WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. (2015)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/wcm.2612

RESEARCH ARTICLE

Secure machine-type communications in LTE networks

Chengzhe Lai[1,2,3]*, Rongxing Lu[4], Hui Li[2], Dong Zheng[1] and Xuemin (Sherman) Shen[5]

[1] National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an, China
[2] State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China
[3] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[4] School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore
[5] Department of Electrical and Computer Engineering, University of Waterloo, Waterloo ON, Canada

ABSTRACT

With a great variety of potential applications, machine-type communications (MTC) is gaining a tremendous interest from mobile network operators and research groups. MTC is standardized by the 3rd Generation Partnership Project (3GPP), which has been regarded as the promising solution facilitating machine-to-machine communications. In the latest standard, 3GPP proposes a novel architecture for MTC, in which the MTC server is located outside the operator domain. However, the connection between the 3GPP core network and MTC server in this scenario is insecure; consequently, there are distrustful relationships among MTC device, core network, and MTC server. If the security issue is not well addressed, all applications involved in MTC cannot be put into the market. To address this problem, we propose an end-to-end security scheme for MTC based on the proxy-signature technique, called E^2SEC. Specifically, both the MTC device and MTC server can establish strong trustful relationships with each other by using the proxy signatures issued by the 3GPP core network. Moreover, we present some implementation considerations of E^2SEC and analyze the performance during authentication by comparing the operational cost of three cases that apply three different signature algorithms, that is, ElGamal, Schnorr, and DSA. Through security analysis by using Automatic Cryptographic Protocol Verifier (ProVerif), we conclude that the proposed E^2SEC scheme can achieve the security goals and prevent various security threats. Copyright © 2015 John Wiley & Sons, Ltd.

KEYWORDS
M2M; MTC; security; proxy signature; authentication and key agreement (AKA); LTE

***Correspondence**
Chengzhe Lai, National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an, China 710121.
E-mail: lcz.xupt@gmail.com

1. INTRODUCTION

Machine to machine (M2M) communications [7,11] refers to the technologies that allow both wireless and wired systems to communicate with other devices that have the same ability. M2M uses a device (e.g., a sensor or meter) to capture an event (e.g., temperature variation, and energy consumption), which is relayed through a network (wireless, wired, or hybrid) to an application (software program) that translates the captured event into meaningful information. Such communication was originally accomplished by having a remote network of machines relay information back to a central hub for analysis, which would then be rerouted into a system like a personal computer [1]. Because it does not need direct human intervention, M2M communications is fast becoming a market-changing force for the next-generation intelligent real-time networked application. Many research papers have explored the applications related to the M2M technology. Niyato *et al*. [26] propose an architecture of cognitive radio-based M2M communications for the smart grid, which can realize power efficiency of electricity distribution as well as spectrum efficiency. Fadlullah *et al*. [8] also investigate some applications of intelligent M2M communications in the smart grid. Zhang *et al*. [33] introduce some promising applications of M2M communications, for example, home multimedia distribution and sharing, intelligent transportation systems, and eHealthcare. Nowadays, M2M communications has become one of the most popular technologies in the standardization and industry areas. Many standards forums and organizations have actively engaged in M2M standard development, including the

Institute of Electrical and Electronics Engineers, the European Telecommunications Standards Institute, the China Communications Standards Association, oneM2M, Third Generation Partnership Project (3GPP), and 3GPP2. In release 10 of 3GPP, M2M communications is also called machine-type communication (MTC) [32], which works in the long-term evolution (LTE) networks. The scenarios of 3GPP have been regarded as the promising solution facilitating M2M communications [8].

Recently, the majority of studies on MTC have focused on congestion control, resource management, sensing, computing, and controlling technologies. [9,20–22,28,34]. Indeed, cyber security is of paramount importance in MTC because all applications involved in MTC cannot be put into the market without security guarantee. In the existing literature, Lu *et al.* [23] point out that the existing challenges of M2M, that is, energy efficiency (green), reliability and security. Bailey [2] analyzes M2M's impact on privacy and safety. Taleb *et al.* [29] present some potential challenges and solutions of MTC in 3GPP networks. Our previous works [15–18] also discuss the related security issues on M2M communications, for example, group access authentication and key agreement and efficient data authentication. In standardization, some security threats and candidate solutions for MTC have been introduced in 3GPP TR 33.868 [30], including MTC device (MTCD) triggering, secure connection, security of small data transmission, and external interface security. Besides, one of the most important requirements of security is that the network operator should be able to provide efficient security protection for connection between the MTCD and MTC server (MTCS)/MTC application server. 3GPP TS 22.368 [31] also requires that the operator must provide an end-to-end security protection for information interaction between MTCD and MTC applications. Nevertheless, when MTC devices are roaming, that is, these devices are connected via a visited public land mobile network, the network operator cannot control the security policy of operators located in the roaming domain. In this situation, how to fulfill the MTC application owner's end-to-end security requirement has become a critical issue in MTC.

In addition, another more complex situation will happen in the latest 3GPP standard. As shown in Figure 1, when MTCD communicate with the MTCS that is located in the operator domain and regulated by the 3GPP core network, the security of this scenario is the same as the existing standard. However, when MTCD communicate with the MTCS that is located outside the operator domain and cannot be regulated by the 3GPP core network, as shown in Figure 2, the connection between the MTCS and 3GPP core network is insecure. As a result, there are distrustful relationships among MTCD, core network, and MTCS. In this case, how to realize an end-to-end security communication between the MTCS and MTCD becomes more challenging.

Besides, in the radio access network, the MTCD can access the 3GPP core network not only via the evolved universal terrestrial radio access network (E-UTRAN), but also via other non-3GPP radio access technologies, such as worldwide interoperability for microwave access (WiMAX), wireless local area network (WLAN), and code division multiple access (CDMA). These radio access networks have different network architectures and separate security policies. Therefore, a generic security scheme dedicated to MTC in LTE networks is also desirable.

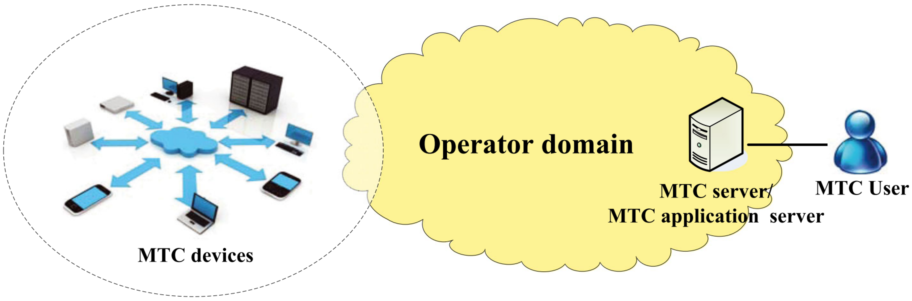


Figure 1. Machine-type communications (MTC) devices communicate with MTC server located in the operator domain.

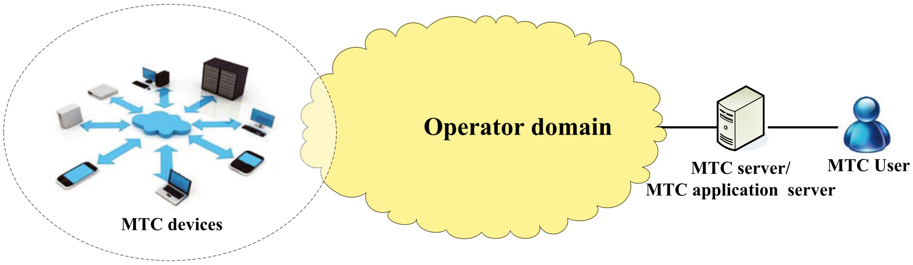


Figure 2. Machine-type communications (MTC) devices communicate with MTC server located outside the operator domain.

In this paper, we focus on building more secure machine-type communications in LTE networks by designing a unified end-to-end security scheme, called E^2SEC. In the proposed E^2SEC scheme, both the MTCD and MTCS can establish a strong trustful relationship with each other through using the proxy signatures [4] issued by the 3GPP core network. The main contributions of this paper are four-fold.

- Firstly, we adopt the proxy signature technique together with ElGamal signature scheme to design a secure and efficient authentication and key agreement (AKA) protocol between the MTCD and MTCS. After a successful mutual authentication, a trust relationship can be built between the MTCD and MTCS; meanwhile, an end-to-and secure channel can be established between them.
- Secondly, the secure communication between the MTCD and MTCS does not depend on security features defined in the network domain in which the MTCD is visiting; that is, they can efficiently perform mutual authentication and communicate with each other securely no matter where the MTCD is visiting (i.e., in a home network domain or roaming network domain).
- Thirdly, the security between the MTCD and MTCS does not depend on the specific radio access network technology. That is, no matter what kind of radio access technologies is used by the MTCD (e.g., E-UTRAN, WLAN, or WiMAX), the MTCD and MTCS can perform mutual authentication without regard to either security policies or architectures of each radio access network.
- Finally, we use Automatic Cryptographic Protocol Verifier (ProVerif) [3] to verify the security of our scheme to show its security strength, and the performance evaluations are given by comparing the operational cost of three cases that apply three different signature algorithms, that is, ElGamal, Schnorr, and DSA.

