Computer Standards & Interfaces 44 (2016) 85–93

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

A security gateway application for End-to-End M2M communications

Hsing-Chung Chen [a,b,*], Ilsun You [c], Chien-Erh Weng [d], Chia-Hsin Cheng [e], Yung-Fa Huang [f,*]

[a] Dept. of Computer Science and Information Engineering, Asia University, Taichung 41354, Taiwan
[b] Research Consultant with Dept. of Medical Research, China Medical University Hospital, China Medical University Taichung, 40402, Taiwan
[c] Soon Chun Hyang University, South Korea
[d] National Kaohsiung Marine University, Taiwan
[e] National Formosa University, Taiwan
[f] Chaoyang University of Technology, Taiwan

ARTICLE INFO

Article history:
Received 1 February 2015
Received in revised form 28 July 2015
Accepted 2 September 2015
Available online 12 September 2015

Keywords:
Security gateway application
IoTs
M2M
End-to-End
Mutual authentication

ABSTRACT

M2M (Machine-to-Machine) communication for the Internet of Things (IoTs) system is considered to be one of the major issues in future networks. Considering the characteristics of M2M networks in IoTs systems, traditional security solutions are not able to be applied to E2E (End-to-End) M2M networks because the M2M network itself is vulnerable to various attacks. We consider security aspects for M2M communications and then propose a security gateway application (SGA) including the lightweight symmetric key cryptographic negotiation function, secure E2E M2M key exchange generation function and secure E2E M2M messages delivery function. The proposal of the SGA is newly suggested to improve the gateway application (GA) of the ITU-T M2M service layer in the IoTs reference model. We prove that it could prevent various attacks via the theoretical security analyses. Therefore, it could meet the basic security requirements of the M2M service layer.

1. Introduction

In recent years, with the development of computer science, communication technology and perception recognition technology, the network of things has made a great breakthrough. The Internet of Things (IoTs) [1,2] has immense potential to change many of our daily activities, routines and behaviors. The IoTs could find applications in many fields, from the earliest wireless sensor networks such as the military reconnaissance to the present intelligent transportation, smart grid, smart healthcare, smart agriculture, smart logistics and so on [3]. IoTs [4–7] are also defined as a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies. Through the exploitation of identification, data capture, processing and communication capabilities, the IoTs make full use of things to offer services to all kinds of applications, while ensuring that security and privacy requirements are fulfilled [4–7]. The pervasive nature of the information sources means that a great amount of data pertaining to possibly every aspect of human activity, both public and private, will be produced, transmitted, collected, stored and processed. Consequently, integrity and confidentiality of transmitted data as well as the authentication of (and trust in) the services offering that data is crucial. Hence, security is a critical functionality for the IoTs [2]. Data networks, especially wireless, are prone to a large number of attacks such as eavesdropping, spoofing, denial of service and so on. Legacy Internet systems mitigate these attacks by relying on link layer, network layer, transport layer or application layer encryption and authentication of the underlying data. Though some of these solutions are applicable to the IoTs domain, the inherently limited processing and communication capabilities of IoTs devices prevent the use of full-fledged security suites [2].

In an ubiquitous environment, more and more devices are deployed in our daily life, and they need to communicate with one another. M2M (Machine-to-Machine) communication is considered to be one of the major issues in future networks. M2M is expected to bring various benefits in wireless communications when it is interconnected with the Internet [8]. Considering the characteristics of M2M networks, traditional security solutions are not suitable when being applied to M2M service networks because the M2M server network itself is vulnerable to various attacks [8]. A machine could communicate with another machine directly in wireless manners. The M2M communication has attracted a lot of people and industries for its ability to increase efficiency and improve productivity while reducing operating costs. It has great application areas and it could be connected with other infrastructure and brings much more powerful and efficient results. M2M devices or smart devices will ultimately connect to core services networks through a variety of means, from direct broadband or capillary wireless networks, to wired networks [8]. From the ITU-T perspective, the M2M technologies are a key enabler of the

* Corresponding authors.
E-mail addresses: shin8409@ms6.hinet.net, cdma2000@asia.edu.tw (H.-C. Chen), ilsunu@gmail.com (I. You), ceweng@mail.nkmu.edu.tw (C.-E. Weng), chcheng@nfu.edu.tw (C.-H. Cheng), yfahuang@cyut.edu.tw (Y.-F. Huang).

http://dx.doi.org/10.1016/j.csi.2015.09.001
0920-5489/

IoTs [4–7]. The M2M service layer in the ITU-T scope of the ITU-T M2M service layer includes a set of generic and specific functions for the support of a variety of applications enabled by the M2M technologies [4–7]. These functions include management functions and security functions as well as service support and application support functions. The capabilities of the ITU-T M2M service layer are a subset of the whole set of capabilities of the IoTs [4–7].

Moreover, in 2014, the number of mobile subscribers who use smart mobile devices has now surpassed more than 1.2 billion people in the world [9]. Smart devices especially in the third and fourth generations of cellular systems are able to connect fast to the Internet, and the subscribers are easily capable of sending and receiving M2M messages via the smart devices. Therefore, the M2M plays a very important role in the IoTs communications. Despite the critical role of IoTs in the typical Internet users' life, M2M is not so secure. The processing capabilities of smart devices are increasingly enhanced but it cannot compete with the processing capabilities of personal computers. With the increasingly growing reliance on E2E and M2M for the IoTs system in one hand, and the growing number of vulnerabilities and attacks on the other hand, there is an increasingly demand for security solutions [10–12]. There are also some additional security problems in the M2M communication that are not the case in the IoTs system. Therefore, special secure protocols are required for variety E2E (End-to-End) and M2M for the IoTs platforms.

In addition, data confidentiality, authentication, integrity, non-repudiation, access control, and availability are the most important security services in the security criteria that should be taken into account in secure applications and systems [13,10–12,14–17]. However, there is no provision for such security services in the E2E M2M for IoTs system. Both smart device and M2M gateway applications servers are vulnerable to both passive and active attacks. Passive threats include release of message contents, and traffic analysis while active threats include modification of message contents, masquerade, replay, and denial of Service (DoS) [10–12,18–20]. Actually, all the mentioned threats are applicable to the E2E M2M communications [21,22].

The requirement of secure E2E M2M communication enables smart devices to securely communicate in the relationship of M2M. Despite the many solutions [13,10–12,14] that are available now that provide the E2E secure communications, most of them are using the classical ciphers, traditional symmetric cryptosystems and public key cryptography which are dealing with processing the secure communications among a variety of personal computers and server platforms, the solutions designed for M2M could not be suitable for E2E and secure M2M communication. However, several implementations [13,10–12] provide these services, but none of them offers real simple utility security and preserve the privacy of the end-users. However, it will permit traditional security solutions to become more vulnerable, because it is easily suffers from key guessing attacks [16,17]. In the other words, the new SGA approach for E2E M2M service proposed in this paper will realize new applications that are more suitable for secure E2E M2M communications for use with IoTs systems.

Our contribution is that we have newly suggested the basic definitions of a security gateway application (SGA) to improve the gateway application (GA) of the ITU-T M2M service layer in the IoTs reference model [4–7]. The SGA is proposed and defined as a security gateway application for secure E2E M2M communication consisting of the Lightweight Symmetric Key Cryptographic Negotiation Function, Secure E2E M2M Key Exchange Generation Function and Secure E2E M2M Messages Delivery Function for the IoTs system. In addition to that, a lightweight cryptographic symmetric key algorithm will be negotiated by both smart devices for each E2E M2M communication session, which is an efficient choice in the energy and flexibility for the energy-limited smart device. However, the proposed lightweight cryptographic exchange key algorithm could provide the mutual authentication mechanism and prevent the key guessing attack, the undetectable on-line key guessing attack, the data privacy attack as well as the relay attack. The proposed SGA in this paper meets the basic security requirements of the M2M service layer [4–7]. Our proposal includes following:

- Secure Exchange Key establishment between a pair of smart devices and SGA server for E2E M2M communication in the IoTs;
- Lightweight Symmetric Key Cryptographic Negotiation Function;
- The basic definitions of the SGA is first proposed to improve the GA of the ITU-T M2M service layer in the IoTs reference model [4–7];
- The proposed SGA in this paper meets the basic security requirements of the M2M service layer.

