J Supercomput (2018) 74:3521–3535
https://doi.org/10.1007/s11227-016-1684-0

Secure IoT framework and 2D architecture for End-To-End security

Jongseok Choi[1] · Youngjin In[1] · Changjun Park[1] · Seonhee Seok[1] · Hwajeong Seo[1] · Howon Kim[1]

Published online: 2 March 2016

Abstract In this paper, we proposed an secure IoT framework to ensure an End-To-End security from an IoT application to IoT devices. The proposed IoT framework consists of the IoT application, an IoT broker and the IoT devices. The IoT devices can be deployed along a board line or a boundary of the area of IoT broker. The IoT broker manages their own devices and aggregates their sensing data. The IoT application provides users with IoT services. To use the IoT services, it needs to access to sensing data. Especially, the case of real-time healthcare services should consider intermediate security issues because medical information of patients is one of very sensitive privacy information. However, most of IoT protocols such as CoAP and MQTT have no concern about the End-To-End security, they only depended on the security of DTLS. Therefore, we proposed a new IoT framework to satisfy the End-To-End security feature under the CoAP communication. The proposed framework encrypts sensitive data by a symmetric encryption and an attribute-based encryption

✉ Howon Kim
howonkim@pusan.ac.kr

Jongseok Choi
js.choi.85@gmail.com

Youngjin In
cubya0104@gmail.com

Changjun Park
pcj006@gmail.com

Seonhee Seok
seokseonhee@gmail.com

Hwajeong Seo
hwajeong84@gmail.com

[1] Pusan National University, 6-Eng.Bldg., Jangjeon 2(i)-dong, Geumjeong-gu,
Busan 609-735, Republic of Korea

for efficiencies of communication and computation costs. In addition, each IoT device has a unique identification used as one of their attributes. Consequently, although the IoT broker is one of the intermediate nodes, it decrypts and shows data only if it satisfies all attributes.

Keywords IoT · IoT platform · End-To-End security · CoAP · ABE

1 Introduction

Recently having progressed about researches of Internet of Things (IoT), interest about an IoT security [9,12,18,26,28] is on the increase. However, most of the researches were ceased in level of survey. In contrast, Raza et al. [21] proposed a key management solution. Henceforth, many researches were conducted in the IoT architectural point of view. There are two approaches on a security of the IoT architecture. The first approach is researches [3,14,22,27] on a systemic security of the IoT architecture. However, these studies have no concern about the End-to-End security. The second approach is in the network security point of view. Ning et al. [19] proposed Information, Physical, and Management security (IPM) on the IoT. A key idea of that is to provide a dynamic security by combining the cyber world, physical world and human social. Jiang [11] aimed to provide the trusting environment and the architecture considered the End-to-End security. However, the practical method for the End-to-End security cannot be presented.

In IoT environment, it consists of big three entities: IoT device, IoT broker, and IoT application. The IoT application is an entity which substantially provides users with IoT services such as healthcare or energy conservation. A provider who provides an IoT application for the public also generally has high computability and communication capabilities easily able to communicate a message. An IoT broker has high computability and communication capabilities as an entity which relies communication between IoT applications and IoT devices. The IoT devices, otherwise, have constrained computability and communication capabilities compared with IoT applications and IoT brokers able to have high computability and communication capabilities. These properties of each entity are important components of entire framework of the IoT.

Because the IoT application and the IoT broker have high specifications, they can communicate by the normally used RESTful protocol such as HTTP or SOAP. An existing RESTful protocol is not suitable for IoT devices because of the demand for high process ability and communication capabilities. Therefore, the research has progressed much about lightweight protocols [4,13,23] for replacing RESTful protocol such as HTTP or SOAP. Especially, Constrained Application Protocol (CoAP) [23] and Message Queue Telemetry Transport (MQTT) [10,15] are mainly used protocols on communication between the IoT broker and the IoT devices.

CoAP, a lightweight protocol applied to REST communication such as HTTP, is one of the most used protocols in the domain of the current IoT environment. CoAP supports four types of REST methods: GET, POST, PUT, DELETE and provides a function of HTTP–CoAP proxy in the standard. CoAP provides a CoAP security channel using DTLS instead of TLS which is a security for HTTP because of using communication based on UDP contrary to HTTP.