The remainder of this paper is organized as follows. In Section 2, we introduce the network architecture, security requirements, and the design goal. In Section3, we recall

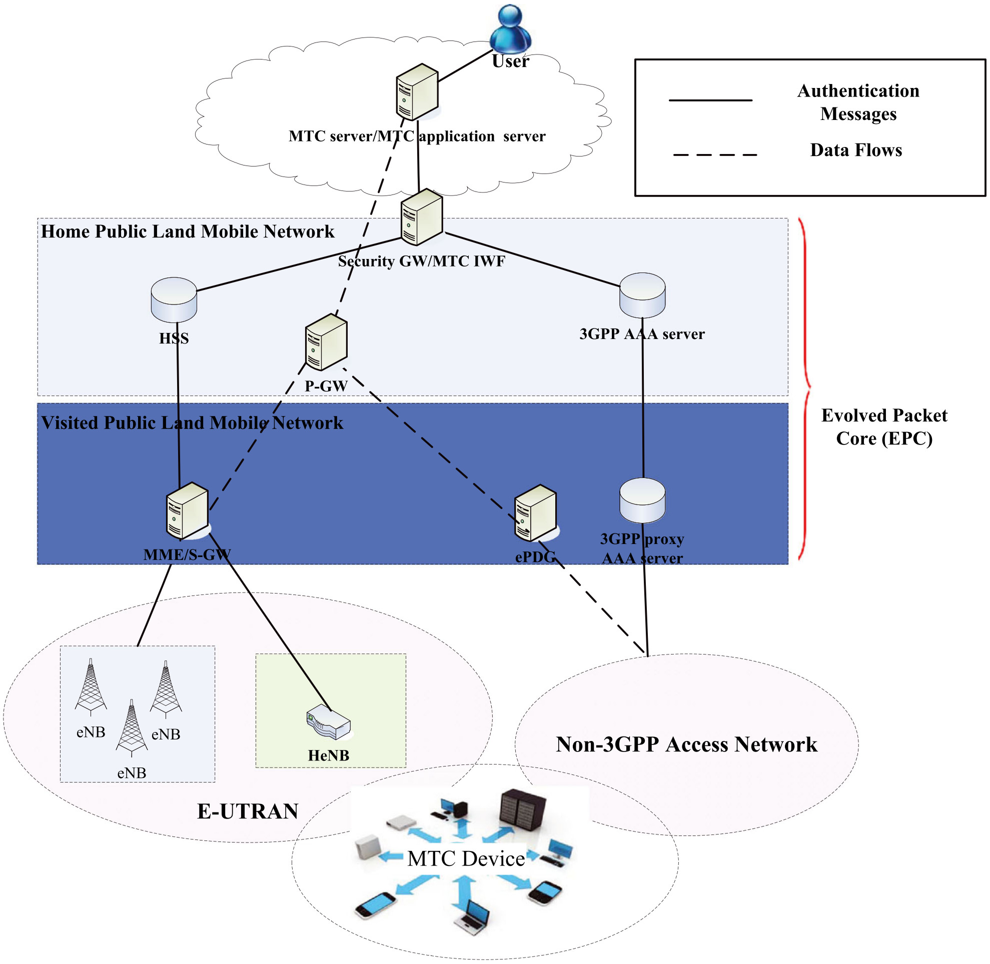


Figure 3. Network architecture.

the proxy signature technique as the preliminaries. Then, we present the E^2SEC scheme, followed by its security analysis, and implementation and performance evaluation in Sections 4 and 5, respectively. Finally, we draw the conclusions in Section 6.

2. NETWORK ARCHITECTURE, SECURITY REQUIREMENTS AND DESIGN GOALS

In this section, we will introduce the network architecture, security requirements, and identify our design goals.

2.1. Network architecture

As shown in Figure 3, MTCD can firstly access the E-UTRAN or non-3GPP access networks, and then remotely communicate with the MTC server (or MTC application server) via the evolved packet core (EPC). Typically, an LTE network mainly consists of an EPC and several access networks, that is, E-UTRAN. In addition, LTE also supports non-3GPP access networks connected to the EPC and there are multiple types of non-3GPP access networks, for example, WLAN and WiMAX.

Our network architecture is based on the 3GPP standard, and can be divided into three domains: (i) Access Network Domain, which consists of E-UTRAN and other non-3GPP access networks; (ii) the EPC, including roaming network domain, i.e., visited public land mobile network (VPLMN) and home network domain, i.e., home public land mobile network (HPLMN); (iii) non-3GPP Domain, for example, the Internet. In the proposed E^2SEC scheme, the EPC could be viewed as a whole, and thus, we do not introduce the entities located in it. The MTC security gateway (GW) is used between the MTCS and the EPC as the first point of entry into a secure operator network. Note that the MTC security GW can be an independent node or colocated with an intermediate node, for example, MTC interworking function (MTC IWF). In this paper, the security GW performs access control functionality in order to prevent the unauthorized MTCS from accessing the EPC, and it can authenticate with the MTCS in place of the 3GPP network operator. In the network architecture, MTCS colocates with MTC application server, and they are deployed outside the EPC.

2.2. Security requirements

In our security model, we assume that all the entities located in the EPC are trustworthy, but may be curious. Firstly, both the MTCD and MTCD are distrustful for the EPC, and thus, they need to be authenticated by the EPC. Secondly, the relationship between the MTCD and MTCS is distrustful as well. In addition, there exists an adversary $\mathcal{A}$ residing in the network (inside or outside the EPC) and forwards the authentication messages between the MTCD and MTCS. Because the adversary $\mathcal{A}$ can eavesdrop, modify, or replay the authentication messages, he or she could launch some active attacks to break the authentication and secure channel establishment procedures. Therefore, the following security requirements should be fulfilled for a secure MTC.

- *Entity Mutual Authentication*. Firstly, the MTCD and MTCS must access the EPC securely by adopting specified security mechanisms. In addition, a successful mutual authentication requires that all the authentication messages that are interacted by two legal entities (i.e., MTCD and MTCS) have not been altered during the transmission; that is, if the adversary $\mathcal{A}$ forges and/or modifies the authentication messages, all the malicious operations could be detected. In this way, there can be a successful mutual authentication performed between the MTCD and MTCS.
- *Secure Channel Establishment and Key Forward and Backward Secrecy*. After the successful authentication, a secure channel should be established, and the whole procedure should make sure key forward/backward secrecy (KFS/KBS). The former requires that even if the adversary $\mathcal{A}$ can eavesdrop, modify, or replay the authentication messages, he or she can not also obtain the final secure key between the two legal entities or share a key with the legal entities without them being aware of that. For the latter requirement, forward secrecy implies that a compromise of the current key should not compromise any future key, while backward secrecy means that a compromise of the current key should not compromise any earlier key.

2.3. Design goals

Under the aforementioned network architecture and security requirements, our design goal is to develop a generic and efficient security scheme for secure MTC in LTE networks. Specifically, the following goals should be achieved.

- *The security requirements should be guaranteed in the proposed scheme*. As stated earlier, if the MTC technology does not consider the security, they cannot be applied in real communication scenarios. As a result, the proposed E^2SEC scheme should achieve the entity mutual authentication, secure channel establishment, and KFS/KBS simultaneously.
- *The security between the MTCD and the MTCS does not depend on the security policy of network domain where the MTCD is visiting*. The secure communication between the MTCD and MTCSr does not depend on security features defined in the network domain where it is visiting; that is, they can efficiently perform mutual authentication and communicate with each other securely no matter where the MTCD is visiting (i.e., in the home network domain or roaming network domain).

- *The security between the MTCD and the MTCS does not depend on the specific radio access network technology*. This objective means that no matter what kind of radio access technologies is used by the MTC device (i.e., E-UTRAN, WLAN, or WiMAX), the MTC device and MTC server can perform mutual authentication without regard to the either security features or the architecture of each radio access network.

3. PROPOSED E^2SEC SCHEME

In this section, we present our E^2SEC scheme, which consists of the mutual authentication and key agreement protocols between (i) MTCS and the EPC, (ii) MTCD and the EPC, and finally (iii) MTCD and MTCS. In the considered network architecture, the MTCD must confirm whether an MTCS wants to communicate with it as a valid server that has been verified by the MTCD's EPC operator. Therefore, the MTCD should authenticate the MTCS in company with the corresponding EPC. Similarly, the MTCS needs to authenticate the MTCD with the assistance of the EPC. Proxy signature technique provides an outstanding way of delegating and verifying among entities. Before going to the details, we first recall the proxy signature technique, which serves as the basis of the proposed E^2SEC scheme.

