagrange component and calculating GTK, each MTCD recovers r_{MME} executing an XOR operation with GTK. Next, they choose a random number $r_{MTCDi-j}$ and perform an XOR operation with GTK to keep the value secret.

Finally, the devices broadcast the new Lagrange component and the random number, $LC_{MTCDi-j}||r_{MTCDi-j} \oplus GTK$ to all group members and MME.

Step 8.

$$\xrightarrow{MTCD_{i-j} \qquad \text{Success/Failure} \qquad MME}$$

When each device has received all the new Lagrange components from other group members, they can authenticate the MME, recalculating secret S with the Lagrange component of MME

$$S'' = \left(\sum_{j=1}^{\frac{n}{m}} LC_{MTCDi-j} + LC_{MME} \right) \text{mod}\, p \quad (16)$$

If S'' calculated is equal to S' previously calculated, MME is authenticated by the devices and each of them sends it a success message. If the verification fails, each device that has detected an authentication failure sends MME a failure message.

When MME receives its Lagrange components, LC_{i-j}, from each $MTCD_{i-j}$, it checks them calculating secret S'

$$S' = \left(\sum_{j=1}^{\frac{n}{m}} LC_{i-j} + LC_{MME} \right) \text{mod}\, p \quad (17)$$

If $H(S')$ is equal to $H(S)$ published by KGC, the devices are authenticated by MME and it sends a success message to the $MTCD_{i-j}$ group. If the verification fails, it sends them a failure message. Finally, the authentication procedure finishes.

If the mutual authentication procedure is successful, MME integrates the binary tree as a new element. Each $MTCD_{i-j}$ calculates a session key shared between them and MME. MME also calculates a session key shared between itself and each $MTCD_{i-j}$. The session key, $SK_{i-j-MME}$, is calculated as follows

$$\begin{aligned} SK_{i-j-MME} = h_4(&SEC_a \oplus SEC_b \oplus \ldots \oplus SEC_z \\ &||r_{MTCDi-j}||r_{MME}||S) \end{aligned} \quad (18)$$

where SEC_a, SEC_b,..., SEC_z are the secrets of the nodes each $MTCD_{i-j}$ and MME have in common. This model of session key is based on the binary tree presented in Choi et al.[6] and can be used for the device-to-device communication ($D2D$) among all M $MTCD_{i-j}$. A different session key is generated at the end of each session performed by the group.

Group secret and group key update

In our protocol, secret S and group key GK are important parameters, because the group authentication depends on them. A legitimate group will have a valid GK based on the members and must find the right S, with the components of each member, to obtain authentication. Therefore, the parameters must remain secret for the devices that do not integrate the current group. The scheme of secret update is based on Li et al.[3] and the group key update is based on Choi et al.[6]

Members joining/leaving the group

When an MTCD joins or leaves the group, secret S and group key GK must be updated, so that the old member does not continue knowing the secret parameters and new members do not discover the last secret values of S and GK. Such an update process occurs whenever the group's configuration has been altered.

Members joining. HSS creates a new leaf in the binary tree related to the new member and a new value secret, SEC_{i-y}, for the node. It also generates a new secret S as follows

$$S_{new} = S + \Delta S \tag{19}$$

where ΔS is a random value generated whenever secret S is updated. HSS sends new term ΔS and the secret value of new node SEC_{i-y} to MME, which encrypts ΔS and SEC_{i-y} with $SK_{i-j-MME}$ and sends them to each device of the group

$$En_{SK_{i-j-MME}}\left[\Delta S || SEC_{i-y}\right] \tag{20}$$

When all devices (including MME) have received the new secret and decrypted it with the session key, they update their tokens to equation (21)

$$f_{new}(TID) = f(TID) + \Delta S \tag{21}$$

and group key GK to equation (25)

$$GK_i' = h_3(GK_i \oplus SEC_{i-y}) \tag{22}$$

Members leaving. All members know the secret value of the node related to the member that has left the group $\boldsymbol{SEC}_{i-y}$; therefore, each member updates group key GK, as

$$GK_i'' = GK_i \oplus SEC_{i-y} \tag{23}$$

and secret S, as it occurs when a new member has joined the group.

The token each device has received from KGC is the result of a polynomial function $f(x)$

$$f(x) = \sum_{i=0}^{t-1} a_i x^i \, mod \, p \tag{24}$$

where the secret is a constant in the polynomial, $f(0) = a_0 = S$; therefore, all tokens have secret S as a constant in their composition. When each member has updated its own token with ΔS, they update secret S present in their token for a new secret S_{new}, as equation (25)

$$\begin{aligned} f_{new}(x) = \sum_{i=0}^{t-1} a_i x^i \, mod \, p + \Delta S = (S + \Delta S) \\ + \sum_{i=1}^{t-1} a_i x^i \, mod \, p \end{aligned} \tag{25}$$

Each member has the new secret and when the secret is recovered, the result is S_{new}.

Security analysis of the protocol

This section is devoted to the evaluation of the accomplishments of the protocol's security properties and resistance to attacks.

Mutual authentication

- $MTCD_{i-j} \rightarrow HSS$

HSS authenticates $MTCD_{leader}$ and all $MTCD_{i-j}$ simultaneously by verifying MAC_{Gi}, which authenticates the group, because only a legitimate group has a valid GK and a valid ID_{Gi}. MAC_{Gi} also authenticates each $MTCD_{i-j}$, because only legitimate and registered devices can find the original secret S produced by KGC in the registration phase.

- $MTCD_{i-j} \rightarrow MME$

MME authenticates all $MTCD_{i-j}$ calculating secret S'' by their Lagrange component received in message 7 and comparing $H(S'')$ with $H(S)$, provided by KGC in the registration phase. Only legitimate MTCDs can generate valid Lagrange components and secret S can be recovered only with valid Lagrange components.

- $MME \rightarrow MTCD_{i-j}$

Each $MTCD_{i-j}$ authenticates MME verifying its Lagrange component. All devices calculate secret S using the Lagrange component of MME and comparing it with $H(S)$ published by KGC. Such verification authenticates MME because it generates a valid Lagrange's component only if it has received a legit token from HSS.

- $MTCD_{i-j} \rightarrow MTCD_{i-k}$

$MTCD_{i-j}$ authenticates themselves prior to the authentication procedure in the core network. Each device sends its Lagrange component to all members in the group. Each member uses such components to calculate secret S and compares the value found with the value published by KGC. The Lagrange interpolating formula guarantees the original secret is recovered only if all devices are legitimate.

MITM attack

- The channel between HSS and MME is secure; therefore, only the entity affected by an attacker may act between $MTCD_{i-j}$ and MME. The mutual authentication phase is protected from MITM because:

- Shamir's secret and Lagrange interpolating formula are used. The formula enables the construction of a Lagrange component based on the secret token. The recovery of the secret token from the Lagrange component is quite complex; the secret can be recovered only with valid Lagrange components.
- The group's ID is secret; only TID is public. Only the one that knows the ID can generate or verify MAC_{Gi}.
- GK and GTK are used. As only $MTCD_{i-j}$ of the same group and HSS know GK, only they can generate the GTK; and
- The session key is used in the communication between the device and MME; only legitimate devices can obtain a session key.

Replay attack

- Each authentication process is different from the previous ones, because new random values are generated to compose the messages. Therefore, the repetition of messages is almost impossible.
- The parameters responsible for such a protection are as follows:
- Random values r_{MME}, r_{HSS}, and $r_{MTCDi-j}$ present in session key and GTK and
- Use of temporary identities TID_{MTCD} and TID_{Gi}, which are updated in each new authentication process to a never used value and are never repeated.