Therefore, seeing the entire framework of the IoT, the IoT application uses a protocol providing various functions such as HTTP and SOAP, and the IoT devices use a protocol demanding low process ability and low traffic such as CoAP and MQTT. The IoT broker provides protocols such as HTTP and SOAP for mediating role between the IoT application and the IoT devices and supports lightweight protocols such as CoAP and MQTT. CoAP provides a proxy function for connecting smoothly between high specifications protocols and lightweight protocols as well.

However, there is a drawback that, in case that CoAP uses a proxy and a security [6,21] in the way used originally, CoAP satisfies the peer-to-peer security but does not do the End-to-End security. The case of HTTP or SOAP uses mainly TLS [5,16] for the security as communication protocols based on TCP and generates TLS sessions between the IoT application and the IoT broker. It uses DTLS [17] because lightweight protocols based on UDP are used between the IoT broker and the IoT devices and generates DTLS sessions between the IoT broker and the IoT devices for that.

In other words, the IoT broker performs a task of changing from data of TLS sessions to DTLS sessions for seamless communication between the IoT application and the IoT devices and it is difficult to protect that the IoT broker watches payloads as existing security specifications. A HTTP–CoAP proxy and a security requirement is defined in the CoAP standard. However, the standard can only satisfy the security over peer-to-peer communication, not end-to-end.

In this paper, we design an IoT framework and a 2D IoT architecture for providing the End-to-End security for overcoming a limitation of the HTTP–CoAP proxy and propose a new model of the HTTP–CoAP security. To provide the End-to-End security with compatibility of HTTP and CoAP, the simplest way is to encrypt payloads of HTTP. Encryption methods used in TLS require to share a key between two nodes before communication. The proposed scheme performs encryption using attributes of nodes communicating with each other. It is Attribute-Based Encryption (ABE) [2,8,20,24]. In the proposed model, selected attributes can be contained into a cipher text using Ciphertext-policy Attribute-Based Encryption (CP-ABE) [1,7,25]. One of the features of CP-ABE is that the size of a cipher text than a plaintext increases because the cipher text includes attributes. Therefore, payloads can be encrypted by Advanced Encryption Standard (AES) and the usage key of AES is encrypted by ABE with selected attributes. This feature minimizes difference of the size between cipher payloads and plain payloads. The proposed model is organized on highly three types of phases: Issue, Attributes estimation, Communication. We adopt IoT Certificate Authority (CA) on the proposed model. The IoT CA creates security parameters and issues attribute certificate in Issue phase. In Attributes exchange phase attributes of the IoT devices are aggregated and an aggregator, which wants to send secure data, selects and estimates attributes to encrypt a key of AES. In Communication phase, a sender and a receiver can securely communicate with each other using AES and ABE.

It leads three advantages to adapt our proposed framework for the IoT services. These advantages are as follows:

- Our framework provides the IoT services with the End-To-End security feature from the IoT application to the IoT devices based on communications of HTTP and CoAP protocols.
- Usage of ABE makes the IoT application select IoT devices that can decrypt sent messages. In other words, the IoT application does not need several keys to send each other devices cipher messages.
- Our 2D architecture allows the IoT broker to enable security functions in the entire layers from the communication layer to the application layer because the security layer is vertically placed.

The remainder of this paper is organized as follows: we discuss related works: ABE and CoAP in Sects. 2 and in 3 we present an IoT framework, 2D architecture of the IoT broker and the End-To-End security from IoT application to IoT device. In Sect. 4, we give the experiment environment of the proposed scheme for the End-to-End security and we prove the security of the proposed scheme in Sect. 5. Finally, we conclude with Sect. 6.

2 Preliminaries

In this section, we review Constrained Application Protocol (CoAP) and Attribute-Based Encryption (ABE).