3.1. Proxy signature

A proxy signature protocol [12] allows an entity, called the designator or original signer, to delegate another entity, called a proxy signer, to sign messages on its behalf. The proxy signature primitive and the first efficient solution were introduced by Mambo, Usuda and Okamoto [25]. Furthermore, many extensions of the basic proxy signature primitive have been considered. Recently, Boldyreva *et al*. [4] propose a scheme to prove the security of proxy signature schemes by designing a formal model. Meanwhile, they modify the Kim–Park–Won scheme [13], preserving its efficiency, and prove that the resulting scheme is secure in the random-oracle model. Generally, a proxy signature scheme is a tuple that embraces PS $= (\mathcal{G}, \mathcal{K}, \mathcal{S}, \mathcal{V}, (\mathcal{D}, \mathcal{P}), \mathcal{PS}, \mathcal{PV}, \mathcal{ID})$, where the constituent algorithms run in polynomial time; a digital signature scheme DS $= (\mathcal{G}, \mathcal{K}, \mathcal{S}, \mathcal{V})$, and other components defined as follows.

- $(\mathcal{D}, \mathcal{P})$ is a pair of interactive randomized algorithms forming the proxy-designation protocol. The input to each algorithm includes two public keys pk_i, pk_j for the designator i and the proxy signer j, respectively. $\mathcal{D}$ also takes as input the secret key sk_i of the designator, the identity j of the proxy signer, and a message space descriptor ω for which user i wants to delegate its signing rights to user j. $\mathcal{P}$ also takes as input the secret key sk_j of the proxy signer. As a result of the interaction, the expected local output of $\mathcal{P}$ is skp, a proxy-signing key that user j uses to produce proxy signatures on behalf of user i, for messages in ω. $\mathcal{D}$ has no local output. We can write $skp \leftarrow [\mathcal{D}(pk_i, sk_i, j, pk_j, \omega), \mathcal{P}(pk_j, sk_j, pk_i)]$ for the result of this interaction.
- $\mathcal{PS}$ is the randomized proxy signing algorithm. It takes as input a proxy signing key skp and a message $M \in \{0,1\}^*$, and outputs a proxy signature $p\sigma$.
- $\mathcal{PV}$ is the deterministic proxy verification algorithm. It takes as input a public key pk, a message $M \in \{0,1\}^*$, and a proxy signature $p\sigma$, and outputs 0 or 1. In the latter case, we say that $p\sigma$ is a valid proxy signature for M relative to pk.
- $\mathcal{ID}$ is the proxy identification algorithm. It takes as input a valid proxy signature $p\sigma$, and outputs an identity $i \in \mathbb{N}$ or **Stop** in case of an error.

3.2. System initialization

In the system initialization phase, the 3GPP LTE system selects two large prime number p and q such that $q|(p-1)$, a generator $g \in \mathbb{Z}_p^*$ with order q, and one way hash function $h(\cdot)$ that is assumed to be public. EPC has its private key $x_{EPC} \in \mathbb{Z}_q^*$ and public key $v_{EPS} \equiv g^{x_{EPC}} \mod p$. At the beginning, both the MTCD and MTCS have not been authenticated by the EPC. When they want to access the EPC, they must perform access authentication procedures with the EPC (Sections 3.3 and 3.4). Table I shows the notations used in the proposed E^2SEC scheme.

3.3. Authentication between MTC device and EPC

The authentication between the MTCD and the EPC can be performed by 3GPP AKA protocol (e.g., EPS-AKA). After the AKA procedure, the confidentiality key (CK) can be computed by the MTCD and the EPC. In the last message of 3GPP AKA, the EPC selects a random number $k_{E-D} \in \mathbb{Z}_q^*$ and calculates $K_{E-D} \equiv g^{k_{E-D}} \mod p$. After that, the proxy signature, $(\sigma_{E-D}, m_{E-D}, K_{E-D})$ would be encrypted with CK and sent to the MTCD. The proxy signature generation with warrant is similar to [13] as shown

Table I. The notations used in the proposed scheme.

Notation	Definition
m_{x-y}	x's delegation information sent to y
K_{x-y}	the delegation authentication token of x to y
T_{valid}	delegation signature's valid period
σ_{x-y}	x delegates his or her signing power to y
σ'_{x-y}	y calculates his or her own alternative proxy
$h(\cdot)$	strong one-way hash function
$x_x()$	x's private key
$v_x()$	x's public key
ID_x	x's identity
M_x	x's authentication message

in Equation (1).

$$\begin{aligned} e_1 &= h(m_{E-D}||K_{E-D}) \\ \sigma_{E-D} &\equiv e_1 \cdot x_{EPC} + k_{E-D} \mod q \end{aligned} \tag{1}$$

where $m_{E-D} = (ID_{EPC}||ID_{MTCD}||T_{valid})$.

Upon receipt of message $(\sigma_{E-D}, m_{E-D}, K_{E-D})$, the MTCD checks its validity by computing Equation (2).

$$g^{\sigma_{E-D}} \equiv v_{EPC}^{e_1} K_{E-D} \mod p \tag{2}$$

After the successful verification, the MTCD stores the EPC's proxy signature.

3.4. Authentication between MTC Server and EPC

According to 3GPP TR 33.868 [30], the security GW can perform access control functionality to prevent the unauthorized MTCS from accessing the EPC. It can authenticate with MTCS on behalf of the 3GPP network operator. The Network Domain Security (NDS/IP) security mechanism or private protection mechanism can protect the trigger indication sent from the MTCS to the security GW.

Upon successful authentication, the EPC selects a random $k_{E-S} \in_q^*$ and computes $K_{E-S} \equiv g^{k_{E-S}}$ *mod p*. After that, it issues proxy signature with warrant m_{E-S} $(\sigma_{E-S}, m_{E-S}, K_{E-S})$ to the authenticated MTCS through the secure channel. Similarly, the proxy signature generation with warrant is similar to Section 3.3 as shown in Equation (3).

$$\begin{aligned} e_2 &= h(m_{E-S}||K_{E-S}) \\ \sigma_{E-S} &\equiv e_2 \cdot x_{EPC} + k_{E-S} \mod q \end{aligned} \tag{3}$$

where $m_{E-S} = (ID_{EPC}||ID_{MTCS}||T_{valid})$.

Upon receipt of the message, $(\sigma_{E-S}, m_{E-S}, K_{E-S})$, The MTCS checks its validity by computing Equation (4).

$$g^{\sigma_{E-S}} \equiv v_{EPC}^{e_2} K_{E-S} \mod p \tag{4}$$

After successful verification, the MTCS stores the EPC's proxy signature.

3.5. Security scheme between machine-types communication device and machine-types communication server

When the MTCD and MTCS have performed successful mutual authentication with the EPC, they can communicate with each other directly. Specifically, the authentication between MTCD and the EPC need not synchronize with the authentication between the MTCS and the EPC. Moreover, the authentication can be initiated by the MTCD or MTCS, and it depends on different application scenarios. Without loss of generality, we assume that the MTCS initiates the authentication. When the MTCS wants to communicate with the corresponding MTCD, it first queries the location of the MTCD with the assistance of the EPC. Upon obtaining the location of the MTCD, no matter where the MTCD is visiting, the MTCS can initiate the authentication procedure with the MTCD immediately. All entities in the network (located in the home network domain or roaming network domain) just forward the authentication messages from the MTCS to the MTCD and do not apply any security policy to these messages. Figure 4 shows the mutual authentication and key agreement procedure between the MTCD and MTCS, and the specific steps are as follows.

(1) The MTCS proves its authenticity toward the MTCD by using its proxy signature received in Section 3.4, that is, $(\sigma_{E-S}, m_{E-S}, K_{E-S})$. Firstly, the MTCS generates random number a and computes g^a for key agreement. Then, the MTCS uses