Privacy (anonymity)

- The privacy of the devices is protected by temporary identities (TID) against targeted attacks, so that an attacker does not know the real device's identity.

Redirection attack

- Each MTCD leader includes base station LAI in MAC_{Gi} and MME (that also knows the LAI of the devices assigned base station) sends it to the HSS on a secure channel. If an attacker tries to forge LAI, the verification of MAC_{Gi} fails and the redirection attack is avoided.

Personification attack

- Such an attack occurs when an attacker pretends it is a legitimate MTCD or MME.
- *MTCD → HSS*

An attacker cannot forge valid tokens $f(TID_{MTCDi-j})$ because they can be built only by KGC and are based on secret S, in a way the right secret is recovered. As attackers cannot produce a valid Lagrange component, when secret S is calculated, the value found is different from the one published by KGC. When HSS checks MAC_{Gi} using secret S, it can easily detect it is an attacker in the group.

- *MTCD → MME*

An attacker cannot forge valid tokens $f(TID_{MTCDi-j})$; consequently, they cannot produce a valid Lagrange component. When MME has received all Lagrange components of a group, it tries to recover secret S and realizes it is not the same published by KGC.

- *MME → MTCD*

Similarly, an attacker cannot forge a valid Lagrange component; therefore, when the MTCDs check LC_{MME}, they realize it is an attacker, because the secret found is not the same published by KGC.

- *MTCD Intruder Group → HSS*

A set of attackers may pretend they are a registered MTC group in network; therefore, the attack will not succeed because only legitimate groups know a valid GK and can produce a valid S. HSS will recognize the attack by verifying MAC_{Gi}.

- $MTCD_{i-j} \rightarrow MTCD_{i-k}$

Although from the same group, an MTCD cannot pretend to impersonate another MTCD of its group, because a device does not know the secret tokens, $f(TID_{MTCDi-j})$, of each other and the attacker cannot forge a valid Lagrange component of another member. Before a message is sent to the network, the MTC group authenticates themselves calculating secret S and all members realize at least one attacker is in the group. Consequently, the process fails. Finally, a device cannot generate a valid session key, $SK_{i-j-MME}$, of another device, because it does not know its own secret value in the tree.

DoS attack

This attack occurs when an attacker tries to drop the server or network sending a large number of authentication messages until it stops working properly:

- In our protocol, HSS receives the first message only when the members of the group have authenticated each other; therefore, all devices can detect the presence of attackers and stop the procedure, avoiding involving HSS in the authentication procedure.

- An attacker might create many fake messages to interrupt the HSS service. In our scheme, the first message HSS receives contains MAC_{Gi} and HSS can quickly check if it is valid or not calculating MAC'_{Gi} and comparing the two MAC values. Such verification is performed at the beginning of the process; therefore, the remaining authentication procedure is not affected if an attack is discovered in this stage.

Backward secrecy and forward secrecy

- The keys that guarantee backward secrecy (BS) and forward secrecy (FS) are GK, session key $SK_{i-j-MME}$, and secret S.

In our protocol, when a device enters or leaves the group, GK is updated to perform BS and FS. In other words, if a device leaves, it cannot discover the future GK and if a device enters the group, it cannot discover the past GK.

When a device is added to the group, HSS broadcasts its secret node to all other devices and the new GK is generated

$$GK'_i = h_3(GK_i \oplus SEC_{i-y}) \tag{26}$$

When a device leaves, each device updates its GK as follows

$$GK''_i = GK_i \oplus SEC_{i-y} \tag{27}$$

Our protocol guarantees strong backward secrecy (sBS) and forward secrecy (sFS) to GK, because although an attacker discovers the current GK, it cannot discover past and future GKs, once it does not know the secret value used in the formula. Even if it occasionally discovers the current GK and the secret values used for its generation, it will not compromise past or future GKs, because the values used in the calculation are renewed in each update. The same occurs with $SK_{i-j-MME}$, because it is calculated as follows

$$SK_{i-j-MME} = h_3(SEC_{i-a} \oplus SEC_{i-b} \oplus \cdots \oplus SEC_{i-z} || r_{MME} || r_{MTCDi-j} || S) \tag{28}$$

If an attacker discovers the current value of the session key, it cannot associate it with past or future keys, because it does not know the secret values (even if it is a member's group, it does not know its own secret value) and secret S and r_{MME} (if the attacker is not a group member). Although the attacker can eventually discover all secret values, secret S, and currents r_{MME} and $r_{MTCDi-j}$, it cannot calculate past or future keys, because such values are randomly generated in each new authentication process. If the attacker is not part of the group, it will not know secret S and r_{MME}. Therefore, our session key has strong BS and FS.

Secret S must guarantee BS and FS; otherwise, each new or old member will know the secret of the group and can try to perform attacks with this information. Consequently, any modification in the group formation requires an update in secret S. The new secret is defined as

$$S_{new} = S + \Delta S \tag{29}$$

where ΔS is a random term defined whenever an update in S is required. Even if an attacker discovers the current or last secret S, it will not discover the next or the other past secrets, because S is defined by ΔS. Even if ΔS is discovered, this value is not correlated with future or past values and S is not compromised. Therefore, secret S has strong FS and BS. Table 4 shows a comparison of protocols based on the previously discussed set of security objectives.

Performance evaluation

This section addresses the evaluation of the protocol performance and a comparison with the performance of some other protocols.[1,3,5,6] All of them consider an MTC architecture with MTCD, MTC leader, MME, HSS, and MTC server and a safe channel between HSS and MME. They also have a registration/initialization phase that defines all parameters necessary for authentication and an authentication and key agreement phase that authenticates the MTCDs and establishes a session key between MTCD and MME.

Computational cost

The comparison of the computational cost of the protocol with the other schemes analyzed is here addressed. This cost is evaluated considering the processing time necessary to execute each operation necessary for the execution of protocols here considered. Table 5 shows the values of time cost for each operation, based on experimental evaluation by previous works,[1,3,5,6] with some natural differentiation regarding the processing power of MTCD and the components of the EPC network (core network). The time spent on an XOR operation has been omitted, since it is negligible in comparison to the other operations.

The analysis considered computational costs related to MTCDs and the core network in separated parts, as shown in Table 6. An environment with n devices, divided into m groups, where all groups have n/m

Table 4. Comparison of security objectives among protocols.

Security objectives	Schemes				
	EPS-AKA[1]	GLARM[5]	CHOI[6]	GR-AKA[3]	Proposed protocol
Mutual authentication and key agreement	Yes	Yes	Yes	Yes	Yes
Confidentiality	No	Yes	Yes	Yes	Yes
Integrity	No	Yes	Yes	Yes	Yes
Privacy (anonymity)	No	No	No	No	Yes
Perfect FS/BS	No	Yes	Yes	Yes	Yes
Resistance to replay attack	No	Yes	Yes	Yes	Yes
Resistance to DoS attack	No	Yes	Yes	No	Yes
Resistance to man-in-the-middle attack	No	Yes	Yes	Yes	Yes
Resistance to redirection attack	No	Yes	Yes	Yes	Yes
Resistance to impersonation attack	No	Yes	No	No	Yes

EPS-AKA: evolved packet system authentication and key agreement; GLARM: group-based lightweight authentication scheme for resource constrained machine-to-machine; GR-AKA: group-based authentication and key agreement; FS/BS: forward secrecy/backward secrecy.