2.1 Constrained Application Protocol (CoAP)

The Constrained Application Protocol is an application layer standard protocol proposed by Working Group (WG) Constrained RESTful Environment (CoRE) of Internet Engineering Task Force (IETF). CoAP is one of the web transfer protocols and light weight protocols. It is popularly used in IoT environment, which has constrained nodes and poor network capabilities. The protocol is designed for machine-to-machine (M2M) applications such as smart energy and building automation. In the Internet of Things environment, there are many constrained devices and constrained networks and they require lightweight protocols such as CoAP because they do not have enough performances for using heavy communication protocols such as TCP and HTTP. CoAP can be easy of access because of CoAP following features of Representational Status Transfer (RESTful) used for HTTP. CoAP transmits messages over UDP in the transport layer. CoAP supports various options such as unicast, multicast and broadcast. For the security, Datagram Transport Layer Security (DTLS) can be used instead of TLS because CoAP is one of protocols over UDP. It is registered as “CoAPs” in Internet Assigned Number Authority (IANA) policy. CoAP is defined as four types of message type. Those are Confirmable, Non-confirmable, Acknowledgement, and Reset. CoAP Message Format consists of a very simple message header and an option header for reducing the size of data transmitting. The headers of CoAP consists of 2-bit version, 2-bit message type, 4-bit token field length, 8-bit code, 16-bit message ID, 0 8-byte token and option. Therefore, the minimum size of the header can be 32 bits.

2.2 ABE

Attribute-Based Encryption is an extended concept of Identity-Based Encryption (IBE). ABE is based on Group and Access Tree of attribute information for encryption. ABE is cryptography that a user only who has sufficient attribute value about encrypted data can decrypt this data. In ABE, The Access Tree can be used to decide whether to authorize access to data or not. It denotes whether to authorize or not by inputting an attribute group. It can generate a new access authorization by combination of attribute values which can provide various access services easily.

Generally, there are two types of ABE according to the place of Access Tree. One is Key-Policy Attribute-Based Encryption (KP-ABE) which designates an Access Tree on a key in case of generating a private key and the other is Cipher text-Policy Attribute-Based Encryption (CP-ABE) which designates an Access Tree on a cipher text in case of encrypting a plaintext. The biggest difference between KP-ABE and CP-ABE is the authorization that the access of a user can be controlled. Let you know this reason by understanding each cryptography on the next steps. A cipher text of KP-ABE consists of an attribute group and encrypted data. A private key of a user consists of an Access Tree based on attributes provided to users by a key provider. A user can decrypt a cipher text only if an Access Tree which a user has matches attributes of a cipher text. KP-ABE is that the private key issued to users includes Access Tree so a person who wants to encrypt a data does not have an access authorization of the user. The sender does not know which attributes the receiver has because the Access Tree is designated in case of generating a private key. Therefore, a person who wants to encrypt own data for sending it to specific target cannot designate a receiver and control the access authorization. CP-ABE is the way that solves the problem of a user access authorization which KP-ABE has. CP-ABE is that a sender can designate an Access Tree based on the attribute group of a receiver to receivers. A cipher text of CP-ABE consists of an Access Tree about attribute value for the decryption and encrypted data, and a private key of a receiver consists of its attributes. The attribute value that the receiver has can decrypt a cipher text in case of matching the Access Tree included on the cipher text.

The CP-ABE scheme consists of four fundamental algorithms: Setup, Encrypt, KeyGen, and Decrypt.

$Setup(1^k)$ outputs MK(Master Key) and PK(Public Key). As the first step, The setup algorithm selects a bilinear group G_0 of prime order p and chooses a generator g. In the next step, it randomly chooses two exponents $\alpha, \beta \in Z_p$. At this, the public key is Eq. 1.

$$PK = G_0, \quad g, h = g^{\beta}, \quad f = g^{1/\beta}, \quad e(g, g)^{\alpha} \tag{1}$$

and the master key is $MK = (\beta, g^{\alpha})$.