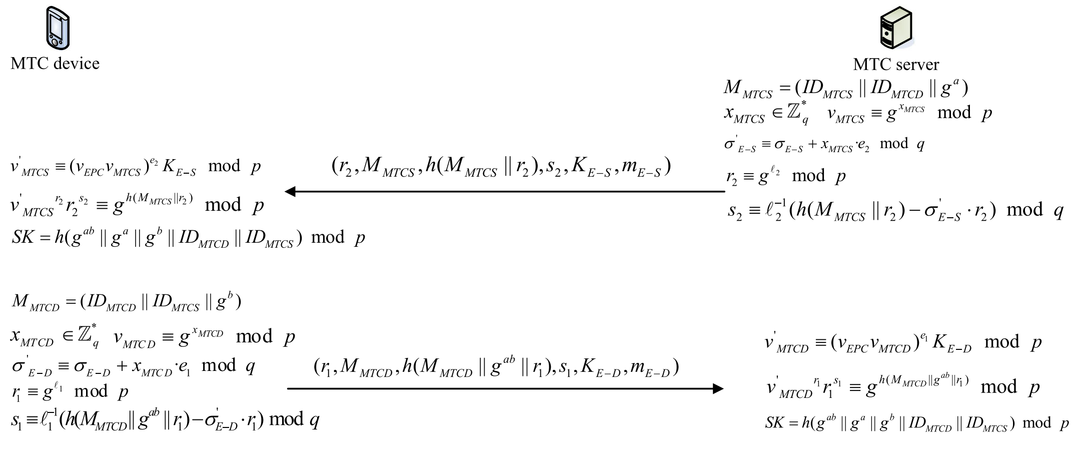

MTC device
MTC server
$M_{MTCS} = (ID_{MTCS} \| ID_{MTCD} \| g^a)$
$x_{MTCS} \in \mathbb{Z}_q^* \quad v_{MTCS} \equiv g^{x_{MTCS}} \mod p$
$\sigma'_{E-S} \equiv \sigma_{E-S} + x_{MTCS} \cdot e_2 \mod q$
$r_2 \equiv g^{\ell_2} \mod p$
$s_2 \equiv \ell_2^{-1}(h(M_{MTCS} \| r_2) - \sigma'_{E-S} \cdot r_2) \mod q$
$(r_2, M_{MTCS}, h(M_{MTCS} \| r_2), s_2, K_{E-S}, m_{E-S})$
$v'_{MTCS} \equiv (v_{EPC} v_{MTCS})^{e_2} K_{E-S} \mod p$
${v'_{MTCS}}^{r_2} r_2^{s_2} \equiv g^{h(M_{MTCS} \| r_2)} \mod p$
$SK = h(g^{ab} \| g^a \| g^b \| ID_{MTCD} \| ID_{MTCS}) \mod p$
$M_{MTCD} = (ID_{MTCD} \| ID_{MTCS} \| g^b)$
$x_{MTCD} \in \mathbb{Z}_q^* \quad v_{MTCD} \equiv g^{x_{MTCD}} \mod p$
$\sigma'_{E-D} \equiv \sigma_{E-D} + x_{MTCD} \cdot e_1 \mod q$
$r_1 \equiv g^{\ell_1} \mod p$
$s_1 \equiv \ell_1^{-1}(h(M_{MTCD} \| g^{ab} \| r_1) - \sigma'_{E-D} \cdot r_1) \mod q$
$(r_1, M_{MTCD}, h(M_{MTCD} \| g^{ab} \| r_1), s_1, K_{E-D}, m_{E-D})$
$v'_{MTCD} \equiv (v_{EPC} v_{MTCD})^{e_1} K_{E-D} \mod p$
${v'_{MTCD}}^{r_1} r_1^{s_1} \equiv g^{h(M_{MTCD} \| g^{ab} \| r_1)} \mod p$
$SK = h(g^{ab} \| g^a \| g^b \| ID_{MTCD} \| ID_{MTCS}) \mod p$

Figure 4. Mutual authentication and key agreement procedure between machine-type communications (MTC) device and MTC server.

ElGamal signature scheme to sign message $M_{MTCS} = (ID_{MTCS}||ID_{MTCD}||g^a)$. Its private key is $x_{MTCS} \in_q^*$ and public key $v_{MTCS} \equiv g^{x_{MTCS}} \mod p$. The specific steps are as follows:

- **Step 1.** The MTCS calculates an alternative proxy (σ'_{E-S}, K_{E-S}):

$$\sigma'_{E-S} \equiv \sigma_{E-S} + x_{MTCS} \cdot e_2 \mod q \qquad (5)$$

- **Step 2.** It generates a random number $\ell_2 \in_q^*$ and computes

$$r_2 \equiv g^{\ell_2} \mod p \qquad (6)$$

$$s_2 \equiv \ell_2^{-1}\left(h(M_{MTCS}||r_2) - \sigma'_{E-S} \cdot r_2\right) \mod q \qquad (7)$$

- **Step-3.** It sends the message $(r_2, M_{MTCS}, h(M_{MTCS}||r_2), s_2, K_{E-S}, m_{E-S})$ to the MTCD.

(2) When the MTCD receives the message $(r_2, M_{MTCS}, h(M_{MTCS}||r_2), s_2, K_{E-S}, m_{E-S})$, it computes Equation (8) and verifies Equation (9).

$$v'_{MTCS} \equiv (v_{EPC}v_{MTCS})^{e_2}K_{E-S} \mod p \qquad (8)$$

$$v'^{r_2}_{MTCS}r_2^{s_2} \equiv g^{h(M_{MTCS}||r_2)} \mod p \qquad (9)$$

Upon verification passes, the MTCD can verify (i) the MTCS has been authenticated by the EPC and the period of validity authorized by the EPC is still valid and (ii) it has performed a correct authentication with the MTCs that wants to communicate with it.

(3) When the verification succeeds, the MTCD proves its authenticity toward the MTCS by using its proxy signature received in Section 3.3, $(\sigma_{E-D}, m_{E-D}, K_{E-D})$. Similarly, the MTCD generates random number b and computes g^b for key agreement. Then, the MTCD uses an ElGamal signature scheme to sign message $M_{MTCD} = (ID_{MTCD}||ID_{MTCS}||g^b)$. Its private key is $x_{MTCD} \in_q^*$ and public key $v_{MTCD} \equiv g^{x_{MTCD}} \mod p$.

- **Step 1.** The MTCD calculates an alternative proxy (σ'_{E-D}, K_{E-D}):

$$\sigma'_{E-D} \equiv \sigma_{E-D} + x_{MTCD} \cdot e_1 \mod q \qquad (10)$$

- **Step 2.** It generates a random number $\ell_2 \in_q^*$ and computes

$$r_1 \equiv g^{\ell_1} \mod p \qquad (11)$$

$$s_1 \equiv \ell_1^{-1}\left(h\left(M_{MTCD}||g^{ab}||r_1\right) - \sigma'_{E-D} \cdot r_1\right) \mod q \qquad (12)$$

- **Step 3.** It sends $(r_1, M_{MTCD}, h(M_{MTCD}||g^{ab}||r_1), s_1, K_{E-D}, m_{E-D})$ to the MTCS.

(4) The MTCS receives $(r_1, M_{MTCD}, h(M_{MTCD}||g^{ab}||r_1), s_1, K_{E-D}, m_{E-D})$, computes Equation (13), and verifies Equation (14).

$$v'_{MTCD} \equiv (v_{EPC}v_{MTCD})^{e_1}K_{E-D} \mod p \qquad (13)$$

$$v'^{r_1}_{MTCD}r_1^{s_1} \equiv g^{h(M_{MTCD}||g^{ab}||r_1)} \mod p \qquad (14)$$

If the verification is successful, the MTCS can verify (i) the corresponding MTCD has been authenticated by the EPC and the period of validity authorized by the EPC is still valid and (ii) it has performed a correct authentication with the MTCD that it wants to communicate with.

(5) Once they perform a successful mutual authentication, the session key SK between them can be derived by Diffie–Hellmen (DH) key agreement as shown in Equation (15).

$$SK = h\left(g^{ab}||g^a||g^b||ID_{MTCD}||ID_{MTCS}\right) \mod p \qquad (15)$$

4. SECURITY ANALYSIS

In this section, we analyze the security properties of the proposed E^2SEC scheme. In particular, following the security requirements discussed earlier, our analysis will focus on how the proposed E^2SEC scheme can achieve the mutual authentication, secure channel establishment, and key forward and backward secrecy.

4.1. Security analysis

Firstly, our proposed E^2SEC scheme satisfies the following proxy signature properties:

Verifiability: From the proxy signatures of the MTCS and MTCD, both of them can be convinced of the original signer's (i.e., the EPC) agreement on the signed messages. Therefore, the proposed E^2SEC scheme can fulfill verifiability.

Strong unforgeability: Only the delegated MTCS or MTCD can generate the legal proxy-signed signature by using $\sigma_{E-D}/\sigma_{E-S}$ on behalf of the EPC. Because the computing $\sigma_{E-D}/\sigma_{E-S}$ are based on the discrete logarithm problem, it is infeasible for an adversary to break them. Therefore, the proposed E^2SEC scheme can fulfill strong unforgeability.

Strong identifiability: In our proposed scheme, the public key of the original signer (i.e., the EPC) is used to verify a signature. Therefore, the verifier, that is, the MTCS or MTCD, knows a valid signature of the signer after the verification of the signature. In addition, as the original signer, the EPC can also identify the MTCS and MTCD by checking the proxy-signed signatures s_1 and s_2. Hence, the proposed E^2SEC scheme can fulfill strong identifiability.