Table 5. Time costs in milliseconds of each operation considered.

Notation	Cost (ms)	Description
T_M	0.013	Cost of a normal multiplication operation
Thash	0.06	Cost of a one-way hash operation
Tmul (MTCD/core)	1.537/0.475	Cost of a multiplication operation over an elliptical curve
Tmod	0.12	Cost of a modular operation
Taes	0.16	Cost of an AES encryption operation
TL_{MTCD}	0.0572	Cost of a Lagrange component creation in the MTCDs
TL_{Core}	0.0351	Cost of a Lagrange component creation in the core network

MTCD: mobile terminal communication devices; AES: Advanced Encryption Standard.

Table 6. Comparison of the computation costs among protocols.

Schemes	MTCDs (ms)	Core network (ms)	Total (ms)
EPS-AKA[1]	6nThash + nTaes = 0.52n	6nThash + nTaes = 0.52n	1.04n
CHOI[6]	(7n + 3m)Thash + nTmod + m Taes = 0.54n + 0.34m	(3n + 6m)Thash + nTmod + mTaes = 0.3n + 0.52m	0.84n + 0.86m
GLARM[5]	8nThash + mThash = 0.48n + 0.06m	5nThash + 4mThash = 0.3n + 0.24m	0.78n + 0.3m
GR-AKA[3]	2nTmul + 3nThash + nTL_{MTCD} + 2mTmod + 4mThash = 3.31n + 0.48m	nThash + nTmul + mTL_{Core} + 2mThash + mTmul = 0.53n + 0.63m	3.84n + 1.11m
Proposed Protocol	nTL_{MTCD} + nT_M + (3n + m)Thash + 2nTmod = 0.49n + 0.06m	mTL_{Core} + (n + 3m)Thash + 2mTmod = 0.06n + 0.48m	0.55n + 0.54m

MTCD: mobile terminal communication devices; EPS-AKA: evolved packet system authentication and key agreement; GLARM: group-based lightweight authentication scheme for resource constrained machine-to-machine; GR-AKA: group-based authentication and key agreement.

members, is considered. Each MTCD performs three hash operations (GTK_{Gi}, $H(S)$, $SK_{i-j-MME}$), one modular operation (mod p), and one Lagrange component generation (LC_{i-j}). The group leader performs only a hash operation (MAC_{Gi}), with a total of nTL_{MTCD} + (3n + m)Thash + 2nTmod = 0.49n + 0.06m ms in all operations.

According to Table 6, the proposed protocol required the lowest computational cost and reached the best performance in comparison to the other protocols. For example, it performs only 0.55n + 0.54m operations in the authentication procedure, which is much fewer than 3.84n + 1.11m of GR-AKA.[3]

Figures 4–7 show the computational costs of the five evaluated protocols as a function of number of devices for specific values of m (m = number of groups). According to the figures and the expressions in the rightmost column of Table 6, the increase in the communication cost is linear as a function of the number of devices (n).

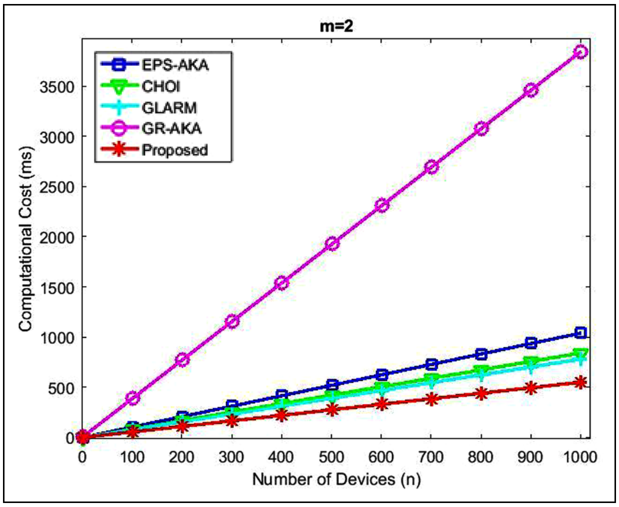


Figure 4. Comparison of computational costs, for m = 2.

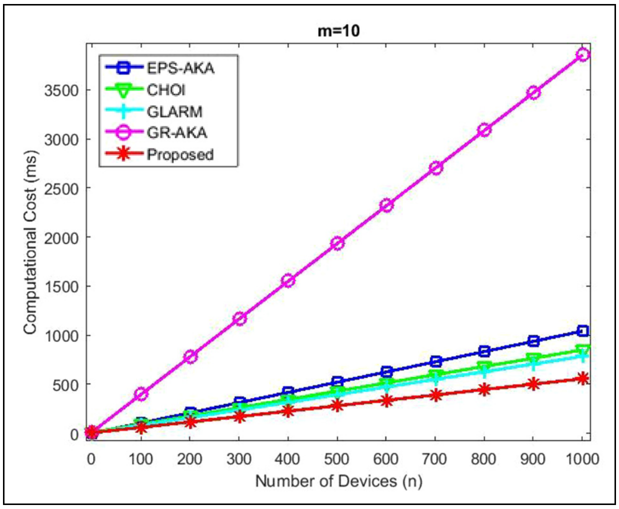


Figure 5. Comparison of computational costs, for m = 10.

Figure 4 shows the best performance of the proposed protocol in comparison to other protocols, even for a small number of groups (m = 2).

Figure 5 shows our protocol keeps the best performance in comparison to the other protocols if the number of devices increased to 10.

According to Figure 6, the proposed protocol achieves the best performance for 50 groups of devices.

Finally, Figure 7 shows the best performance achieved by our protocol when the number of groups is increased to m = 100, for n > 100. For n $\leqslant$ 100, the best protocol is EPS-AKA.[1] However, problems related to security of EPS-AKA[1] must be emphasized.

Communication cost

The communication cost was measured in bits according to the messages exchanged. The values adopted for each parameter transmitted are shown in Table 7. They were carefully chosen and based on the values used in previous works.[1,3,5,6] An environment with n devices, divided into m groups, where each group n/m members was considered. The calculations were based on the number of messages, with their respective parameters exchanged in each message, that is, each parameter sent through the channel. Taking message 5 as an example, HSS sends $LC_{MME}, GTK_{Gi} = h_2(\text{GK}||r_{HSS})$ and r_{HSS} to MME. Therefore, the message has two hash functions with 128 bits each and a random number with 128 bits, which totals 384m bits. Table 8 shows a comparison among the communication cost of the proposed protocol and those of the other protocols analyzed.

According to Table 8, the proposed protocol required the lowest communication cost in comparison to the other protocols analyzed, once it sends a reduced number of bits, depending on the number of devices n. For example, it requires only 640n + 1320m bits of message to perform an authentication procedure. This is a reduced number, in comparison to GR-AKA,[3] which demands 1108n + 996m bits. Figure 7 also shows the comparison and the good performance of the protocol.

Figures 8–11 show the communication costs of the five evaluated protocols as a function of number of devices, for specific values of m. According to the figures and the expressions in the rightmost column of Table 8, the increase in the communication cost is linear as a function of number of devices (n).

Figure 8 shows a comparison of the communication cost of the protocols for two groups of devices. Our protocol clearly achieves the best performance for a small number of groups, as the number of devices increases.

According to Figure 9, if the number of groups is increased to 10, our protocol still shows the best communication costs, as the number of devices increases.