$Encrypt(PK, M, T)$ outputs the cipher text CT corresponding a message M inputted the public key PK and Access Tree T. The algorithm chooses a polynomial q_x for each node x in the access tree T. Once the polynomials are chosen, it is started from the root node R in a top-down manner. In addition, the degree d_x of the polynomial q_x is set to the node x. Cipher text CT can be computed by Eq. 2 where Y is the set of leaf nodes in T.

$$CT = (T, \tilde{C} = Me(g,g)^{\alpha s}, \quad C = h^s, \quad \forall y \in Y : C_y = g^{q_y(0)},$$
$$C'_y = H(att(y))^{q_y(0)}) \tag{2}$$

$KeyGen(MK, S)$ outputs Secret Key (SK) with input a set of attributes S. a random $r, r_j \in Z_p$ are chosen for each attributes $j \in S$. After that, the secret key can be computed by Eq. 3

$$SK = (D = g^{\alpha+r)/\beta}, \quad \forall j \in S : D_j = g^r \cdot H(j)^{r_j}, \quad D'_j = g^{r_j}) \tag{3}$$

$Decrypt(CT, SK)$ outputs the message M corresponding the cipher text CT. We let $i = att(x)$ where the node x is a leaf node. And if $i \in S$, Eq. 4 is performed.

$$\begin{aligned} DecryptNode(CT, SK, x) &= \frac{e(D_i, C_x)}{e(D'_i, C'_x)} \\ &= \frac{e(g^r \cdot H(i)^{r_i}, h^{q_x(0)})}{e(g^{r_i}, H(i)q_x(0))} \\ &= e(g,g)^{rq_x(0)} \\ &= e(g,g)^{rs}. \end{aligned} \tag{4}$$

Otherwise, $DecryptNode(CT, SK, x) = \perp$. If the access tree is satisfied by S, we can obtain the message M by Eq. 5

$$\tilde{C}/(e(C, D)/A) = \tilde{C}/(e(h^s, g^{\alpha+r)/\beta})/e(g,g)^{rs}). \tag{5}$$

3 Proposed architecture

In this section, we present an IoT framework, a 2D architecture of IoT broker and an End-to-End security. Firstly, a definition of the IoT framework is described in Sect. 3.1. In Sect. 3.2, the 2D architecture is presented. Finally, we explain the procedure of End-to-End security in Sect. 3.3.

3.1 IoT framework

In this section, we present the IoT framework composed of the IoT devices, the IoT brokers and the IoT application. The IoT devices can be physically deployed in the point that there are the information we want to aggregate. In addition, the devices have a low computational ability and communicate over heterogeneous network protocols. For these reasons, we adopt the IoT broker as the intermediary among the IoT devices and the IoT services. The IoT broker supports multiple communication stacks, such as 6LoWPAN, ZigBee, Bluetooth, WiFi, etc, and various protocols and proxy, such as

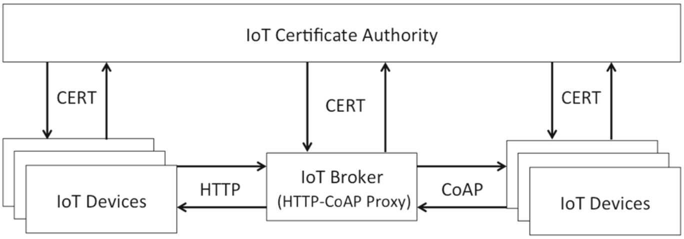

Fig. 1 IoT framework

CoAP, MQTT, HTTP, etc. As the final entity, the IoT application practically provides the IoT services to users.

Figure 1 describes an entire structure of the IoT framework. As we mentioned above, the IoT framework consists of the IoT application, the IoT broker and the IoT devices. In this framework, the HTTP can be used for delivering a data between the IoT application and the IoT broker. After that the CoAP can be used as communication protocols among the IoT broker and the IoT devices. So the IoT broker should perform the role of a proxy because two or more kinds of protocols can be needed in the IoT framework like this. In the proposed framework, we adopt the IoT broker as the HTTP–CoAP proxy.

Commonly, HTTP and CoAP employ TLS or DTLS for the secure communication. But in the aspect of the total solutions, the End-to-End security from the IoT application to the IoT devices cannot be satisfied. In other words, there is a unsecure point. The IoT broker decrypts and re-encrypts a payload when switching HTTP to CoAP because sessions of TLS and DTLS are unconditionally split.