The remainder of the paper is structured as follows: the next section addresses previous work on the topic. Section 3 describes the system's overall architecture and presence. Section 4 introduces the designed SGA approach for E2E M2M service. Section 5 evaluates the proposed schemes in terms of security and provides theoretical analyses. Section 5.3 describes why the proposed SGA in this paper is suitable to be a new standard interface. The last session concludes the paper and gives pointers to future work.

2. Related works

In this section, the introduction of the ITU-T Machine to Machine (M2M) service layer and requirements of the ITU-T M2M security service layer in secure E2E M2M communication are described briefly below.

The M2M service layer and their relationship with the IoTs reference model are described as follows: from the ITU-T perspective, the M2M technologies are a key enabler of the IoTs [4–7]. The M2M service layer in the ITU-T scope, the "ITU-T M2M service layer," includes a set of generic and specific functions for the support of a variety of applications enabled by the M2M technologies. These functions include management functions and security functions as well as service support and application support functions. The capabilities of the ITU-T M2M service layer are a subset of the entire set of capabilities of the IoTs. Fig. 1 shows the ITU-T M2M service layer and its position in the IoTs reference model [4–7]. The layered architectural approach, as illustrated in Fig. 1, reduces the implementation complexity while providing interoperability between different applications enabled by the M2M technologies. The specific support capabilities in the service support and application support layer include application specific support capabilities (e.g., the e-health support and telematics support capabilities as shown in Fig. 1). Three types of applications are identified on top of the ITU-T M2M service layer (Application layer): device applications (DAs), gateway applications (GAs) and network application servers (NAs). DA, GA and NA reside, respectively, in a device, gateway and network application server. All these applications can use capabilities provided by the ITU-T M2M service layer [4–7].

The ETSI M2M SCL, the ITU-T M2M service layer as shown in Fig. 1 [4–7] includes specific support capabilities in the service support and application support layer, specific management capabilities and specific security capabilities which are compared in Ref. [4–7]. The interfaces are anticipated and are required to extend including the support of the specific support capabilities in the service support and application support layer, the specific management capabilities and the specific security capabilities [4–7].

However, the ITU-T M2M service layer and ETSI M2M service layer in the IoTs reference model mentioned above did not address the mutual authentication mechanism for the smart device in secure E2E M2M communication. Thus, the mutual authentication mechanism is suggested into the basic requirements of the secure M2M service layer in this paper for smart devices. The basic requirements of the secure M2M service layer are listed below.

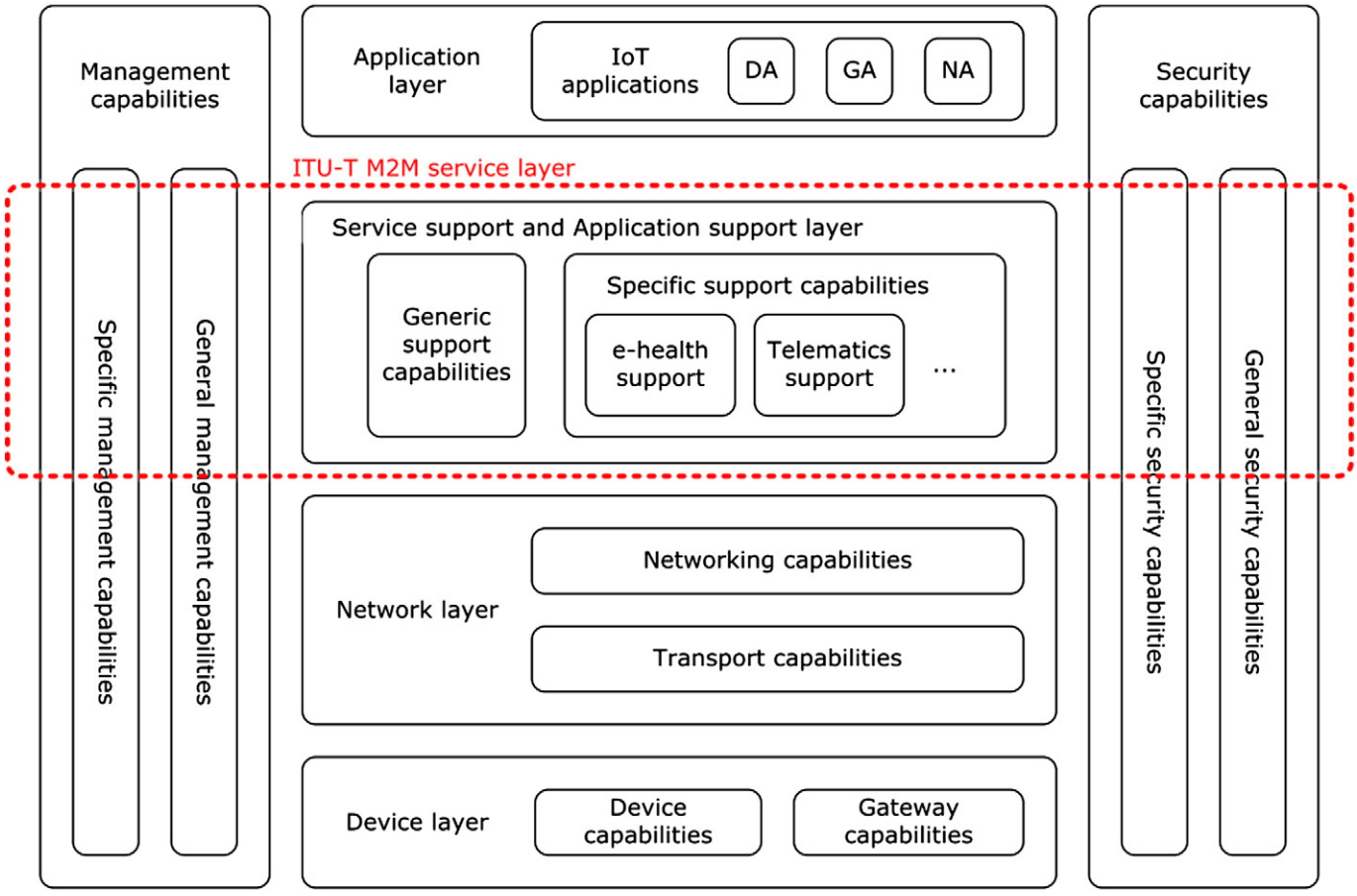

Fig. 1. The ITU-T M2M service layer in the IoTs reference model [19].