Figure 10 shows if the number of groups is increased to 50, our protocol still has the best communication cost, as the number of devices increases, for n $\geqslant$ 37. For n < 37, EPS-AKA[1] outperforms the other protocols.

Finally, according to Figure 11, if the number of groups is increased to 100, the proposed protocol shows, in most cases, the best performance, as the number of devices increases, in comparison to previous works.[3,5,6] It has confirmed the expected results of the calculations shown in Table 8.

In a summarized way, the graphs in Figures 8–11 confirmed the lowest communication costs of the proposed protocols for almost all values of m, as the number of devices (n) increases. Figures 8 and 9 show the protocol has the best overall performance with groups with 2 or 10 devices. Figures 12 and 13 display three-dimensional representations of computational and

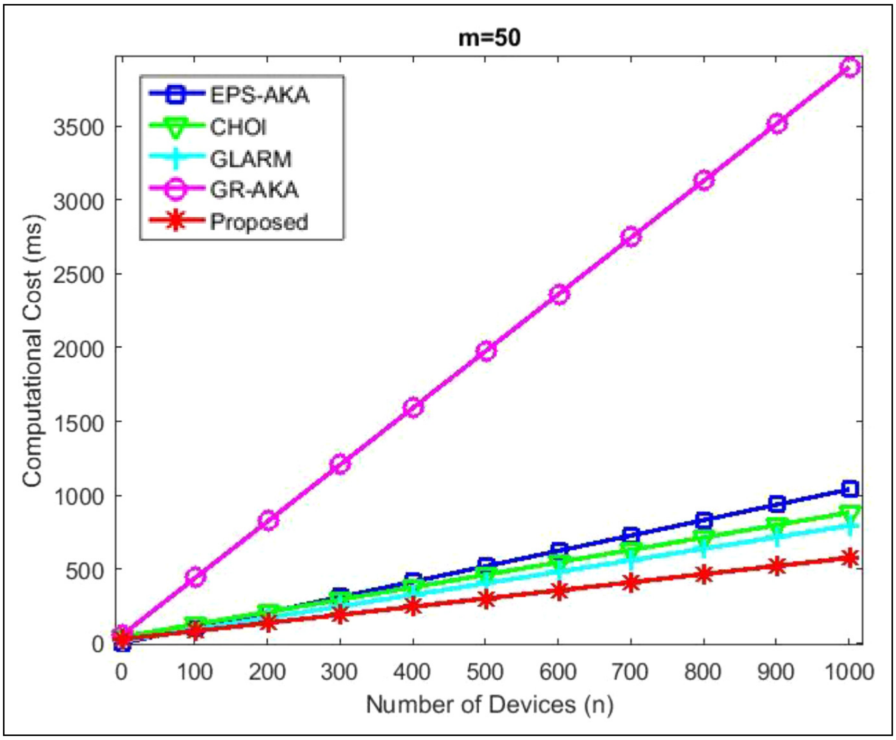


Figure 6. Comparison of computational costs, for m = 50.

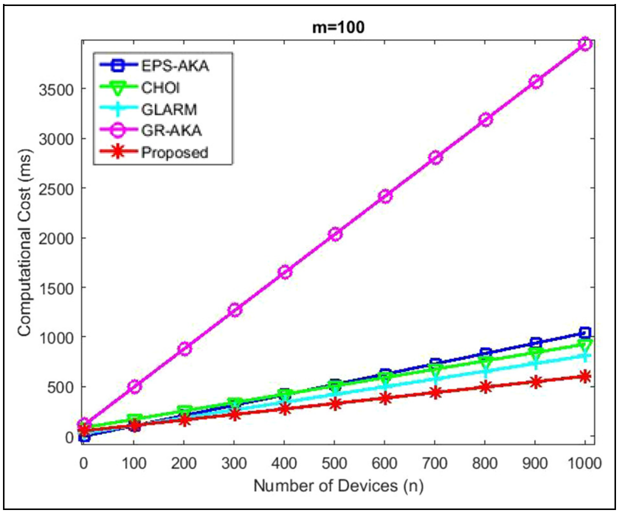


Figure 7. Comparison of computational costs, for m = 100.

communication costs, respectively, for providing a global view of the proposed protocol's performance.

Figure 12 shows how the proposed protocol has the lowest computational cost while number of devices (n) and number of groups (m) increase separately or while both n and m increase.

Figure 13 provides a three-dimensional view of the communication costs for emphasizing the lowest cost required by the proposed protocol and its best performance in comparison to the other protocols, as the number of devices (n) and groups (m) rises.

Table 7. Communication cost of each parameter transmitted.

Parameter	Size (bits)
ID/TID	128
ECDH	192
MAC	64
Hash	128
LC	128
Rand	128
LAI	40

ID/TID: Identification/Temporary Identification; ECDH: Elliptic Curve - Diffie Hellman; MAC: Message Authentication Code; LC: Lagrange Component; LAI: Location Area Identification.

Formal verification of the proposed protocol

This section addresses a formal verification of the protocol's security properties conducted by AVISPA,[7,8] a tool widely used for Internet security assessments. It employs HLPSL (High-Level Protocol Specification Language), which describes the exchange of messages necessary for the operation of the protocol, as well as the behavior of each entity for simulating the functioning of the protocol.

Table 8. Communication cost in bits per message.

Schemes	M1	M2	M3	M4	M5	M6	M7	M8	Total
EPS-AKA[1]	–	128 bits	256 bits	704 bits	576 bits	128 bits	–	–	1792n bits
CHOI[6]	128n + 128m bits	128n + 256m bits	960m bits	576m bits	128m bits	128m bits	448n – 448 bits	256m bits	704n + 1984m bits
GLARM[5]	448n – 448m bits	384n + 64m bits	384n + 104m bits	880m bits	560m bits	560m bits	128(n –m) bits	128m bits	1344n + 1720m bits
GR-AKA[3]	340(n – m) bits	340m bits	380m bits	212m bits	404m bits	768n bits	–	–	1108n + 996m bits
Proposed Protocol	128n bits	128n bits	128n + 192m bits	232m bits	384m bits	512m bits	256n bits		640n + 1320m bits

EPS-AKA: evolved packet system authentication and key agreement; GLARM: group-based lightweight authentication scheme for resource constrained machine-to-machine; GR-AKA: group-based authentication and key agreement.

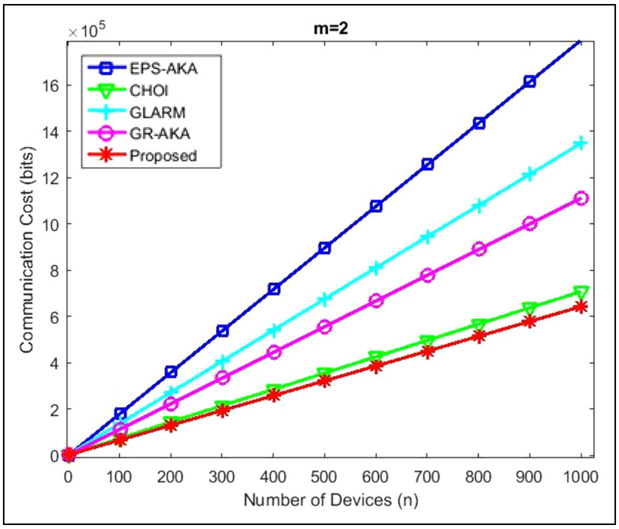


Figure 8. Comparison of communication costs, for m = 2.