3.2 The 2D architecture of IoT broker

IoT-A is one of the typical associations defining the reference architecture of the IoT. A Security layer in the IoT-A is also one of the important layers of the reference architecture. In this architecture, it only considers a security in the peer-to-peer security. However, the sensitive data which include privacy such as health information have to be protected in all communication channel.

To prevent to reveal the sensitive data on intermediate devices or nodes, we designed a architecture of the IoT broker. In this section, we present the proposed two-dimensional architecture of the IoT broker. Figure 2 shows the architecture of the IoT broker. There are two big layers: a horizontal layer and a vertical layer. Our proposed architecture splits into two vertical layers. In the first vertical layer, there are four horizontal layers: a application layer, a proxy layer, a middleware layer, a communication layer. The second vertical layer is a security layer. The security layer covers all of the first vertical layer.

In the application layer, RESTful protocols are popularly used such as HTTP, SOAP, etc. Since data in typical RESTful protocols, such as HTTP and SOAP, are represented by markup languages such as XML, these protocols are unsuitable to the IoT devices, which have a poor computability and a small memory space. For this reason, the

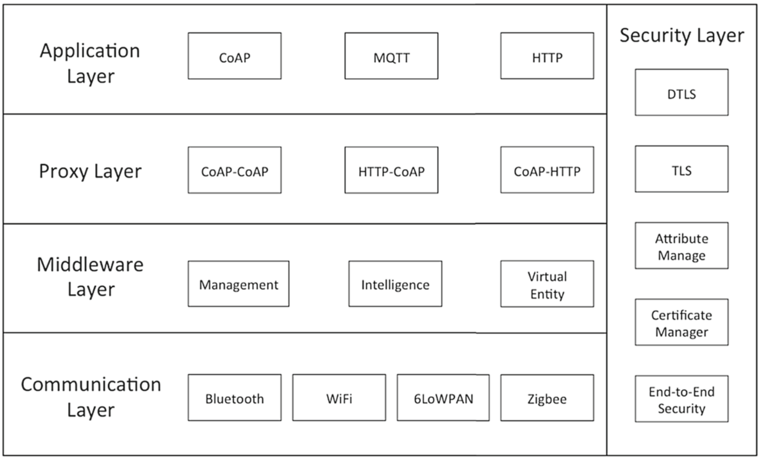

Fig. 2 2D architecture of IoT broker

IoT broker and the IoT devices communicate by light weight protocols such as CoAP, MQTT, LWM2M, etc. Especially, because CoAP supports the RESTful methods: GET, POST, PUT, DELETE with low communicational cost, CoAP is one of the popular protocols in the IoT area. In addition, the standard of CoAP has concerned about security issues and CoAP can be integrated with DTLS, which is cryptographically secure protocols over UDP communication. However, only using DTLS for the security of CoAP has a restriction on secure communication of full path from the IoT application to the IoT devices.

Therefore, we adopt a ABE for a AES key and AES to the encrypt payload of CoAP. In ABE scheme, a payload can be encrypted with attributes and ABE allows satisfier who has attributes defined in encrypted data to decrypt the data. Due to the reason, only selected nodes, which have chosen attributes, can extract the plain data from the encrypted payload in our framework.

The middleware layer manages users and devices. The middleware layer security includes an authentication, an authorization and ABE. Especially, the protocol management includes the HTTP–CoAP proxy. When the payload re-encryption is needed, it can be performed in the protocol management with interaction of the security. Once the payload is re-encrypted, attributes of the payload can be modified.

3.3 End-to-End security

In this section, we present the method for the End-to-End security from the IoT application to the IoT devices. As we mentioned in Sect. 3.2, because the sessions of HTTP and the sessions of CoAP are unconditionally split, the payload should be re-created in HTTP–CoAP proxy. For this reason, it needed fully secure protocols from the IoT application to the IoT devices with compatibility of HTTP and CoAP.