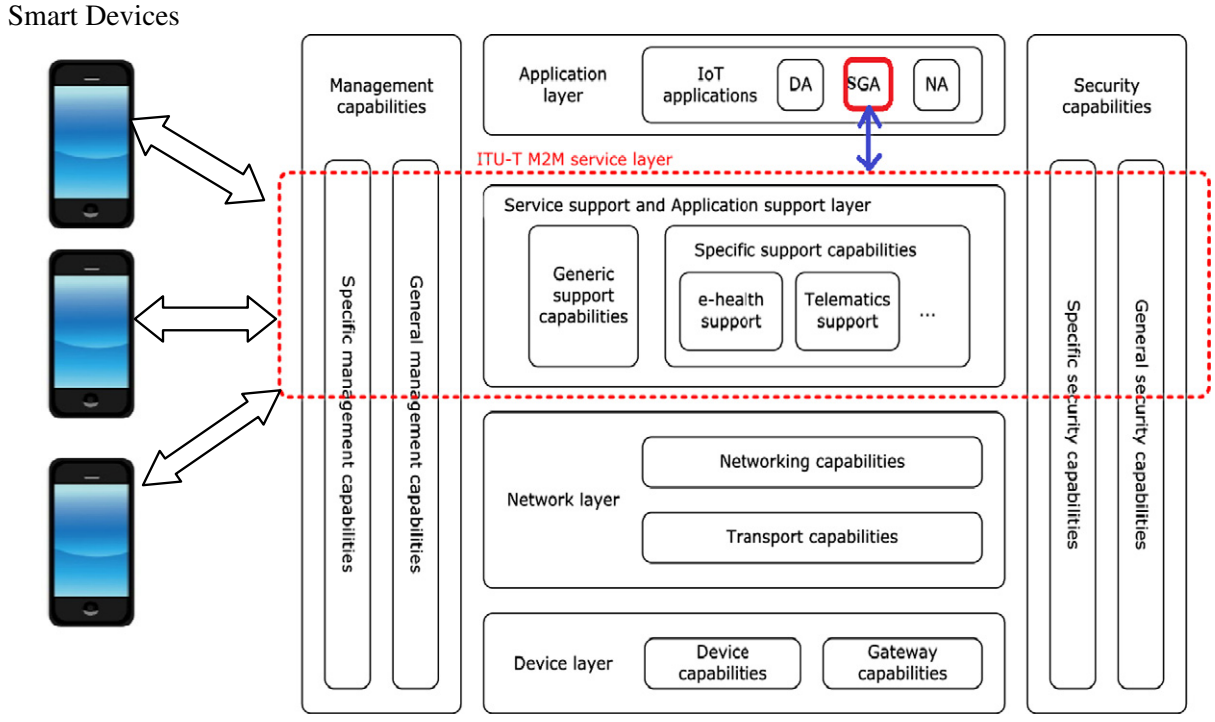

Fig. 2. The architecture of a secure E2E M2M SGA.

- Mutual Authentication: The M2M service layer is required to provide authentication mechanism for applications, smart devices and prevent unauthorized use of the smart devices in secure E2E M2M communication.
- Privacy: The M2M service layer is required to support privacy protection capabilities, such as temporary session key, anonymity of identity and location, according to regulation and laws in secure E2E M2M communication.
- Confidentiality: The M2M service layer is required to support data transfer confidentiality in secure E2E M2M communication.
- Integrity: The M2M service layer is required to support data integrity protection in secure E2E M2M communication.

3. Design of the SGA

Due to the definitions of the ITU-T M2M service layer and ETSI M2M service layer, they did not address the SGA. In this paper, the basic definitions of the SGA are defined as SGA functions for secure E2E M2M communication consisting of the lightweight Symmetric Key Cryptographic Function Negotiation, Secure E2E M2M Key Exchange Generation and Secure E2E M2M Messages Delivery and secure protocols which smart devices could use when building SGA functions for a specific E2E M2M communication system for an IoTs system. SGAs are E2E-based, and they also enable the performance, scalability and security needed for high scale M2M deployments. At a high level, SGAs are ideal for the integration of an E2E M2M enabled infrastructure, and indeed many readily available SGAs could be used for this purpose. The Architecture of Secure E2E M2M SGA as shown in Fig. 2.

4. Design SGA approach for E2E M2M service

It is assumed that there are m smart devices (SDs) in the IoTs system. First, each smart device needs to then initiate the registration phase which is used to create a new account consisting of a distinct Identification ID_{d_i}, e.g. a Radio Frequency Identification (RFID), a set of lightweight symmetric key cryptographic functions $\mathbb{C}^G = \{(E^{G,x}(\cdot), D^{G,x}(\cdot)), x = 1, 2, 3, \ldots\}$ plus their identification set $\mathbb{C}idSet^G = \{(E^{G,x}, D^{G,x}), x = 1, 2, 3, \ldots\}$ and an initial key k_{d_G} shared and kept secure by the SGA Server and itself for getting the secure M2M services in the IoTs System. Second, if the smart device wants to securely communicate to a group of machines or smart devices which consist of some invited members for a specific purpose, e.g. secure broadcasting and multicasting messages, the smart device needs to launch a group request for the M2M construct phase to the SGA Server in the IoTs System for organizing a special secure group.

The system's overall architecture including smart device registration for the SGA and Secure E2E M2M for SGA in the IoTs system is presented below.

4.1. Smart device registration for SGA in IoTs system

This subsection will illustrate the scheme of the registration phase as shown in Fig. 3. The Cryptographic Accelerator (CA for short) module connects with the SGA server and selects and then generates a series of parameters for the exchange of public information while holding a secure M2M communication in the IoTs system as per the procedure below. The secret key PR is only kept secure by the SGA server, a large prime number p and a generator, $g \in F_p^*$, are pre-chosen. The public key PK is computed, such that $D_{PR}(E_{PK}(M)) = M$. The CA of the SGA server provides a public key encryption algorithm $E^G(\cdot)$, decryption algorithm $D^G(\cdot)$ and public key PK together with the secret key PR. The CA of the SGA server also provides a set of lightweight symmetric key cryptographic functions $\mathbb{C}^G = \{(E^{G,x}(\cdot), D^{G,x}(\cdot)), x = 1, 2, 3, \ldots\}$ and their relevant cryptographic identification set $\mathbb{C}idSet^G = \{(E^{G,x}, D^{G,x}), x = 1, 2, 3, \ldots\}$ for the security gateway of the smart device. For the key exchange phase, the CA of the SGA server prepares a trapdoor one-way hash function $F_G(\cdot)$ and a one-way hash function $f_k(\cdot)$. The important parameters and relative functions mentioned above will be packed into a download package.

After downloading the package from the SGA server, the smart device d_i chooses an identity ID_{d_i}, secret initial or registered key k_{d_i} and the lightweight symmetric key cryptographic function set $\mathbb{C}idSet^i$. In this registration phase, the smart device d_i will send the registration request messages $E_{PK}^G(k_{d_i})\|E^{i,x}\|(ID_{d_i})(\mathbb{C}idset^i)$ to the CA of the SGA server, where the initial cryptographic function $E^{i,x}(\cdot) \in \mathbb{C}^i$ is arbitrary choice by the smart device d_i. Once receiving the request messages from the smart device d_i, the CA of the SGA server will decrypt and verify the initial and registered secret key $D_{PR}^G(k_{d_i}) = k_{d_i}$, the device identity ID_{d_i}, the set of lightweight symmetric key cryptographic functions $\mathbb{C}^G = \{(E^{G,x}(\cdot), D^{G,x}(\cdot)), x = 1, 2, 3 \ldots\}$ plus their relevant cryptographic identifications in the set

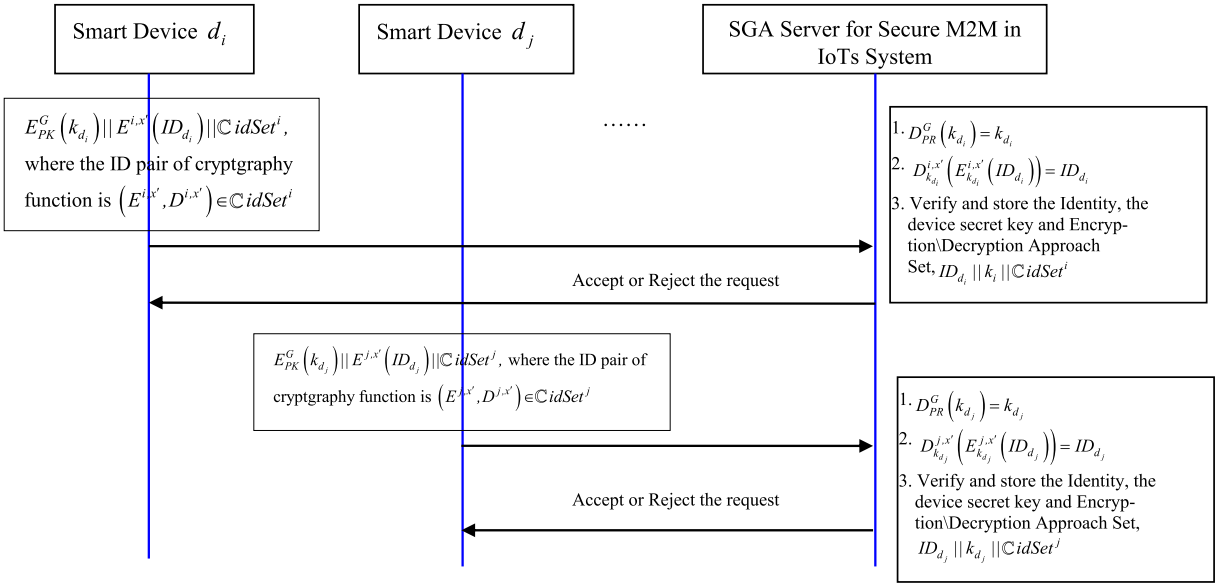

Fig. 3. Registration phase.