Strong undeniability: Once the proxy-signed signatures $(r_2, M_{MTCS}, h(M_{MTCS}||r_2), s_2, K_{E-S}, m_{E-S})$ and $(r_1, M_{MTCD}, h(M_{MTCD}||r_1), s_1, K_{E-D}, m_{E-D})$ are verified, the warrant m_{E-S} and m_{E-D} are checked. In addition, the original signer's public key, v_{EPC}, and proxy signer's own public keys, $v^{'}_{MTCD}$ and $v^{'}_{MTCS}$, are used. The signer, that is, the MTCD/MTCS, cannot deny its signature to be sent to the MTCS/MTCD. Therefore, the proposed E^2SEC scheme satisfies the strong undeniability property.

Prevention of misuse: In the proposed scheme, the proxy signing key $\sigma_{E-D}/\sigma_{E-S}$ is only used for the authentication and key agreement between the MTCD and MTCS. Both of them cannot use their own proxy-signing key for other purposes.

Next, we first analyze the security requirements discussed earlier in an informal way, and then a formal verification will be given.

Entity mutual authentication. *The primary goal of our proposed E^2SEC scheme is to provide mutual authentication between the MTCD and the MTCS.*

Firstly, both the MTCD and the MTCS need to be authenticated by the EPC when they want to access the EPC. After the successful authentication, the EPC generates the corresponding proxy-signing key $\sigma_{E-D}/\sigma_{E-S}$ for the MTCD and the MTCS, respectively. Then, the MTCD and the MTCS use the EPC's public key v_{EPC} and their own public keys, that is, $v^{'}_{MTCD}$ and $v^{'}_{MTCS}$, to sign their own message. Upon successful verification of the signed messages, it demonstrates that (i) the corresponding entity has been authenticated by the EPC and the period of validity authorized by the EPC is still valid, and (ii) they have performed a correct authentication with each other with which they want to communicate.

Secure key agreement and key forward/backward secrecy (***KFS/KBS***)). *Another important goal of our proposed E^2SEC scheme is to provide secure key agreement services between the MTCD and the MTCS and guarantee KFS/KBS.*

(1) After performing the mutual authentication successfully, the MTCD and MTCS can generate the session key $SK = h(g^{ab}||g^a||g^b||ID_{MTCD}||ID_{MTCS}) \mod p$ through using the DH algorithm based on the discrete logarithm problem. However, the basic DH protocol is insecure and vulnerable to man-in-the-middle (MITM) attack. Krawczyk [14] proposes a provable secure and efficient DH key exchange approach, called SIGn-and-MAc (SIGMA) to solve this problem. In this paper, we design our key agreement procedure based on the criterion in [14]. Hence, a secure channel can be established between the MTCD and MTCS.

(2) The KFS implies that a compromise of the current key should not compromise any future key, while KBS means that a compromise of the current key should not compromise any earlier key. To provide KFS and KBS between the MTCS and MTCD, our proposed E^2SEC scheme adopts the DH key exchange. Because the DH secret keys are generated by the random values of the MTCS and MTCD, they can guarantee the freshness of the DH session key if two entities have chosen their random exponents properly.

Withstanding protocol attacks: Basically, our key agreement is similar to the DH key exchange. However, an MitM attacker is prevented because we design our scheme followed by SIGn-and-MAc (SIGMA) [14], which is a provable secure and efficient DH key exchange approach proposed by Krawczyk. The proposed E^2SEC scheme can provide KFS and KBS and resist an MitM attack by combining the proxy-signatures. An attacker without possessing proxy signature from the EPC could not masquerade as a valid MTCD or MTCS.

4.2. Formal verification

4.2.1. ProVerif.

We will use ProVerif to verify the security of our protocol. ProVerif is a tool for automatically analyzing the security of cryptographic protocols, which is provided for, but not limited to, cryptographic primitives including symmetric and asymmetric encryption, digital signatures, hash functions, bit-commitment, and non-interactive zero-knowledge proofs. ProVerif is capable of proving reachability properties, correspondence assertions, and observational equivalence. These capabilities are particularly useful to the computer security domain because they permit the analysis of secrecy and authentication properties. In addition, emerging properties such as privacy, traceability, and verifiability can also be considered. Protocol analysis is considered with respect to an unbounded number of sessions and an unbounded message space. Furthermore, the tool is capable of attack reconstruction, that is, when a property cannot be proved, ProVerif tries to reconstruct an execution trace that falsifies the desired property.

Cryptographic primitives are modeled as functions, and messages are represented by terms built over an infinite set of names *a, b, c, …*, an infinite set of variables *x,*

Table II. Main process grammar.

$P, Q ::=$	Processes
0	Null process
$P \mid Q$	Parallel composition
$!P$	Replication
$new \quad n; P$	Name restriction
$in(M, x); P$	Message input
$out(M, N); P$	Message output
$if \quad M = N \quad then \quad P \quad else \quad Q$	Conditional
$let \quad M = D \quad in \quad P \quad else \quad Q$	Term evaluation
$R(M_1, ..., M_k)$	Macro usage

y, *z*, …, and a finite set of function symbols $f_1 ..., f_n$. Function symbols represent cryptographic primitives that can be applied to messages. The effect of applying function symbols to terms is described by a set of reduction rules. The syntax of ProVerif calculus processes is given in Table II [3]. ProVerif can be run under Windows or Linux/Mac. In this paper, we conduct the experiments with ProVerif running on a 2.30 GHz-processor 4 GB-memory computing machine to test the proposed E^2SEC protocol under Windows[†].

4.2.2. Specification of our scheme.

The primary goal of our proposed protocol is to provide mutual authentication and key agreement services between the MTCD and MTCS. The ability of our protocol to resist the typical attacks has been discussed in Section 4.1. Thus, the main security goals to be verified are as follows, and their individual specific requirements have been described in Section 4.1.

- *Mutual authentication between the MTCD and MTCS*;
- *Secrecy of SK*;
- *KFS/KBS*.

We formalize the basic cryptographic primitives used by the E^2SEC scheme as follows. Digital signature is defined in Table III, and the DH key agreement is given in Table IV.

We further model three security goals in this paper:

(1) *Mutual authentication between MTCD and the MTCS*: We declare the events:

- **event** acceptsMTCDparam(spkey), which is used by the MTCD to record the belief that it has accepted to run the protocol with the MTCS, with the MTCS's public key as the first argument.
- **event** termMTCDparam(spkey), which denotes the MTCD's belief that it has terminated a protocol run with the MTCS, with the MTCS's public key as the first argument.
- **event** acceptsMTCSparam(spkey), which is used by the MTCS to record the belief that it has accepted to run the protocol with the MTCD and the MTCD's public key as the first argument.
- **event** termMTCSparam(spkey), which denotes the MTCS's belief that it has terminated a protocol run with the MTC device with the MTC device's public key as the first argument.

Table III. Digital signatures.

1. **type** sskey.
2. **type** spkey.
3. **fun** spk (sskey) : spkey.
4. **fun** sign (bitstring, sskey): bitstring .
5. **reduc forall** m: bitstring, k: sskey; getmess (sign(m,k)) = m.
6. **reduc forall** m: bitstring, k: sskey; checksign (sign(m,k), spk (k)) = m.

Table IV. Diffie–Hellman key agreement.

1. **type** G.
2. **type** exponent.
3. **const** g: G [**data**].
4. **fun** exp (G, exponent): G.
5. **equation forall** x: exponent, y: exponent; exp(exp(g, x), y)=exp(exp(g, y), x).

Next, we use the correspondence assertion **event** (**termMTCD**(spkey))==>**event**(**acceptsMTCS** (spkey)), and **event**(**termMTCS**(spkey))==>**event** (**acceptsMTCD** (spkey)) to test if E^2SEC can achieve mutual authentication.

(2) *Secrecy of SK* and *KFS/KBS*: We first define a **query** attacker (s), where *s* is session key shared between the MTCD and the MTCS. Internally, ProVerif attempts to prove that a state in which the session key *s* is known to the adversary is unreachable (that is, it tests the query **not** attacker, and the query is true when the *s* is *not* derivable by the adversary).

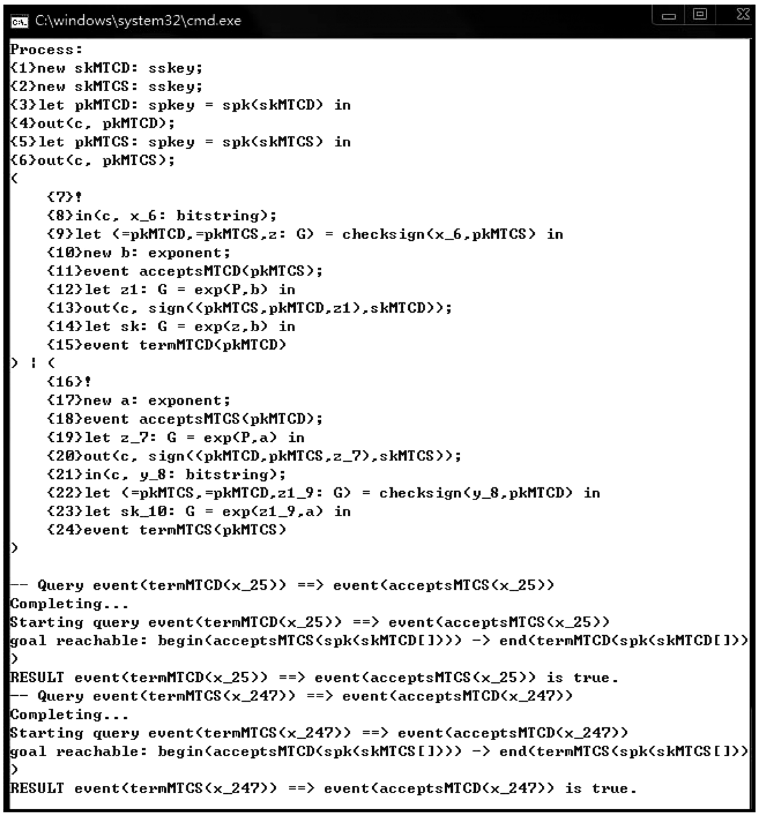
4.2.3. Results of analysis.