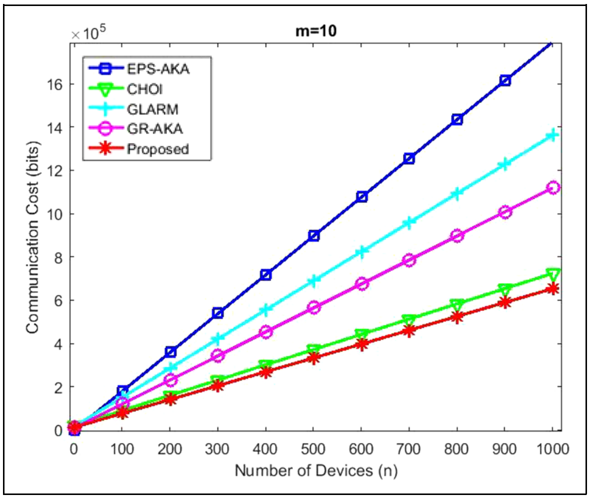


Figure 9. Comparison of communication costs, for m = 10.

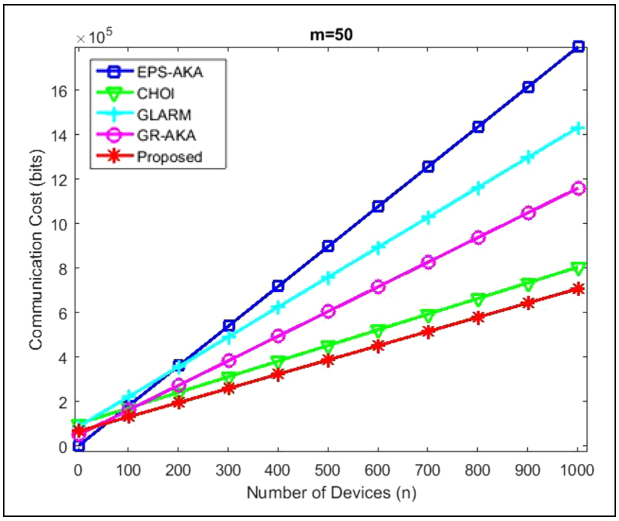


Figure 10. Comparison of communication costs, for m = 50.

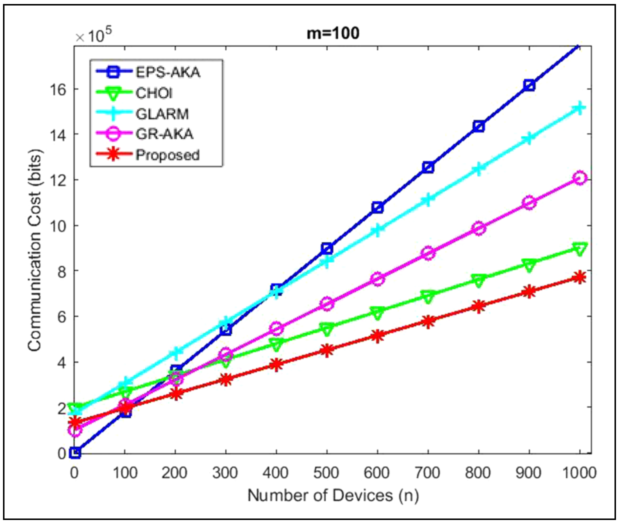


Figure 11. Comparison of communication costs, for m = 100.

In HLPSL, each entity plays a role. In the proposed protocol, each MTCD performs an authentication procedure. Two MTCDs, namely $MTCD_{leader}$ and an ordinary $MTCDij$, were assumed in the verification procedure. Therefore, the roles implemented were those of an ordinary $MTCDij$, $MTCD_{leader}$, MME, and HSS. Figure 14 describes the role of MTCDij. Transitions from a state to another occur simultaneously with the exchange of messages. The verification is performed in eight states. When State = 1, $MTCDij$ sends the other devices ($MTCD_{leader}$, in this verification) its $TDI_{MTCDi-j}$.The state is changed from 1 to 2. $MTCD_{leader}$ performs the same procedure and sends its $TDI_{MTCDi-j}$ to $MTCDij$. When State = 2, $MTCDij$ calculates its Lagrange component $LC_{\mathrm{MTCD}i-j}$.

Figure 15 shows the security goals, which must be accomplished by the proposed protocol, including mutual authentication between MTCD (auth_1) and

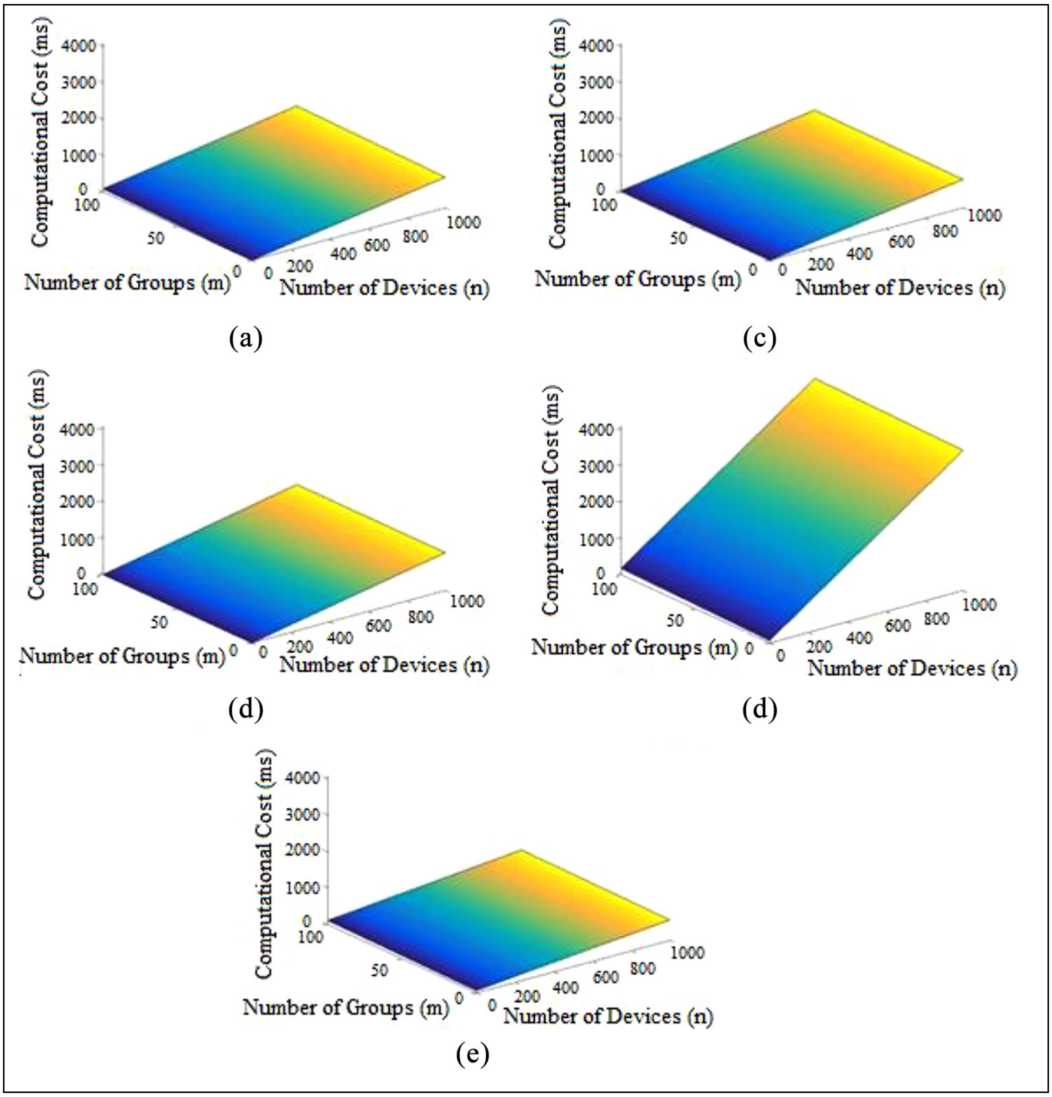


Figure 12. Comparison of computational costs: (a) CHOI,[6] (b) EPS-AKA, (c) GLARM, (d) GR-AKA, and (e) proposed protocol.