To ensure the End-to-End security, we adopt ABE for the payload encryption. Because ABE scheme allows entities having specified attributes to decrypt the payload, it is possible to share a key without key establishment with the IoT broker. In other words, the IoT broker can decrypt the payload only if having the attributes specified to the encrypted payload. One of the weaknesses of ABE is a big size of a cipher text. Therefore, the session key for AES is only encrypted by ABE and the payload is encrypted by AES.

The proposed scheme consists of an issue phase, an attribute exchange phase, a secure communication phase. In issue phase, all of IoT entities including the IoT devices, the IoT brokers and the IoT applications are verified by the IoT CA (IoT certificate authority). The IoT CA creates and issues certificates of verified entities with their attributes. The attribute exchange phase is performed before encryption and needed to encrypt a session key with ABE. In this phase, a sender requests the attributes to receivers and the sender selects specified attributes from responded attributes for elected receivers. Finally, the sender and the receivers use encrypted payload and encrypted a session key to communicate with each other. The processes of each phase are described below in details.

3.3.1 Issue phase

In this phase, the IoT CA chooses attributes of IoT entities and issues their certificates to them. This phase has to be performed in the secure channel. The detail progress is described in below:

$setup(attr)$ chooses attributes of the entity to create a certificate. A combination of selected attributes should represent the entity as unique in the IoT framework. A device type, a location, a serial, a role, etc. can be used as attributes of the certificate.

$issue(cert)$ creates and issues the certificate with selected attributes in $setup(attr)$.

3.3.2 Attribute exchange phase

This phase can be started by two type of request methods:unicast and broadcast. Below is the description of the attribute estimation phase:

$req(attr)$ requests attributes to receivers over unicasting or broadcasting a attributes collection message.

$res(attr)$ responses a their attributes. If the $req(attr)$ was broadcasted, all entities broadcast the their ID and the attributes.

$$responsemessage = \{ID_i, attr_0, attr_1, \ldots, attr_t\} \tag{6}$$

$est(attire)$ The sender selects the attributes for the end node that the sender allows to decrypt payload. Once the sender chooses attributes, attributes of other nodes should be considered to satisfy uniqueness of the selected attributes.

3.3.3 Communication phase

In this phase, the IoT application and the IoT device communicate with ABE and AES. To encrypt the payload, the proposed scheme uses the AES algorithm and the secret key of AES is encrypted by ABE with selected attributes in the attribute exchange phase.

$keygen(1^k)$ randomly chooses the key for AES encryption. A k is the key length of AES, which can be 128, 192 and 256.

$enc_{sk}(p, T)$ encrypts the payload by AES and a sk by ABE with the certificate of the sender, a $cert_s$, where the sk is generated key from $keygen(1^k)$ and let T and p are time stamp and payloads, respectively.

$$ep = ABE_{cert_s}(sk, attr)||AES_{sk}(p||T)||T \tag{7}$$

$comm(ep)$ sends the ep, which is calculated by Eq. 7, to the end node through the IoT broker. At this, the IoT broker can decrypt and know the payload only if it satisfy all attributes of embedded attributes in cipher text.

$dec(ep)$ decrypts the key and the payload. At first ep is split into two parts: ABE and AES. Let a ABE part and a AES part be α and β, respectively. For decryption, the sk should be extracted with certificate of receiver, $cert_r$, and payload can be decrypted using it. Equation 8 shows decryption progress.

$$\begin{aligned} sk'||T' &= ABE_{cert_r}(\alpha, attr) \\ p' &= AES_{sk'}(\beta). \end{aligned} \tag{8}$$

4 Experimental results

In this section, we introduce experimental results of the proposed scheme. For the experimentation, we used a PC, an ODROID XU3 and a raspberry Pi Model B as an IoT application, an IoT broker and an IoT device, respectively. Table 1 gives specification of each system. As we can see in Table 1, the IoT application has i7 4790 3.6 GHz processor and 8 GB memory. The IoT broker has ARM Cortex A15 2.0 GHz quad cores and 2 GB memory. Although the IoT broker has lower computability than the IoT application, the IoT broker also has redundant computability. However, the IoT device has ARM11 700 MHz and 256 MB memory. In the practical IoT environment, the IoT device is more constraint than a Raspberry Pi Model B such as an Arduino.