$\mathbb{C}idSet^i$ whether they are available for the IoTs system or not. If they are available, then the device identity ID_{u_i}, corresponding cryptographic sets and secret key k_{d_i} will be secure kept and stored by the SGA server and the acceptance acknowledgment will be sent to the requestor d_i.

For example, the smart devices d_i and d_j initial the registration processes in Fig. 3. Finally, CA of SGA server will keep their identities, secret keys together with the set of lightweight symmetric key cryptographic functions as well as their relevant cryptographic identification sets: $\{ID_{d_i}, k_{d_i}, \mathbb{C}^i, \mathbb{C}idSet^i\}$ and $\{ID_{d_j}, k_{d_j}, \mathbb{C}^j, \mathbb{C}idSet^j\}$ for smart devices d_i and d_j, individually.

4.2. Secure E2E M2M for SGA in the IoTs system

This subsection will illustrate the secure E2E M2M for SGA in the IoTs system. There are three phases consisting of the Symmetric Key Cryptographic Function Negotiation Phase, Secure E2E M2M Key Exchange Generation Phase and Secure E2E M2M Messages Delivery Phase in this SGA. In the Symmetric Key Cryptographic Function Negotiation Phase as shown in Fig. 4, two smart devices negotiate the lightweight symmetric key cryptographic function and generate the temporary key via their GA modules and SGA server. In the Secure E2E M2M Key Exchange Phase as shown in Fig. 5, two smart devices cooperatively generate the exchange session key via their GA modules and SGA server. The secure M2M E2E communication will be held by the processes shown in the Secure E2E M2M Messages Delivery Phase in the IoTs System.

4.3. Symmetric key cryptographic function negotiation phase

Step A1: Secure E2E M2M symmetric key cryptographic function invitation request (SE2E M2M SKCF invitation request): If a smart device d_i wants to launch a secure M2M communication with smart device d_j. At first, the smart device d_i will choose a symmetric key cryptographic function $E^{i,x}(\cdot) \in \mathbb{C}^i$ together with a temporary session key k_t^i via his GA module, where the relevant identity of the symmetric key cryptographic function is $E^{i,x} \in \mathbb{C}idSet^i$. The smart device d_i will then perform $E_{PK}^G((k_t^i \oplus k_{d_i}) || ID_{d_i} || ID_G || ID_{d_j} || (E^{i,x}, D^{i,x}))$ by using the public key and cryptographic function of the SGA server. Moreover, the smart device d_i sends these encrypted messages $E_{PK}^G((k_t^i \oplus k_{d_i}) || ID_{d_i} || ID_G || ID_{d_j} || (E^{i,x}, D^{i,xs}))$ to the SGA server.

Because of all the messages are delivered from the smart

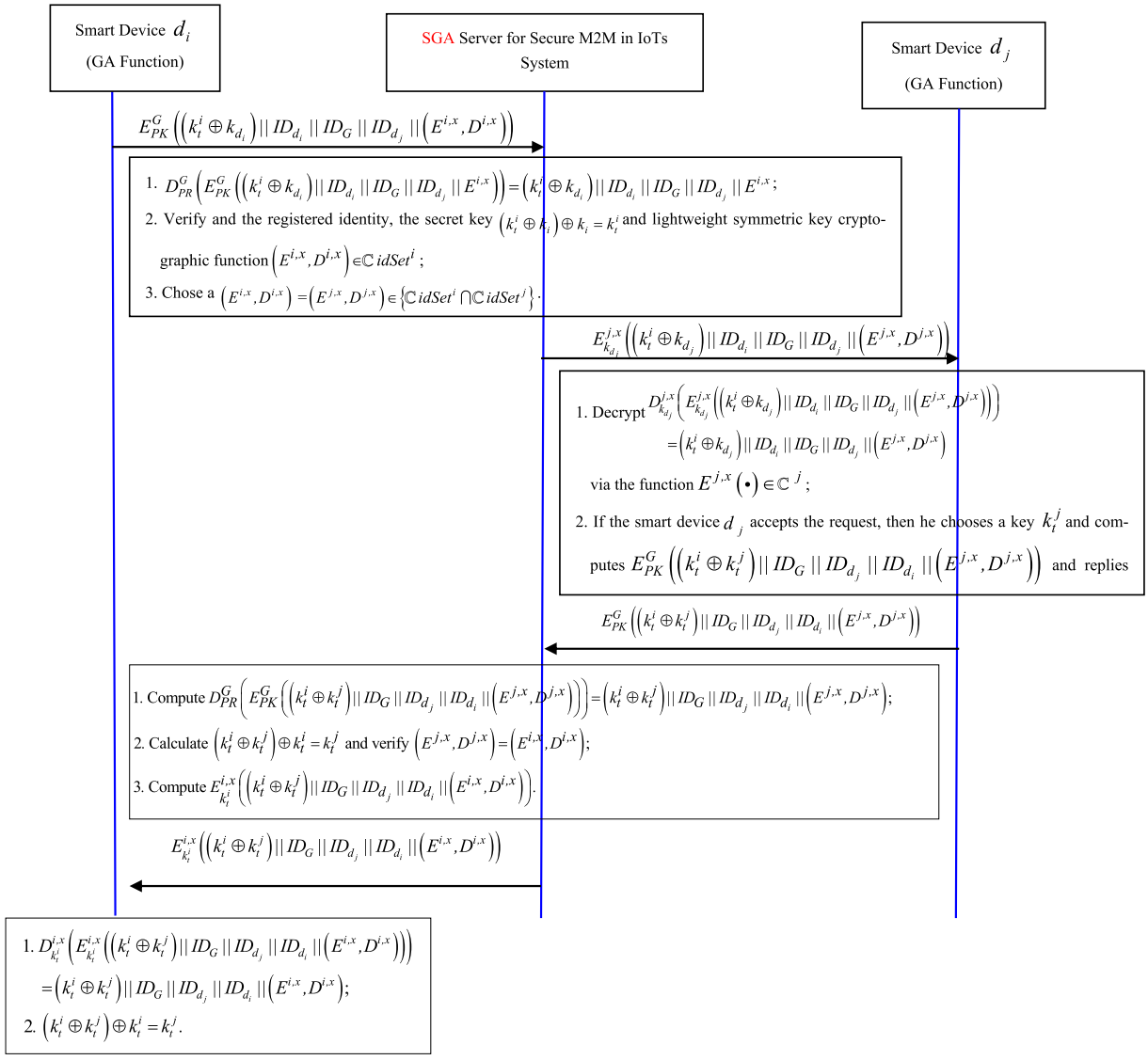

Fig. 4. Secure E2E M2M for SGA in IoTs system.