The verification results are shown in Figures 5 and 6. Firstly, in Figure 5, the verification result shows that RESULT event(termMTCD(25)) ==> event(acceptsMTCS(25)) is true and that RESULT event(termMTCS(247)) ==> event(acceptsMTCD(247)) is true. We can conclude that there has been a successful mutual authentication between the MTCD and MTCS. In addition, in Figure 6, the verification result shows RESULT not attacker_p1 (s[ ]) is true. It manifests that the secrecy of *SK* and KFS/KBS hold.

5. IMPLEMENTATION AND PERFORMANCE EVALUATION

In this section, we first present some implementation considerations of the E^2SEC scheme, that is, flexible signature algorithms selection. Moreover, we analyze the performance of the proposed E^2SEC scheme in terms of operational cost, and compare the operational cost

[†] User manual and tutorial can be downloaded in http://prosecco.gforge.inria.fr/personal/bblanche/proverif/manual.pdf



```
Process:
{1}new skMTCD: sskey;
{2}new skMTCS: sskey;
{3}let pkMTCD: spkey = spk(skMTCD) in
{4}out(c, pkMTCD);
{5}let pkMTCS: spkey = spk(skMTCS) in
{6}out(c, pkMTCS);
(
    {7}!
    {8}in(c, x_6: bitstring);
    {9}let (=pkMTCD,=pkMTCS,z: G) = checksign(x_6,pkMTCS) in
    {10}new b: exponent;
    {11}event acceptsMTCD(pkMTCS);
    {12}let z1: G = exp(P,b) in
    {13}out(c, sign((pkMTCS,pkMTCD,z1),skMTCD));
    {14}let sk: G = exp(z,b) in
    {15}event termMTCD(pkMTCD)
) | (
    {16}!
    {17}new a: exponent;
    {18}event acceptsMTCS(pkMTCD);
    {19}let z_7: G = exp(P,a) in
    {20}out(c, sign((pkMTCD,pkMTCS,z_7),skMTCS));
    {21}in(c, y_8: bitstring);
    {22}let (=pkMTCS,=pkMTCD,z1_9: G) = checksign(y_8,pkMTCD) in
    {23}let sk_10: G = exp(z1_9,a) in
    {24}event termMTCS(pkMTCS)
)

-- Query event(termMTCD(x_25)) ==> event(acceptsMTCS(x_25))
Completing...
Starting query event(termMTCD(x_25)) ==> event(acceptsMTCS(x_25))
goal reachable: begin(acceptsMTCS(spk(skMTCD[]))) -> end(termMTCD(spk(skMTCD[]))
)
RESULT event(termMTCD(x_25)) ==> event(acceptsMTCS(x_25)) is true.
-- Query event(termMTCS(x_247)) ==> event(acceptsMTCD(x_247))
Completing...
Starting query event(termMTCS(x_247)) ==> event(acceptsMTCD(x_247))
goal reachable: begin(acceptsMTCD(spk(skMTCS[]))) -> end(termMTCS(spk(skMTCS[]))
)
RESULT event(termMTCS(x_247)) ==> event(acceptsMTCD(x_247)) is true.
```

Figure 5. Verification result of mutual authentication between machine-type communications (MTC) device and MTC server.

of three cases that apply three different signature algorithms, that is, ElGamal, Schnorr, and DSA. We call these three implementations E-ElGamal, E-Schnorr and E-DSA, respectively.

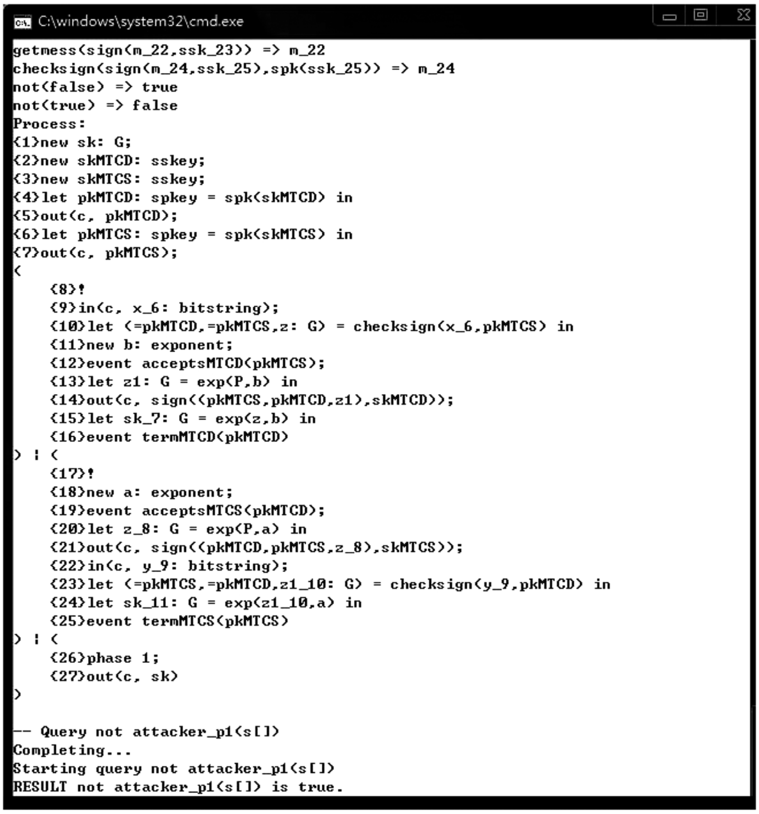
5.1. Implementation

In our proposed E^2SEC scheme, the MTCD and the MTCS can trust each other by performing mutual authentication using the proxy signatures issued by the 3GPP core network. Initially, both the MTCD and MTCS have not been authenticated by the EPC. When they want to access the EPC, they must perform access authentication procedures with the EPC by standard AKA. The authentication between the MTC device and EPC can be performed by the 3GPP AKA protocol (e.g., EPS-AKA). The security GW can perform access control functionality to prevent the unauthorized MTCS from accessing to the EPC. It can authenticate with MTC server in behalf of the 3GPP network operator. The NDS/IP security mechanism or private protection mechanism can protect the trigger indication sent from the MTC server to the security GW. Therefore, the initial authentication should involve the EPC, and the preparation phase of the proposed E^2SEC scheme can be embedded in this procedure.

Except initial access authentication of MTCD and the MTCS, during the delegation signature's valid period, the MTCD and the MTCS can establish secure communication anywhere with little intervention from the EPC, which reduces the burden of the core network and improves efficiency. Moreover, in the mutual AKA phase, the MTCD and MTCS need to authenticate each other by using the proxy signatures issued by the EPC. We adopt the proxy signature technique together with the ElGamal signature scheme to present the secure and efficient AKA procedure between the MTCD and MTCS. In fact, both the MTCD and MTCS can choose multiple signature techniques to achieve security functions depending on their respective capabilities and the different applications. In this way, they can configure their security policies, which makes the proposed E^2SEC scheme more flexible.