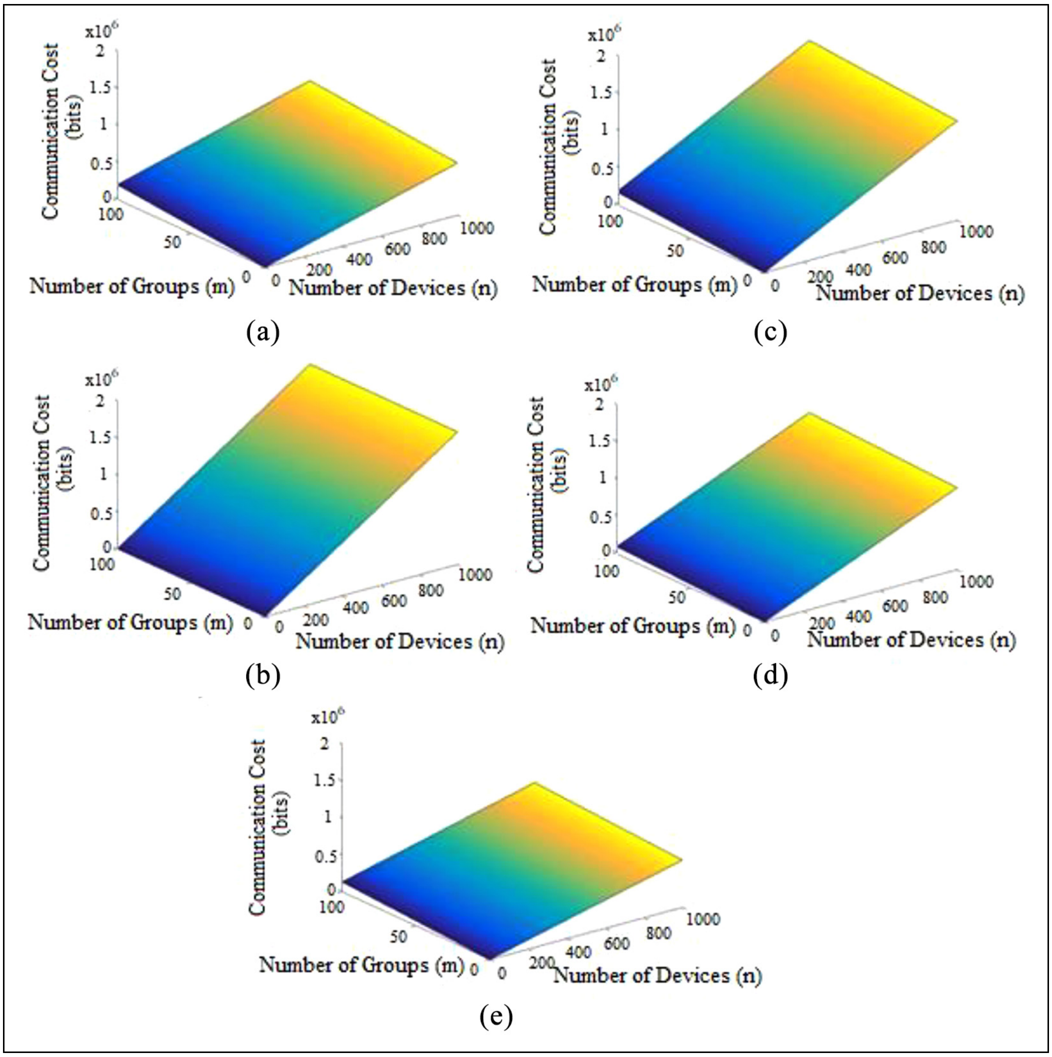


Figure 13. Comparison of communication costs: (a) CHOI,[6] (b) EPS-AKA, (c) GLARM, (d) GR-AKA, and (e) proposed protocol.

MME (auth_2) and secrecy of important parameters between different entities, as group temporary key, device's random number, permanent identities, and session key.

Results of security verification

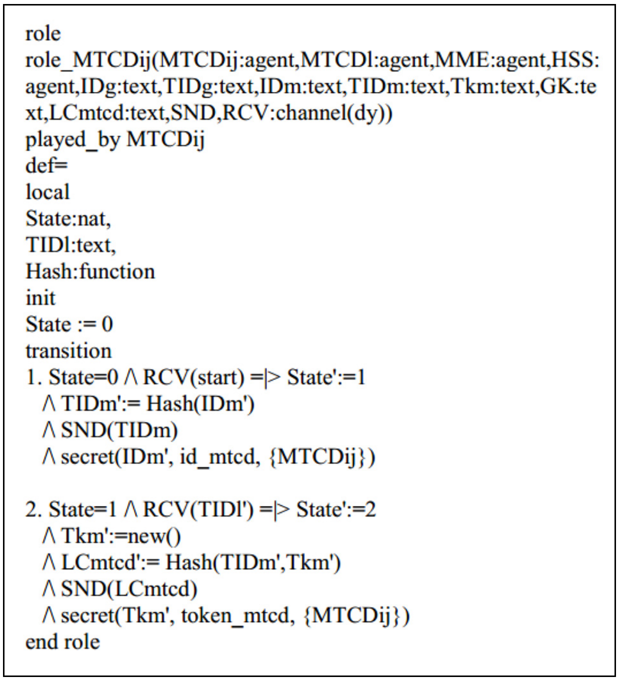
Two security simulations based on On-the-fly Model-Checker (OFMC)[17] and Constraint-Logic-based Attack Searcher (CL-AtSe)[8] were conducted. The results show that the proposed protocol is considered safe by both checker mechanisms for the goals specified (Figures 16 and 17). CLAtSe[8] results show all eight states were reached.

AVISPA[7,8] also comprehends a graphic simulation tool, SPAN (security protocol animator for AVISPA),[18] which enables a better visualization of exchanged messages and the participation of the intruder during the protocol. The graphical animations of our protocol are shown in Figure 18, and Figure 19 displays the simulation of an intruder's action. In the scenario adopted, an intruder might completely control the network, that is, intercept, analyze, and modify the messages.