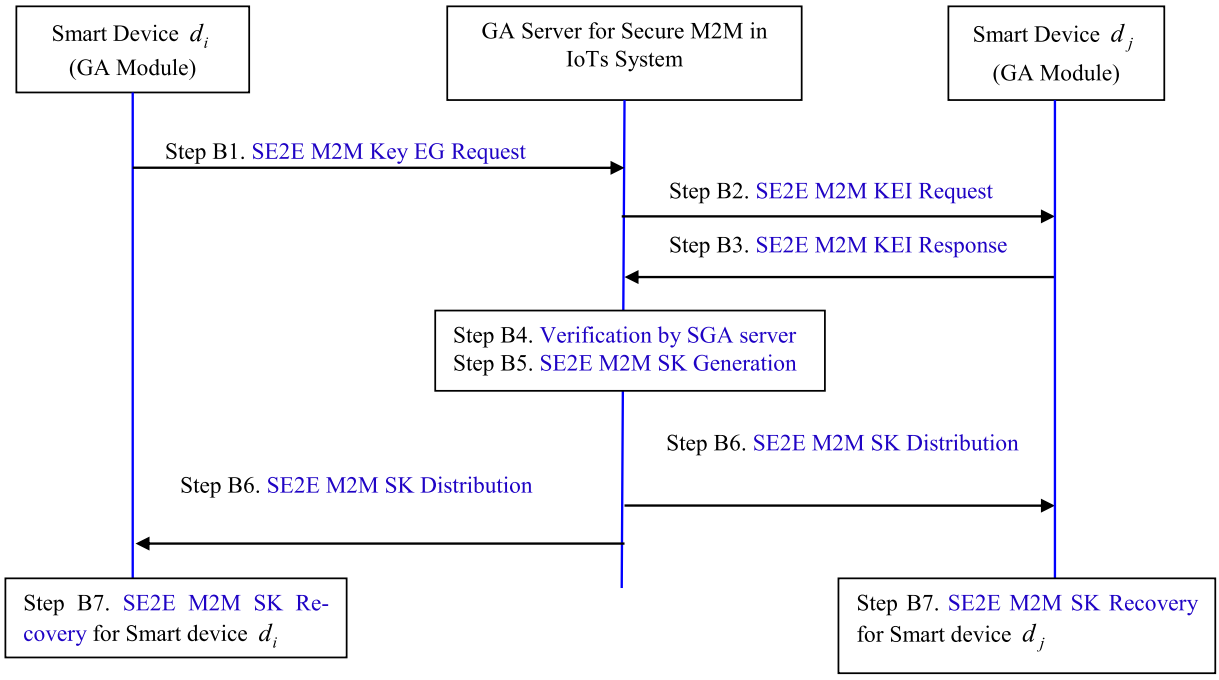

Fig. 5. Secure E2E M2M SGA key exchange generation phase.

devices in the IoTs system, they will be stored in the database of the SGA server.

Step A2: Secure E2E M2M cryptographic function negotiation request (SE2E M2M CF negotiation request): After receiving the encrypted messages sent by the smart device d_i, the server SGA performs the decryption $D^G_{PR}(E^G_{Pk}((k^i_t \oplus k_{d_i}) || ID_{d_i} || ID_G || ID_{d_j} || (E^{i,x}, D^{i,x}))) = (k^i_t \oplus k_{d_i}) || ID_{d_i} || ID_G || ID_{d_j} || (E^{i,x}, D^{i,x})$, and verifies his registered identity, the secret key $(k^i_t \oplus k_{d_i}) \oplus k_{d_i} = k^i_t$ and the identity of the lightweight symmetric key cryptographic function $(E^{i,x}, D^{i,x}) \in \mathbb{C}idSet^i$. The server SGA will also verify the lightweight symmetric key cryptographic functions whether they are satisfying the equation $(E^{i,x}(\cdot), D^{i,x}(\cdot)) = (E^{i,x}(\cdot), D^{i,x}(\cdot)) \in \{\mathbb{C}^i \cap \mathbb{C}^j\}$ or not. If the condition $(E^{i,x}, D^{i,x}) \neq (E^{j,x}, D^{j,x})$ is satisfied, it implies that $(E^{i,x}, D^{i,x})$or$(E^{j,x}, D^{j,x}) \notin \{\mathbb{C}idSet^i \cap \mathbb{C}idSet^j\}$. Thus, the SGA server will reject the request and send a response to the smart device d_i. On the contrary, the SGA server calculates the SE2E M2M CF negotiation request messages $E^{j,x'}_{k_{d_j}}((k^i_t \oplus k_{d_j}) || ID_{d_i} || ID_G || ID_{d_j} || (E^{j,x}, D^{j,x}))$ via the initial and registered encryption function $E^{j,x'}_{k_{d_j}}(\bullet)$ of the target smart device d_j and forwards them to it.

Step A3: Secure E2E M2M cryptographic function negotiation response (SE2E M2M CF negotiation response): Similarly, when the smart device d_j receives the SE2E M2M CF negotiation request messages from the SGA server, it decrypts $D^{j,x'}_{k_{d_j}}(E^{j,x'}_{k_{d_j}}((k^i_t \oplus k_{d_j} || ID_{d_i} || ID_G || ID_{d_j} || E^{j,x}, D^{j,x}))) = (k^i_t \oplus k_{d_j}) || ID_{d_i} || ID_G || ID_{d_j} || (E^{j,x}, D^{j,x})$. In addition, if the smart device d_j accepts the request, then he will find the negotiated symmetric key cryptographic $E^{j,x}(\bullet) \in \mathbb{C}^j$ with a temporary key k^j_t via his GA module, where the relevant identity the symmetric key cryptographic is $E^{j,x} \in \mathbb{C}idSet^j$, and encrypts the response messages $E^G_{PK}((k^i_t \oplus k^j_t) || ID_G || ID_{d_j} || ID_{d_i} || (E^{j,x}, D^{j,x}))$ by using the public key and cryptographic function of SGA server and replies back to them via the SGA server.

Step A4: Negotiation and coordination processes by the SGA server (Negotiation and coordination by SGA): After receiving the encrypted messages $E^G_{PK}((k^i_t \oplus k^j_t) || ID_G || ID_{d_j} || ID_{d_i} || (E^{j,x}, D^{j,x}))$ sent by the smart device d_j, the CA of the SGA will then decrypt by using the private key, retrieve k^j_t by the equations $(k^i_t \oplus k^j_t) \oplus k^i_t = k^j_t$, verify $(E^{j,x}, D^{j,x}) = (E^{i,x}, D^{i,x})$ and compute $E^{i,x}_{k^j_t}((k^i_t \oplus k^j_t) || ID_G || ID_{d_j} || ID_{d_i} || (E^{j,x}, D^{j,x}))$ by using the symmetric key cryptographic function $E^{i,x}(\cdot) \in \mathbb{C}^i$ together with the temporary session key k^i_t chosen by the smart device d_i in Step A1, respectively. Furthermore, the SGA server will respond to these encrypted messages $E^{i,x}_{k^i_t}((k^i_t \oplus k^j_t) || ID_G || ID_{d_j} || ID_{d_i} || (E^{i,x}, D^{i,x}))$ to the smart device d_i.

Step A5: After receiving the response from SGA server, the smart device d_i will decrypt these messages $D^{i,x}_{k^i_t}(E^{i,x}_{k^i_t}((k^i_t \oplus k^j_t) || ID_G || ID_{d_j} || ID_{d_i} || (E^{i,x}, D^{i,x}))) = (k^i_t \oplus k^j_t) || ID_G || ID_{d_j} || ID_{d_i} || (E^{i,x}, D^{i,x})$ by using the symmetric key cryptographic function $E^{i,x}(\cdot) \in \mathbb{C}^i$ together with the temporary session key k^i_t choice by the smart device d_i in Step A1. Next, smart device d_i derives k^j_t by the equation $(k^i_t \oplus k^j_t) \oplus k^i_t = k^j_t$.

4.4. Secure E2E M2M SGA key exchange generation phase

Step B1: Secure E2E M2M Key Exchange Generation request (SE2E M2M Key EG Request): If a smart device d_i wants to launch an E2E M2M Key Exchange Generation request for a secure M2M communication with smart device d_j during a session s, then the smart device d_i needs to generate two random numbers R_i and r_i, and performs $N_i = g^{R_i} (mod\ p)$ and $K_{is} = N_i{}^{r_i} (\ mod p)$ for the session s. Moreover, he performs $E_{k_{d_i}}(N_i \oplus k_{d_i})$, $F_G(r_i)$, and $f_{K_{is}}(N_i)$. Finally, smart device d_i sends the messages $E^{i,x}_{k^i_t}(N_i, ||K_{is}||ID_{d_i}||ID_{d_j}||ID_G||E_{k_{d_i}}(N_i \oplus k_{d_i}) || F_G(r_i) || f_{K_{is}}(N_i))$

encrypted by using the temporary session key k_t^i and a choice cryptographic function pair ($E^{i,x}$, $D^{i,x}$), which are obtained by the *Symmetric Key Cryptographic Function Negotiation Phase*, to the server.