5.2. Performance evaluation

In this section, we evaluate the performance of the proposed E^2SEC scheme in terms of the operational cost. Firstly, the time used for the primitive cryptography operations has been measured by using C/C++ OPENSSL library [27] tested on a Celeron 1.1 GHz processor as an MTCD and Dual-Core 2.6 GHz as an MTCS [6] in Table V.



```
getmess(sign(m_22,ssk_23)) => m_22
checksign(sign(m_24,ssk_25),spk(ssk_25)) => m_24
not(false) => true
not(true) => false
Process:
{1}new sk: G;
{2}new skMTCD: sskey;
{3}new skMTCS: sskey;
{4}let pkMTCD: spkey = spk(skMTCD) in
{5}out(c, pkMTCD);
{6}let pkMTCS: spkey = spk(skMTCS) in
{7}out(c, pkMTCS);
(
    {8}!
    {9}in(c, x_6: bitstring);
    {10}let (=pkMTCD,=pkMTCS,z: G) = checksign(x_6,pkMTCS) in
    {11}new b: exponent;
    {12}event acceptsMTCD(pkMTCS);
    {13}let z1: G = exp(P,b) in
    {14}out(c, sign((pkMTCS,pkMTCD,z1),skMTCD));
    {15}let sk_7: G = exp(z,b) in
    {16}event termMTCD(pkMTCD)
) | (
    {17}!
    {18}new a: exponent;
    {19}event acceptsMTCS(pkMTCD);
    {20}let z_8: G = exp(P,a) in
    {21}out(c, sign((pkMTCD,pkMTCS,z_8),skMTCS));
    {22}in(c, y_9: bitstring);
    {23}let (=pkMTCS,=pkMTCD,z1_10: G) = checksign(y_9,pkMTCD) in
    {24}let sk_11: G = exp(z1_10,a) in
    {25}event termMTCS(pkMTCS)
) | (
    {26}phase 1;
    {27}out(c, sk)
)

-- Query not attacker_p1(s[])
Completing...
Starting query not attacker_p1(s[])
RESULT not attacker_p1(s[]) is true.
```

Figure 6. Verification result of secrecy of session key and key forward/key backward secrecy.

Table V. Time costs of the primitive cryptography operations (1024 bits).

(ms)	T_E	T_M	T_H	T_A
MTC device	1.698	1.537	0.0356	0.0094
MTC server	0.525	0.475	0.0121	0.0033

T_E, T_M, T_H, T_A are the operational costs of the modular exponentiation, multiplication, hash, and arithmetic operation, respectively.

Table VI shows the operational cost in the proposed E^2SEC scheme. The operational cost is divided into two parts: (i) before authentication is performed, both the MTCD and MTCS can pre-compute their own alternative proxy σ'_{E-D} and σ'_{E-S}, respectively, and (ii) during the authentication, the MTCD and MTCS need to compute their own signature, and verify peer's signature. T_{pre} and T_{auth} represent the pre-computation time and the authentication operation time, respectively. The results show that the total authentication costs approximately takes 19.339 μs in the MTCD side, while it takes 5.9808 μs in the MTCS side.

Discussed in Section 5.1, in the mutual AKA phase, both the MTCD and MTCS can choose multiple signature techniques to achieve security functions. Therefore, we further evaluate the operational costs of E^2SEC scheme by implementing three signature techniques in ElGamal signature family, denoted as E-ElGamal, that is, our proposed scheme, E-Schnorr, and E-DSA, respectively.

During the AKA phase, E-Schnorr will cost $4T_E$+$3T_M$+$2T_H$+T_A = 11.4836 *ms* in the MTCD, and cost $4T_E$+$3T_M$+$2T_H$+T_A = 3.5525 *ms* in the MTCS, respectively. E-DSA will cost $6T_E$+$5T_M$+$2T_H$+T_A = 17.9536 *ms* in the MTCD, and costs $6T_E$+$5T_M$+$2T_H$+T_A = 5.5525 *ms* in the MTCS, respectively. Figures 7 and 8 compare the operational costs of E^2SEC scheme by implementing three signature techniques in the MTCD and MTCS side, respectively.

5.3. Further discussion

Because supporting a massive number of MTC devices has been considered as an essential requirement in M2M communications, designing a new scheme to establish the trustful relationships between multiple MTCD and MTCS is desirable. The aggregate signature [5,10,19,24] is one technique towards achieving this goal. In aggregate signature schemes, multiple signatures can be aggregated into a compact aggregate signature, even if these signatures are on (many) different documents and are produced by (many) different signers. Apart from compactness, aggregate signatures have another advantage that can prevent a malicious party from removing a signature from a collec-

Table VI. The operational cost (ms).

ms	MTC device	MTC server
T_{pre}	$T_M + T_A = 1.5464$	$T_M + T_A = 0.4783$
T_{auth}	$5T_E + 6T_M + 2T_H + T_A = 17.7926$	$5T_E + 6T_M + 2T_H + T_A = 11.5025$
T_{total}	19.339	5.9808

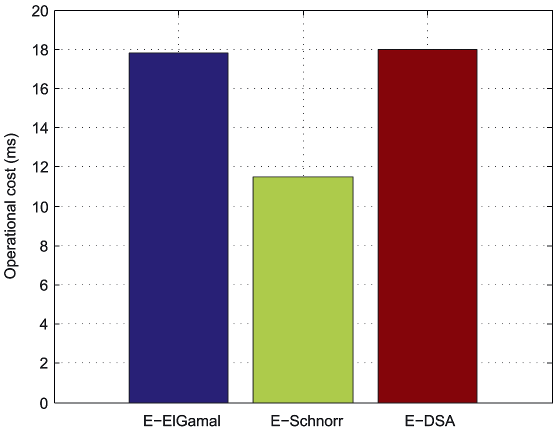


Figure 7. Operational cost in the macine-type communications device side during authentication by implementing three signature techniques.

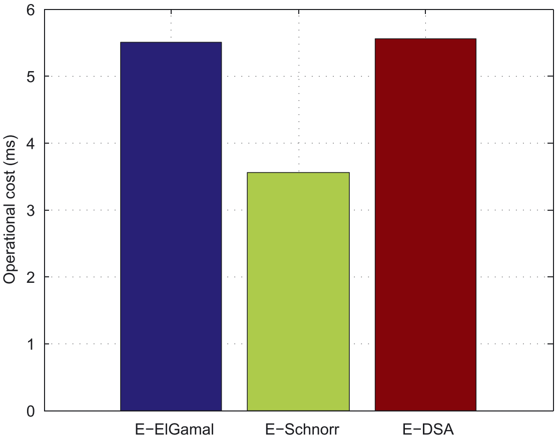


Figure 8. Operational cost in the machine-type communications server side during authentication by implementing three signature techniques.

tion of signatures without being detected. Two aggregate signature schemes exist. D. Boneh *et al.* [5] use bilinear maps and support flexible aggregation. A. Lysyanskaya *et al.* [24] use a weaker assumption, certified trapdoor permutations, but it permits only sequential aggregation. Different from these two schemes, an identity-based signature (IBS) scheme [10] is proposed by Gentry, which can improve the communication efficiency. Recently, the concept of aggregate proxy signature (APS) is first proposed in [19]. Compared with the previous schemes, APS can be easily applied to the scenario of multiple MTC devices because of its inherent property. It can provide security services and reduce computation and communication cost effectively.

6. CONCLUSION AND FUTURE WORK

In this paper, we have proposed an end-to-end security scheme for MTC by adopting the proxy-signature technique, called E^2SEC. The proposed E^2SEC scheme makes the MTCD and MTCD securely communicate with each other under the management of the 3GPP core network. In addition, theAKA between the MTCD and MTCS can be performed regardless of both security policies and architectures of each radio access network, and no matter where the MTCD is visiting, which shows its effectiveness and flexibility. The security analysis demonstrates that the proposed E^2SEC scheme can achieve the security goals, that is, entity mutual authentication, secure channel establishment, and KFS/KBS. It can also prevent various security threats. Furthermore, we discuss several implementations of E^2SEC for machine-type communications in LTE networks, that is, flexible signature algorithms selection and multiple MTCD scenario. In addition, we analyze the performance of the proposed E^2SEC scheme in terms of operational cost and compare the operational costs of three cases that apply three different signature algorithms, namely E-ElGamal, E-Schnorr, and E-DSA.

Besides the considered scenario in this paper, MTC communication could occur among MTC devices. In our future work, we will consider to design the new scheme to establish the trustful relationships among MTCDs and realize secure and efficient device-to-device communications.

ACKNOWLEDGEMENTS

Our research is financially supported by the National Natural Science Foundation of China Grant 61402366, 61472472, and the Natural Science Foundation of Shaanxi Province (Grant No. 2013JZ020), the International Science and Technology Cooperation and Exchange Plan in Shaanxi Province of China (2015KW-010). This work is also supported by a research grant from the Natural Science and Engineering Research Council (NSERC), Canada. Part of this paper was presented at the second IEEE/CIC International Conference on Communications in China (ICCC 2013).