Conclusion

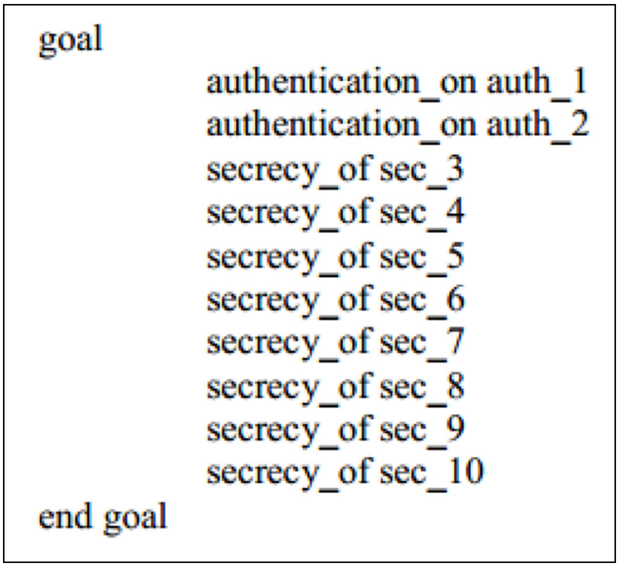
Authentication represents a critical issue regarding the widespread adoption of the IoT paradigm and the development of 5G networks. A large number of sensors are expected to provide massive streams of real-time and non-real-time data to support decision-making processes, in a large number of applications and scenarios, as e-Health/m-Health, smart grids, smart homes, and public transportation.

In IoT, the traffic produced by an extensive number of devices/sensors is expected to trigger congestion in signaling networks, resulting from the overloading of links, processors, and memory resources.



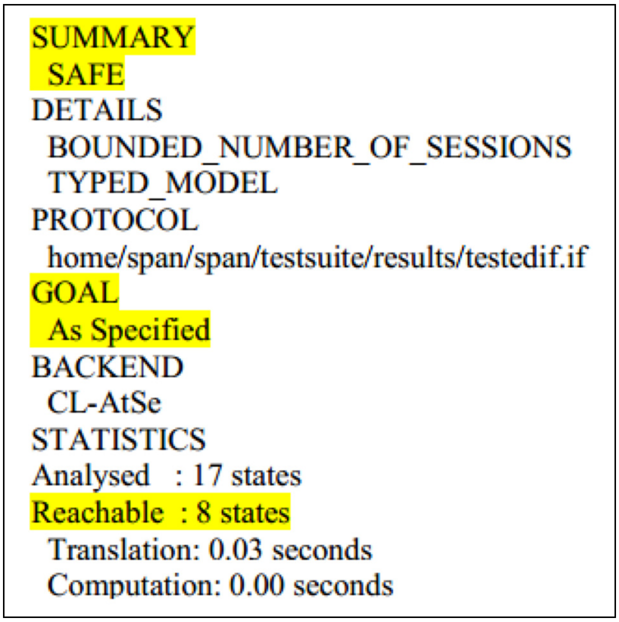
```
role
role_MTCDij(MTCDij:agent,MTCDl:agent,MME:agent,HSS:
agent,IDg:text,TIDg:text,IDm:text,TIDm:text,Tkm:text,GK:te
xt,LCmtcd:text,SND,RCV:channel(dy))
played_by MTCDij
def=
local
State:nat,
TIDl:text,
Hash:function
init
State := 0
transition
1. State=0 ∧ RCV(start) =|> State':=1
  ∧ TIDm':= Hash(IDm')
  ∧ SND(TIDm)
  ∧ secret(IDm', id_mtcd, {MTCDij})

2. State=1 ∧ RCV(TIDl') =|> State':=2
  ∧ Tkm':=new()
  ∧ LCmtcd':= Hash(TIDm',Tkm')
  ∧ SND(LCmtcd)
  ∧ secret(Tkm', token_mtcd, {MTCDij})
end role
```

Figure 14. Role of each MTCD in HLPSL.



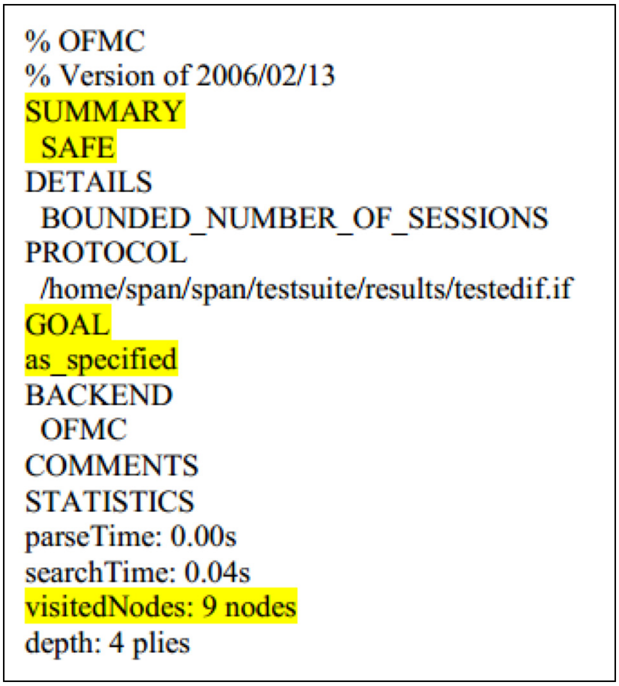
```
goal
             authentication_on auth_1
             authentication_on auth_2
             secrecy_of sec_3
             secrecy_of sec_4
             secrecy_of sec_5
             secrecy_of sec_6
             secrecy_of sec_7
             secrecy_of sec_8
             secrecy_of sec_9
             secrecy_of sec_10
end goal
```

Figure 15. Security goals established in HLPSL.



```
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL
PROTOCOL
 home/span/span/testsuite/results/testedif.if
GOAL
 As Specified
BACKEND
 CL-AtSe
STATISTICS
Analysed   : 17 states
Reachable  : 8 states
 Translation: 0.03 seconds
 Computation: 0.00 seconds
```

Figure 16. Security simulation results for CLAtSe.



```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/testedif.if
GOAL
as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
parseTime: 0.00s
searchTime: 0.04s
visitedNodes: 9 nodes
depth: 4 plies
```

Figure 17. Security simulation results for OFMC.

Moreover, in such technology, sensors with similar characteristics and management are commonly found. Therefore, organizing devices in groups is a natural choice to reduce some bottlenecks regarding computing and communication infrastructures necessary for IoT implementation.

The MTC technology might assist IoT applications by including a MTC server in the 3GPP architecture to provide management of the data collected. In addition, MTC enables MTC users the ability to remotely control collected data, such as a physician monitoring patients' vital signs in e-Health/m-health or a farmer monitoring variables as humidity, sun light, and temperature in intelligent agriculture, for example.