Step B2: Secure E2E M2M Key Exchange Invitation Request (SE2E M2M KEI Request): After receiving the encrypted messages sent by the smart device d_i, the SGA server decrypts the messages $D_{k_t^i}^{i,x}(E_{k_t^i}^{i,x}(N_i||\ K_{is}||ID_{d_i}||ID_{d_j}||ID_G||E_{k_{d_i}}(N_i \oplus k_{d_i})||F_s(r_i)||f_{k_{is}}(N_i))) = N_i||K_{is}||ID_{d_i}||ID_{d_j}||ID_G||E_{k_{d_i}}(N_i \oplus k_{d_i})||F_s(r_i)||f_{K_{is}}(N_i)$, by using the temporary session key k_t^i and a chosen cryptographic function pair $E^{i,x}$, $D^{i,x}$ of the smart device d_i kept by the SGA server. In addition, the SGA server sends the SE2E M2M KEI request to the smart device d_j.

Step B3: Secure E2E M2M Key Exchange Invitation Response (SE2E M2M KEI Response): After receiving the request from the SGA server, the smart device d_j also generates two random numbers R_j and r_j, and performs $N_j = g^{R_j}(mod\ p)$ and $K_{js} = N_j^{r_j}(\mathrm{mod} p)$. Moreover, it performs $E_{k_{d_j}}(N_j \oplus k_{d_j})$, $F_G(r_j)$, and $f_{K_{js}} = (N_j)$. Finally, it sends these messages the messages $E_{K_t^j}^{j,x}(N_j||K_{js}||ID_{d_i}||ID_{d_j}||ID_G||E_{k_{d_j}}(N_j \oplus k_{d_j})||F_s(r_j)||f_{K_{js}}(N_j))$ encrypted by using the temporary session key k_t^j and a choice cryptographic function pair $(E^{j,x}, D^{j,x}) = (E^{i,x}, D^{i,x})$, which are obtained by the *Symmetric Key Cryptographic Function Negotiation Phase*, to the SGA server.

Step B4: Verification by SGA server: After collecting the messages sent by the smart device d_j, the CA of the SGA server will also decrypt

$$D_{k_t^j}^{j,x}\left(E_{k_t^j}^{j,x}\left(N_j||K_{js}||ID_{d_i}||ID_{d_j}||ID_G||E_{k_{d_j}}\left(N_j \oplus k_{d_j}\right)||F_G(r_j)||f_{K_{js}}(N_j)\right)\right) = N_j||K_{js}||ID_{d_i}||ID_{d_j}||ID_G||E_{k_{d_j}}\left(N_j \oplus k_{d_j}\right)||F_G(r_j)||f_{K_{js}}(N_j).$$

Next, the CA of the SGA server decrypts $D_{k_{d_j}}(N_j \oplus k_{d_j}) = N_j \oplus k_{d_j}$ by using the corresponding key k_{d_j}. Similarly, the $D_{k_{d_j}}(N_j \oplus k_{d_j}) = N_j \oplus k_{d_j}$ is decrypted by using the corresponding key k_{d_j}. In addition, the CA of the SGA server retrieves N_i and N_j by the equations $(N_i \oplus k_{d_j}) \oplus k_{d_j} = k_{d_j} = N_j$ and $(N_i \oplus k_{d_i} = N_i)$ using the corresponding keys k_{d_j} and k_{d_j}, respectively. Moreover, the CA of the SGA server derives r_i and r_j from the trapdoor one-way hash functions $F_G(r_i)$ and $F_G(r_j)$. Then, the CA of the SGA server performs $K_{is} = N_i^{r_i}(mod\ p)$ and $K_{js} = N_j^{r_j}(mod\ p)$ from the values r_i, r_j, N_i and N_j. Additionally, the CA of the SGA server then verifies that the received hash values are $f_{K_{is}}(N_i)$ and $f_{K_{js}}(N_j)$. If these verifications for authentication between both smart device d_i and d_j are successful, then the CA of the SGA server will generate a random number R_S for this session s. But, if these verifications for authentication are unsuccessful, the CA of the SGA server will reject this request.

Step B5: Secure E2E M2M Session Key Generation (SE2E M2M SK Generation): The CA of the SGA server performs the secure E2E M2M Session key $K_{iSj} = (N_i^{R_s})^{R_j} = ((g^{R_i})^{R_s})^{R_j} = g^{R_i R_s R_j} = ((g^{R_j})^{R_s})^{R_i} = (N_j^{R_s})^{R_i} = K_{iSj}(mod\ p)$ for both smart device d_i and d_j, and calculates the messages $N_i^{R_s}(mod\ p)$ and $N_j^{R_s}(mod\ p)$, $f_{K_{is}}(ID_{d_i}, ID_{d_j}, K_{is}, N_j^{R_s})$ and $f_{K_{js}}(ID_{d_i}, ID_{d_j}, K_{js}, N_i^{R_s})$.

Step B6: Secure E2E M2M Session Key Distribution (SE2E M2M SK Distribution): Finally, the SGA server will send back only the messages $N_j^{R_S}$, $f_{K_{is}}(ID_{d_i}, ID_{d_j}, K_{is}, N_j^{R_s})$ to smart device d_i, and the messages $N_i^{R_S}$ and $f_{K_{js}}(ID_{d_i}, ID_{d_j}, K_{js}, N_i^{R_s})$ to smart device d_j, individually.

Step B7: Secure E2E M2M Session Recovery (SE2E M2M SK Recovery): After receiving both messages $N_j^{R_s}$ and $f_{K_{is}}(ID_{d_i}, ID_{d_j}, K_{is}, N_i^{R_s})$, the smart device d_i could verify $f_{K_{is}}(ID_{d_i}, ID_{d_j}, K_{is}, N_j^{R_s})$ by using $N_j^{R_s}$ and the messages ID_{d_i}, ID_{d_j}, and K_{is}. If the messages are verified as successful, then the smart device d_i will compute the secure E2E M2M Session key by the equation $(N_j^{R_s})^{R_i} = K_{isj}(\mathrm{mod} p)$. Similarly, after receiving the messages $N_i^{R_s}$ and $f_{K_{js}}(ID_{d_i}, ID_{d_j} K_{js}, N_i^{R_s})$, the smart device d_j could verify $f_{K_{js}}(ID_{d_i}, ID_{d_j} K_{js}, N_i^{R_s})$ by using $N_i^{R_s}$ and the messages ID_{d_i}, ID_{d_j}, and K_{js}. If the messages are verified as successful, then the smart device d_j performs the secure E2E M2M key by the equation $(N_i^{R_s})^{R_j} = K_{jsi} = K_{isj}(mod\ p)$.

4.5. Secure E2E M2M messages delivery phase in IoTs system

In the phase, assumed that the smart device d_i will securely send the set of the messages $M = \{m_1, m_2, \ldots, m_i, \ldots\}, x \in Z^* = \{1, 2, 3, \ldots\}$ to the smart device d_j during a session s. First, the smart device d_i computes the ciphertext $c_i = E_{K_{isj}}^{i,x}(m_i), i \in Z^*$ by using the symmetric key cryptographic function $E_{K_{isj}}^{i,x}(\cdot) \in \mathbb{C}^i = E_{K_{jsi}}^{j,x}(\cdot) \in \mathbb{C}^j$ together with the secure E2E M2M exchange session key $K_{isj} = K_{jsi}$ negotiated by *Secure E2E M2M Key Exchange Phase*. Upon receiving the secure messages, the SGA Server will store them to the Storage and forward them to the smart device d_j. After receiving each ciphertext message $c_i = E_{K_{isj}}^{i,x}(m_i), i \in Z^*$, the smart device d_j will decrypt them by the symmetric key cryptographic function $E_{K_{jsi}}^{j,x}(\cdot) \in \mathbb{C}^j = E_{K_{isj}}^{i,x}(\cdot) \in \mathbb{C}^i$ such as $D_{K_{jsi}}^{j,x}(c_i) = D_{K_{jsi}}^{j,x}(E_{K_{isj}}^{i,x}(m_i)) = D_{K_{jsi}}^{i,x}(E_{K_{jsi}}^{i,x}(m_i)) = m_i, i \in Z^*$.