REFERENCES

1. *Machine to machine*. http://en.wikipedia.org/wiki/Machine_to_machine.
2. Bailey D. Moving 2 mishap: M2M's impact on privacy and safety. *IEEE Security & Privacy* 2012; **10**(1): 84–87.
3. Blanchet B. *Proverif: Cryptographic protocol verifier in the formal model*. http://www.proverif.ens.fr/ [accessed on July 13, 2015].
4. Boldyreva A, Palacio A, Warinschi B. Secure proxy signature schemes for delegation of signing rights. *Journal of Cryptology* 2012; **25**(1): 57–115.
5. Boneh D, Gentry C, Lynn B, *et al.* Aggregate and verifiably encrypted signatures from bilinear maps. In *Proceedings of EUROCRYPT*, Warsaw, Poland, 2003; 416–432.
6. Cao J, Li H, Ma M, *et al.* A simple and robust handover authentication between HENB and ENB in LTE networks. *Computer Networks* 2012; **56**(8): 2119–2131.
7. ETSI. *Machine-to-machine communications (M2M); M2M service requirements*, 2011. TS 102 689 V1.1.2.
8. Fadlullah Z, Fouda M, Kato N, *et al.* Toward intelligent machine-to-machine communications in smart grid. *IEEE Communications Magazine* 2011; **49**(4): 60–65.
9. Fu H, Lin P, Yue H, *et al.* Group mobility management for large-scale machine-to-machine mobile networking. *IEEE Transactions on Vehicular Technology* 2014; **63**(3): 1296–1305.
10. Gentry C, Ramzan Z. Identity-based aggregate signatures. In *Proceedings of PKC*, New York, 2006; 257–273.
11. Gilani S. *The promise of M2M: how pervasive connected machines are fueling the next wireless evolution*, 2009. http://www.mentor.com/embedded-software/resources/overview/the-promise-of-m2m-how-pervasive-connected-machines-are-fueling-the-next-wireless-revolution-37c21bab-596c-489d-8590-a790eca5e4ce [accessed on July 13, 2015].
12. Han CK, Choi HK, Kim IH. Building femtocell more secure with improved proxy signature. In *Proceedings of IEEE GLOBECOM*, Honolulu, Hawaii, 2009; 1–6.
13. Kim S, Park S, Won D. Proxy signatures, revisited. *Information and Communications Security* 1997: 223–232.
14. Krawczyk H. SIGMA: The SIGn-and-MAcapproach to authenticated Diffie–Hellman and its use in the IKE protocols. In *Proceedings of CRYPTO*, Santa Barbara, 2003; 400–425.
15. Lai C, Li H, Li X, *et al.* A novel group access authentication and key agreement protocol for machine-type communication. *Transactions on Emerging Telecommunications Technologies* 2013; **26**(3): 1–18.
16. Lai C, Li H, Lu R, *et al.* LGTH: a lightweight group authentication protocol for machine-type communication in LTE networks. In *Proceedings of IEEE GLOBECOM*, Atlanta, GA, 2013; 832–837.
17. Lai C, Li H, Lu R, *et al.* SEGR: A secure and efficient group roaming scheme for machine to machine communications between 3GPP and WiMAX networks. In *Proceedings of IEEE ICC*, Sydney, Australia, 2014; 1011–1016.
18. Lai C, Li H, Zhang Y, *et al.* Security issues on machine to machine communications. *KSII Transactions on Internet and Information Systems* 2012; **6**(2): 498–514.
19. Li J, Kim K, Zhang F, *et al.* Aggregate proxy signature and verifiably encrypted proxy signature. *Provable security* 2007: 208–217.
20. Lien S, Chen K, Lin Y. Toward ubiquitous massive accesses in 3GPP machine-to-machine communications. *IEEE Communications Magazine* 2011; **49**(4): 66–74.
21. Lien SY, Chen KC. Massive access management for QoS guarantees in 3GPP machine-to-machine communications. *IEEE Communications Letters* 2011; **15**(3): 311–313.
22. Lien SY, Liau TH, Kao CY, *et al.* Cooperative access class barring for machine-to-machine communications. *IEEE Transactions on Wireless Communications* 2012; **11**(1): 27–32.
23. Lu R, Li X, Liang X, *et al.* GRS: The green, reliability, and security of emerging machine to machine communications. *IEEE Communications Magazine* 2011; **49**(4): 28–35.
24. Lysyanskaya A, Micali S, Reyzin L, *et al.* Sequential aggregate signatures from trapdoor permutations. In *Proceedings of EUROCRYPT*, Interlaken, Switzerland, 2004; 74–90.
25. Mambo M, Usuda K, Okamoto E. Proxy signatures: delegation of the power to sign messages. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 1996; **79**(9): 1338–1354.
26. Niyato D, Xiao L, Wang P. Machine-to-machine communications for home energy management system in smart grid. *IEEE Communications Magazine* 2011; **49**(4): 53–59.
27. OpenSSL. http://www.openssl.org/ [accessed on July 13, 2015].
28. Taleb T, Ksentini A. On alleviating MTC overload in EPS. *Ad Hoc Networks* 2014; **18**: 24–39.

29. Taleb T, Kunz A. Machine type communications in 3GPP networks: potential, challenges, and solutions. *IEEE Communications Magazine* 2012; **50**(3): 178–184.
30. 3GPP TR 33.868 V0.11.0. *Security aspects of machine-type communications*, 2012.
31. 3GPP TR 22.368 V12.1.0. *Service requirements for machine-type communications (MTC); Stage 1*, 2012.
32. 3GPP TR 23.888 V1.4.0. *System improvements for machine-type communications*, 2011.
33. Zhang Y, Yu R, Nekovee M, *et al.* Cognitive machine-to-machine communications: visions and potentials for the smart grid. *IEEE Network* 2012; **26**(3): 6–13.
34. Zheng K, Hu F, Wang W, *et al.* Radio resource allocation in LTE-advanced cellular networks with M2M communications. *IEEE Communications Magazine* 2012; **50**(7): 184–192.

AUTHORS' BIOGRAPHIES

Chengzhe Lai received his degree in B.S. in Information Security from Xi'an University of Posts and Telecommunications in 2008 and a PhD degree from Xidian University in 2014. He was a visiting PhD student with the Broadband Communications Research (BBCR) Group, University of Waterloo from 2012 to 2014. At present, he is with the School of Telecommunication and Information Engineering, Xi'an University of Posts and Telecommunications and with the National Engineering Laboratory for Wireless Security, Xi'an, China. He is also a visiting researcher of the State Key Laboratory of Integrated Services Networks and State Key Laboratory of Information Security. His research interests include wireless network security, privacy preservation, and M2M communications security.

Rongxing Lu received his PhD degree in computer science from Shanghai Jiao Tong University, Shanghai, China in 2006 and a PhD degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2012. He is currently an Assistant Professor with the Division of Communication Engineering, School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore. His research interests include wireless network security, applied cryptography, and trusted computing.

Hui Li received his B.Sc. degree from Fudan University in 1990, an M.A.Sc. and PhD degrees from Xidian University in 1993 and 1998. He is a Professor with the School of Telecommunications Engineering, Xidian University, Xi'an, China. In 2009, he was with the Department of ECE, University of Waterloo as a visiting scholar. His research interests are in the areas of cryptography, security of cloud computing, wireless network security, information theory and network coding. He is the co-author of two books. He served as TPC co-chair of ISPEC 2009 and IAS 2009, general co-chair of e-forensic 2010, ProvSec 2011 and ISC 2011.

Dong Zheng received an M.S. degree in mathematics from Shaanxi Normal University, Xi'an, China, in 1988, and a Ph.D. degree in communication engineering from Xidian University, Xi'an, in 1999. He was a Postdoctoral Fellow in the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, from 1999 to 2001 and a Research Fellow at Hong Kong University, Hong Kong, in 2002. He was a Professor in the School of Information Security Engineering, Shanghai Jiao Tong University. He is also with the State Key Laboratory of Integrated Service Networks, Xidian University. He is currently a Professor in the School of Telecommunication and Information Engineering, Xi'an University of Posts and Telecommunications and is also connected with the National Engineering Laboratory for Wireless Security, Xi'an, China. His research interests include provable security and new cryptographic technology.

Xuemin (Sherman) Shen received his B.Sc. degree from Dalian Maritime University, China, in 1982, and his M.Sc. and PhD. degrees from Rutgers University, New Jersey, in 1987 and 1990, all in electrical engineering. He is a Professor and University Research Chair in the Department of Electrical and Computer Engineering, University of Waterloo. His research focuses on resource management in interconnected wireless/wired networks, UWB wireless communications networks, wireless network security, wireless body area networks, and vehicular ad hoc and sensor networks. He is a co-author of three books and has published more than 400 papers and book chapters in wireless communications and networks, control, and filtering. He is Editor-in-Chief of IEEE Network and will serve as a Technical Program Committee Co-Chair for IEEE INFOCOM 2014. He is the Chair of the IEEE ComSoc Technical Committee on Wireless Communications, and P2P Communications and Networking, and a voting member of GITC. He was a

Founding Area Editor for IEEE Transactions on Wireless Communications, and a Guest Editor for IEEE JSAC, IEEE Wireless Communications, and IEEE Communications Magazine. He also served as the Technical Program Committee Chair for GLOBECOM'07, Tutorial Chair for ICC'08, and Symposia Chair for ICC'10. He received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004, 2007, and 2010 from the University of Waterloo, and the Premier's Research Excellence Award in 2003 from the Province of Ontario, Canada. He is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, a Fellow of the Engineering Institute of Canada, a Fellow of Canadian Academy of Engineering, and was a ComSoc Distinguished Lecturer.