Among the authentication protocols considered, the inadequacy of the standardized protocol (EPS-AKA)[1] to deal with groups of terminals/sensors was initially observed. After a literature review, this work proposed

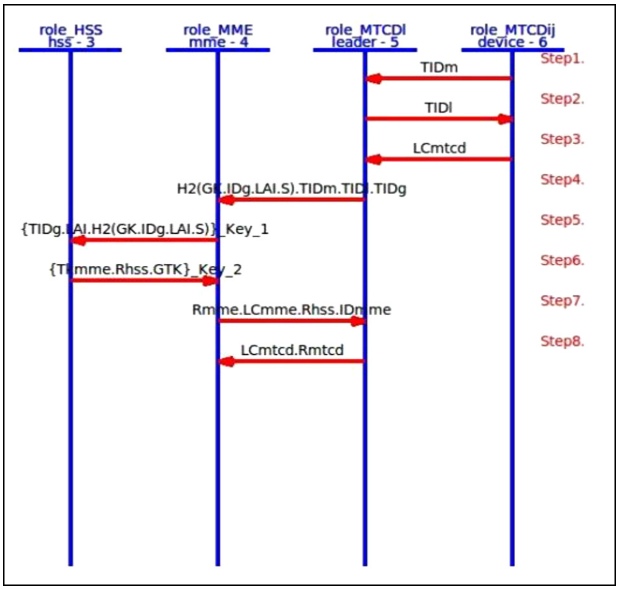


Figure 18. Protocol's message exchange in SPAN.[18]

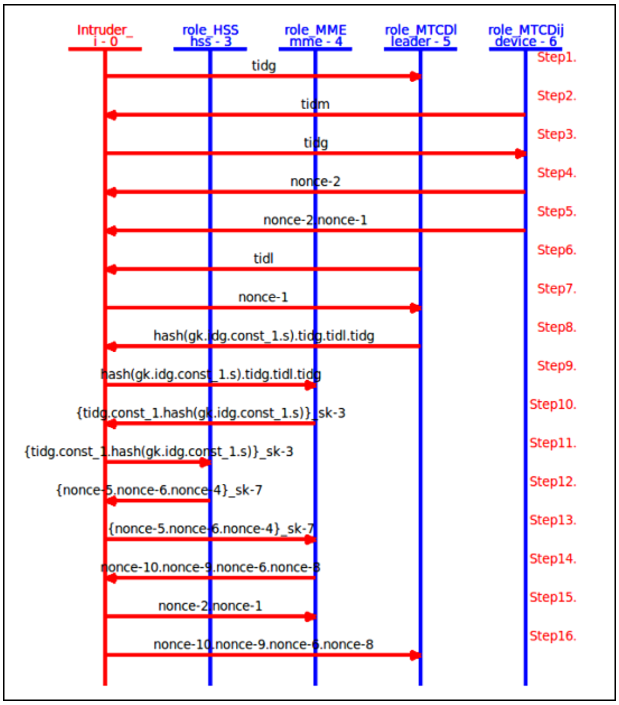

Intruder_
i - 0
role_HSS
hss - 3
role_MME
mme - 4
role_MTCDl
leader - 5
role_MTCDij
device - 6
Step1.
tidg
Step2.
tidm
Step3.
tidg
Step4.
nonce-2
Step5.
nonce-2.nonce-1
Step6.
tidl
Step7.
nonce-1
Step8.
hash(gk.idg.const_1.s).tidg.tidl.tidg
Step9.
hash(gk.idg.const_1.s).tidg.tidl.tidg
Step10.
{tidg.const_1.hash(gk.idg.const_1.s)}_sk-3
Step11.
{tidg.const_1.hash(gk.idg.const_1.s)}_sk-3
Step12.
{nonce-5.nonce-6.nonce-4}_sk-7
Step13.
{nonce-5.nonce-6.nonce-4}_sk-7
Step14.
nonce-10.nonce-9.nonce-6.nonce-8
Step15.
nonce-2.nonce-1
Step16.
nonce-10.nonce-9.nonce-6.nonce-8

Figure 19. Intruder's simulation in SPAN.[18]

a group authentication and key agreement protocol, able to support to a large number of MTC devices; it is based on symmetric cryptography, secret sharing, and Lagrange interpolation and compared it with four other protocols.

The comparison was initially based on security properties and security objectives discussed and evaluated, according to several possible threats and attacks (e.g. confidentiality, integrity, resistance to replay and DoS attacks, resistance to MITM, redirection, and impersonation attacks). The protocol has proven resistant to the threats and attacks considered.

A performance analysis of the computational and communication costs of five protocols was conducted. The computational costs were evaluated according to the number of bits each protocol required in its operations, whereas the communication costs were measured in bits according to the messages exchanged.

The whole set of messages dealt with by each protocol and the respective number of bits were considered in the evaluation of the communication costs. Figures of performance in two and three dimensions showed the proposed protocol outperformed the other four protocols in most scenarios and situations.

Ongoing studies involve the formal validation of the protocol and future work aims at adapting it to smart city environments and some of their specific verticals/sectors (e-Health, smart grids, etc.).

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs

Paulo RL Gondim https://orcid.org/0000-0002-7007-1969
Jaime Lloret https://orcid.org/0000-0002-0862-0533

References

1. 3GPP. TS 33.401 V8.2.1, 3GPP system architecture evolution (SAE), security architecture, 2009, https://www.etsi.org/deliver/etsi_ts/133400_133499/133401/08.02.01_60/ts_133401v080201p.pdf
2. Shamir A. How to share a secret. *Comm ACM* 1979; 22(11): 612–613.
3. Li J, Wen M and Zhang T. Group—based authentication and key agreement with dynamic policy updating for MTC in LTE—a networks. *IEEE Internet Things J* 2016; 99: 1–9.
4. Harn L. Group authentication. *IEEE Trans Comp* 2012; 62(9): 1893–1898.
5. Lai C, Lu R, Zheng D, et al. GLARM: group-based lightweight authentication scheme for resource-constrained machine to machine communications. *Comp Networks* 2016; 99: 66–81.
6. Choi D, Hong S and Choi HK. A group-based security protocol for Machine Type Communications in LTE-advanced. *Wireless Networks* 2015; 21(2): 405–419.

7. The AVISPA project: European Union in the Future and Emerging Technologies (FET Open), http://www.avispa-project.org (accessed 26 November 2017).
8. Armando A. The AVISPA tool for the automated validation of internet security protocols and applications. In: *17th international conference on CAV*, Edinburgh, 6–10 July 2005, pp.281–285. London: Springer.
9. Lacuesta R, Lloret J, Garcia M, et al. Two secure and energy-saving spontaneous ad-hoc protocol for wireless mesh client networks. *IJCNIS* 2011; 34(2): 492–505.
10. Lacuesta R, Lloret J and Garcia M. A secure protocol for spontaneous wireless ad hoc networks creation. *IEEE Trans Parallel Distributed Syst* 2013; 24(4): 629–641.
11. Garcia M, Lloret J, Sendra S, et al. Secure communications in group-based wireless sensor networks. *IJCNIS* 2010; 2(1): 8–14.
12. Fu A, Song J, Li S, et al. A privacy—preserving group authentication protocol for machine—type communication in LTE/LTE—a networks. *Security Comm Networks* 2016; 9(13): 2002–2014.
13. Lai C, Lu R and Shen X. SE-AKA: a secure and efficient group authentication and key agreement protocol for LTE networks. *Comp Networks* 2013; 57(17): 3492–3510.
14. Gupta S, Parne BL and Chaudhari NS. DGBES: dynamic group based efficient and secure authentication and key agreement protocol for MTC in LTE/LTE—a networks. *Wireless Person Comm* 2018; 98(3): 2867–2899.
15. Parne BL, Gupta S and Chaudhari NS. SEGB: security enhanced group based AKA protocol for M2M communication in an IoT enabled LTE/LTE—a network. *IEEE Access* 2018; 6: 3668–3684.
16. Dutta R, Barua R and Sarkar P. Pairing-based cryptography: a survey. *Cryptology Research Group, Statistics, Mathematics and Applied Statistics Unit*, vol. 203, 2004, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.145.9806
17. Basin D, Moedersheim S and Vigano L. OFMC: a symbolic model checker for security protocols. *Int J Informat Security* 2005; 4(3): 181–208.
18. Glouche Y, Genet T and Houssay E. SPAN—a Security Protocol ANimator for AVISPA —user manual. *Irisa Univ Rennes* 2006; 1: 20.