On the contrary, if the message set $M = \{m_1, m_2, \ldots, m_j, \ldots\}, j \in Z^* = \{1, 2, 3, \ldots\}$ will be secure sent one by one to the smart device d_i by the smart device d_j. The smart device d_j compute the ciphertext $c_j = E_{jsi}^{j,x}(m_i), i \in Z^*$ by using the symmetric key cryptographic function $E_{K_{jsi}}^{j,x}(\cdot) \in \mathbb{C}^j = E_{K_{isj}}^{i,x}(\cdot) \in \mathbb{C}^i$ together with the secure E2E M2M exchange session key $K_{jsi} = K_{isj}$ negotiated by *Secure E2E M2M Key Exchange Phase*. Upon receiving the secure messages, the SGA server will store them to message storage of SGA server, and forward them to the smart device d_i. After receiving each ciphertext message $c_j = E_{jsi}^{j,x}(m_i), i \in Z^*$, the smart device d_i will decrypt them by the same the symmetric key cryptographic function $E_{K_{isj}}^{i,x}(\cdot) \in \mathbb{C}^i = E_{K_{jsi}}^{j,x}(\cdot) \in \mathbb{C}^j$, such as $D_{K_{isj}}^{i,x}(c_j) = D_{K_{isj}}^{i,x}(E_{K_{jsi}}^{j,x}(c_j)) = D_{K_{jsi}}^{j,x}(E_{K_{jsi}}^{j,x}(c_j)) = m_j, j \in Z^*$.

5. Security analysis

In this section, the Secure E2E M2M SGA is discussed and analyzed. This could provide a mutual authentication mechanism and prevent the key guessing attack, the undetectable on-line key guessing attack, the data privacy attack, and the relay attack. The proposed SGA in this paper meets the basic security requirements of the M2M service layer [19, 20]. The details of the discussions and analyses are described below.

5.1. Mutual authentication mechanism

The mutual authentication function is proposed in the Secure E2E M2M SGA, which is applied to the smart devices d_i, d_j and CA of SGA server. First, both smart devices d_i and d_j in the proposed scheme are protected by the trapdoor hash function $F_S(\bullet)$ together with the random numbers r_i and r_j. Only CA of SGA server knows this trapdoor information. Therefore, the two generated random number R_i and r_i only CA of

SGA server could be securely retrieved from $F_S(r_i)$ and $F_S(r_j)$ during the communication session s. Moreover, both smart devices d_i and d_j utilize the negotiated symmetric encryption function in order to encrypt and decrypt $N_i \oplus k_{d_i}$ and $N_j \oplus k_{d_j}$, where the two secret keys k_{d_i} and k_{d_j} will be needed to pending some bits in order to fit the same length of N_i or N_j. In the proposed SGA, the secret key of smart device d_i is k_{d_i}, and the key of smart device d_j is k_{d_j}, which are shared securely to CA of SGA server. Thus, CA of SGA server could decrypt N_i and N_j via the both equations $E_{k_{d_i}}(N_i \oplus k_{d_i})$ and $E_{k_{d_j}}(N_j \oplus k_{d_j})$ by using the corresponding secret keys. Furthermore, CA of SGA server then extracts r_i and r_j from the one-way trapdoor hash functions $F_S(r_i)$ and $F_S(r_j)$. And then, CA of SGA server could compute the one-time strong keys $K_{is} = N_i^{r_i} (\mathrm{mod} p)$ and $K_{js} = N_j^{r_j} (\mathrm{mod} p)$ from the values r_i, r_j, N_i and N_j. Accordingly, CA of SGA server could verify the received hash values $f_{K_{is}}(N_i)$ and $f_{K_{js}}(N_j)$. If these mutual authentication procedures between both smart devices d_i and d_j are successful, then both smart devices d_i and d_j are the legitimate users in the IoTs system.

On the contrary, when the mutual authentication procedure are checked and performed by both smart devices d_i and d_j, CA of SGA server could use the pre-computed keys $K_{is} = N_i^{r_i} (\mathrm{mod} p)$ and $K_{js} = N_j^{r_j} (\mathrm{mod} p)$ to compute the hash values $f_{K_{is}}(ID_{d_i}, ID_{d_j}, K_{is}, N_j^{R_S})$ and $f_{K_{js}}(ID_{d_i}, ID_{d_j}, K_{js}, N_i^{R_S})$. Consequently, CA of SGA server sends the messages $N_j^{R_S}$, $f_{K_{is}}(ID_{d_i}, ID_{d_j}, K_{is}, N_j^{R_S})$ to smart device d_i. Similarly, CA of SGA server will be asked to send the messages $N_i^{R_S}$ and $f_{K_{js}}(ID_{d_i}, ID_{d_j}, K_{js}, N_i^{R_S})$ to smart device d_j. When the smart device d_i or d_j receives the messages sent from CA of SGA server, the smart device d_i or d_j then checks $f_{K_{is}}(ID_{d_i}, ID_{d_j}, K_{is}, N_j^{R_S})$ or $f_{K_{js}}(ID_{d_i}, ID_{d_j}, K_{js}, N_i^{R_S})$ in order to verify them again. If the authentication procedure for SGA sever is successful, then the smart device d_i or d_j could trust the SGA sever in order to further perform the secure E2E M2M key as $(N_j^{R_S})^{R_i} = K_{isj} (\mathrm{mod} p) = (N_i^{R_S})^{R_j} = K_{jsi} (\mathrm{mod} p)$.

Finally, through the above discussions, CA of SGA server could verify the both smart devices d_i and d_j are legal or not. Conversely, both smart devices d_i and d_j could also verify the CA of SGA server is fake or not. Therefore, the proposed secure E2E M2M SGA could provide the mutual authentication mechanism in IoTs system.

5.2. Prevention on the key guessing attack

In the subsection, the secure E2E M2M SGA is respectively analyzed how it could prevent the key guessing attack. Generally, an attacker may use the detectable/undetectable on-line guessing attacks to get the key of Smart device. Because of the sensitivity information such as $E_{k_{d_i}}(N_i \oplus k_{d_i})$, $E_{k_{d_j}}(N_j \oplus k_{d_j})$, $F_s(r_i)$, $F_s(r_i)$, $f_{K_{is}}(N_i)$ and $f_{K_{js}}(N_j)$ are protected by using the k_{d_i}, k_{d_j}, N_i, N_j, r_i, r_j and the trapdoor hash function. Therefore, the attacker doesn't know the M2M's key via variety guessing attacks. However, each smart device has her/him own key only shared by the CA of SGA server. If the authentication procedure is fail, the request will be reject. Thus, the proposed schemes could effectually protect the off-line key guessing attack.

5.3. Prevention on the undetectable on-line key guessing attack

For preventing the on-line undetectable key guessing attacks, the operations of both $N_i \oplus k_{d_i}$ and $N_j \oplus k_{d_j}$ are added into the symmetric key exchange algorithms as $E_{k_{d_i}}(N_i \oplus k_{d_i})$, $E_{k_{d_j}}(N_j \oplus k_{d_j})$. However, the keys are shared between the CA of SGA server and the corresponding smart devices. When attackers want to guess the corresponding secret key of both smart devices, they need to pass the authentication procedure. Owing to the corresponding secret keys are only kept by the CA of SGA server and corresponding smart devices, such that the secret key k_{d_i} corresponds to smart device d_i and the key k_{d_j} corresponds to smart device d_j. Therefore, the proposed schemes could resist the undetectable on-line key guessing attack.

5.4. Data privacy attack

In the original standard [4–7,19,15–17], messages transmitted between smart devices and the GA (Gateway Application) are not protected. Adversaries can utilize a malicious smart device to hear any message transmitted between the GA and other smart devices. Our proposed scheme exploits the idea of the SGA in E2E M2M communication. Messages are encrypted by using the negotiated symmetric key cryptographic function $E^{i,x}_{K_{isj}}(\cdot) \in \mathbb{C}^i = E^{j,x}_{K_{jsi}}(\cdot) \in \mathbb{C}^j$ together with the secure E2E M2M exchange session key $K_{isj} = K_{jsi}$ by the *Secure E2E M2M Key Exchange Phase*. Without the symmetric key, adversaries only derive non-meaningful binary data from the cipher. Due to each round of encrypted/decrypted procedures in both the smart devices being protected by two random numbers R_j and r_j for a specific session s, the proposed SGA can easily prevent the data privacy attack.

5.5. Relay attack

The relay attack means that both smart devices and the SGA server in E2E M2M communication controlled by an adversary. The adversary relays a message sent by the smart device to trick the SGA server. In the proposed Secure E2E M2M SGA, messages are encrypted using a negotiated symmetric key cryptographic function and exchange session key produced according to distinct random numbers kept by the CA of the SGA server only for session s. Only if the adversary could derive the symmetric key, the adversary cannot know any message within them. In the proposed Secure End to End M2M SGA, adversaries cannot tamper with messages because they doesn't know the symmetric keys because we use the distinct to distinct random numbers kept by the CA of the SGA server only for each session s. Also, the mutual authentication information mechanism could detect the illegal smart devices that want to obtain confidential information by using a false identity. When both smart devices and SCA server execute the authentication procedures as described above, attackers engaging in relay attacks will be detected by the transponders. Thus, our scheme can prevent the relay attack.

6. Discussion

Due to traditional security solutions are not able to be applied to E2E M2M networks in IoTs systems, the M2M networks itself are vulnerable to various attacks. The proposal functions of the SGA are defined for secure M2M communications including the Lightweight Symmetric Key Cryptographic Negotiation Function, Secure E2E M2M Key Exchange Generation Function and Secure E2E M2M Messages Delivery Function. It is a newly suggestion for improving the gateway application (GA) of the ITU-T M2M service layer in the IoTs reference model [19]. Its lightweight cryptographic symmetric key algorithm could be used to negotiate by both smart devices for each E2E M2M communication session. It could be choice flexibility for the energy-limited smart devices. In addition, the cryptographic exchange key algorithm could provide the mutual authentication mechanism. We prove that it is could prevent the key guessing attack, the undetectable online key guessing attack, the data privacy attack, and the relay attack. Therefore, the proposal functions of the SGA in this paper are suitable to be a new standard interface because their meet the basic security requirements of the M2M service layer mentioned in Section 1.

7. Conclusions

Owing to the increasingly growing reliance on M2M services plays a very important role in the IoTs system, which is accompanied by the growing number of vulnerabilities and attacks on their smart devices and application servers. Thus, there is an increasingly huge demand for security solutions. Therefore, the proposed SGA could provide a mutual authentication mechanism, and prevent the key guessing attack, the undetectable on-line key guessing attack, the data privacy attack as well as the relay attack. The SGA proposed in this paper meets the core security requirements of the M2M service layer [4–7]. Considering the characteristics of the E2E M2M service layer for IoTs systems, traditional security solutions are not proper to be applied to E2E M2M networks. To ensure each smart device is able to securely communicate with each other, the SGA includes the Lightweight Symmetric Key Cryptographic Negotiation Function, Secure E2E M2M Key Exchange Generation Function and Secure E2E M2M Messages Delivery Function. Moreover, the lightweight cryptographic symmetric key algorithm is proposed to negotiate by both E2E energy-limited smart devices providing a flexible choice for each M2M communication session. The proposed SGA could provide a mutual authentication mechanism and prevent the key guessing attack, the undetectable on-line key guessing attack, the data privacy attack and the relay attack. Furthermore, the suggested SGA could be a newly standard interface for satisfying the basic security requirements of the M2M service layer.

Acknowledgment

This work was supported in part by the Ministry of Science and Technology, Taiwan, Republic of China, under Grant MOST 104-2221-E-468-002.

References

[1] L. Atzori, A. Iera, G. Morabito, The internet of things: a survey, Elsevier Computer Networks, vol. 54, no. 15Oct. 2010.
[2] Riccardo Bonetto, Nicola Bui, Vishwas Lakkundi, Alexis Olivereau, Alexandru Serbanati, Michele Rossi, Secure communication for smart IoT objects: protocol stacks, use cases and practical examples, 2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)2012 1–7.
[3] Quandeng Gou, Lianshan Yan, Yihe Liu, Yao Li, Construction and strategies in IoT security system, IEEE International Conference on Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), and IEEE Cyber, Physical and Social Computing 2013, pp. 1129–1132.
[4] ITU-T Recommendation Y.2060, Overview of the Internet of Things, 2012.
[5] International Telecommunication Union, M2M service layer: requirements and architectural framework, Focus Group Technical Report of ITU-T Focus Group on M2M Service Layer, Telecommunication Standardization Sector Of ITU, FG M2M, D2.1 — Version 1.0, April 2014.
[6] International Telecommunication Union, M2M service layer: APIs and protocols overview, Focus Group Technical Report of ITU-T Focus Group on M2M Service Layer, Telecommunication Standardization Sector Of ITU, FG M2M, D3.1 — Version 1.0, April 2014.
[7] J. Park, N. Kang, Designing a secure service manager for internet of things, Advanced Science and Technology Letters, Vol.43 (Multimedia 2013) 2013, pp. 162–165.
[8] Inshil Doh, Jiyoung Lim, Shi Li, Kijoon Chae, Pairwise and group key setup mechanism for secure machine-to-machine communication, Computer Science and Information Systems 11 (3) (2014) 1071–1090.
[9] MobiThinking, "Global Mobile Statistics 2014 Home: All The Latest Stats on Mobile Web, Apps, Marketing, Advertising, Subscribers, and Trends: Smartphone Shipments/Forecasts by Operating System Market Share", http://mobithinking.com/mobile-marketing-tools/latest-mobile-statsAugust 31, 2014 (On-line).
[10] H.C. Chen, Secure multicast key protocol for electronic mail systems with providing perfect forward secrecy, Security and Communication Networks 6 (1) (January, 2013) 100–107.
[11] H.C. Chen, A.L.V. Epa, A rotation session key-based transposition cryptosystem scheme applied to mobile text chatting, Proceedings of The 28th IEEE International Conference on Advanced Information Networking and Applications (AINA2014) May 13-16 2014, pp. 497–503, http://dx.doi.org/10.1109/AINA.2014.163 (Victoria, Canada).
[12] A. Loukas, D. Damopoulos, S.A. Menesidou, M.E. Skarkala, G. Kambourakis, S. Gritzalis, MILC: a secure and privacy-preserving mobile instant locator with chatting, Inf. Syst. Front. 14 (3) (July, 2012) 481–497.
[13] H.C. Chen, C.Y. Yang, H.K. Su, C.C. Wei, C.C. Lee, A secure e-mail protocol using ID-based FNS multicast mechanism, Computer Science and Information Systems, Special Issue on Mobile Collaboration Technologies and Internet Services, volume 11, Issue 3 August, 2014, pp. 1091–1112.
[14] R.N. Akram, R.K.L. Ko, End-to-end secure and privacy preserving mobile chat application, Eighth Workshop in Information Security Theory and Practice: Securing the Internet of Things (WISTP 2014) 2014, pp. 124–139.
[15] H.C. Chen, A. Christiana, A role-based RSA key management approach in a hierarchy scheme, Proceedings of 2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS2014), IEEE, Birmingham, United Kingdom July 2–4, 2014, pp. 258–264, http://dx.doi.org/10.1109/IMIS.2014.32.
[16] H.C. Chen, H.Y. Chuang, An enhanced three-party encrypted key exchange protocol using digital time-stamp, NCM2010: 6th International Conference on Networked Computing and Advanced Information Management, Seoul, Korea August 16–18, 2010, pp. 665–670.
[17] H.C. Chen, H.Y. Chuang, A three-party encrypted key exchange protocol with protected password authentication, The IET International Conference on Frontier Computing–Theory, Technologies and Applications (IET FC 2010) August 4–6, 2010, pp. 275–280 (Taichung, Taiwan).
[18] H.C. Chen, S.S. Tseng, C.H. Mao, C.C. Lee, R. Churniawan, An approach for detecting flooding attack based on integrated entropy measurement in e-mail server, The 8th International Conference on Embedded and Multimedia Computing (EMC-2013), Taipei, Taiwan, August 2013.
[19] M. Toorani, SMEmail — a new protocol for the secure e-mail in mobile environments, Telecommunication Networks and Applications Conference (ATNAC 2008), Australasian 2008, pp. 39–44.
[20] W. Stallings, Network Security Essentials: Applications and Standards, Prentice Hall, 2000.
[21] E. Cole, R. Krutz, J.W. Conley, Network Security Bible, Wiley Publishing Inc., 2005
[22] R.J. Sutton, Secure Communications, Applications and Management, John Wiley & Sons, 2002.