The IoT application and the IoT broker communicate by HTTP. At this, AES and ABE are performed by Javascript, the client side language of the IoT application. After

Table 1 Specification of IoT entities

	IoT application	IoT broker	IoT device
Processor	i7 4790 3.6 GHz	Cortex-A15 2.0 GHz	ARM11 700 MHz
RAM	DDR3 8 GB	LPDDR3 2 GB	512 MB
Storage	SSD 128 GB	Micro SD 16 GB	Micro SD 8 GB
Network	1 Gbps	100 Mbps	100 Mbps

Table 2 Process times

	Encryption	Decryption	Response time (no encryption)	Response time (proposed)
1 MB	1021.20	5443.08	4985	11,583
5 MB	1021.93	6392.94	7136	14,254
10 MB	1022.38	7681.08	12,085	23,023
15 MB	1022.50	9113.14	23,167	32,109

the encryption, the IoT application sends cipher payloads to IoT broker. Except the payloads having the broker as the final destination, the IoT broker simply forwards to the IoT device. Finally, the IoT device extracts a key and then decrypts the payloads upon receiving the cipher message. Table 2 shows process time for size of payloads. Encryption time measured total time of generating a key for AES, encryption of payloads by AES and encryption of the key by ABE. Because the encryption is performed in the IoT application, it does not have a strong influence on the entire response time. Decryption time measured total amount of time of extraction of the key by ABE and decryption of the payload by AES. Because the case of decryption is performed in the IoT device, decryption time exerts a strong influence on the response time. In Table 2, response time measured total amount of time from starting request to finishing the response. We tested all cases with a private network, therefore all measured time cannot reflect routing feature of network. Table 2 gives two measurements of response time. The first response time of them was written without encryption. In other words, this response time is original response time of HTTP. The second response time is measurement of the proposed scheme. As we can see, response time of our scheme was delayed about decryption time of the IoT device.

5 Security analysis

In this section, we evaluate the security on a malicious broker, an eavesdropping, a spoofing attack and a replay attack.

5.1 Malicious broker

In the case of wearable IoT devices, they can be connected to other IoT broker except their own IoT broker because the devices are dynamically moved according to moving ranges of users. In this case, we need to consider the device to connect to malicious broker. The other case is that their own broker was infected by malicious codes. In both cases, the IoT broker can try to aggregate the sensitive information of users from wearable devices illegally. Especially the revelation of pictures and health information can lead to make a loss or negative honor. The HTTP–CoAP proxy in the standard on CoAP supports a secure peer-to-peer communication but if the IoT broker is malicious, the sensitive information can be revealed. In proposed model, the payload is encrypted by AES and the key of AES is encrypted by ABE to prevent illegal access. Therefore, the malicious broker can decrypt the payload only if the broker can solve Eq. 9.

$$\begin{aligned} Decrypt(CT, SK, x) &= \frac{e(g^r \cdot H(i)^{r_i}, h^{q_x(0)})}{e(g^{r_i}, H(i)q_x(0))} \\ &= e(g, g)^{rq_x(0)} \\ &= e(g, g)^{rs} \end{aligned} \tag{9}$$

To obtain the $e(g, g)^{rs}$, the malicious broker can firstly find the rs. However, finding the rs from the $e(g, g)^{rs}$ is the same with the difficulty of discrete logarithm. Therefore, the proposed scheme is mathematically secure on the malicious broker attack.

5.2 Eavesdropping

An attacker can eavesdrop the payload by network monitoring of users and he/she can obtain Eq. 11. After that he/she tries to obtain the useful and meaningful information from the payload. In other words, the eavesdropper tries to decrypt the cipher text by AES with the key, sk. For this, he/she should extract the sk from $ABE_{cert_s}(sk||T)$. However, as we mentioned in Sect. 5.1, he/she should solve the discrete logarithm problem.

$$payload = \{ABE_{cert_s}(sk, attr)||AES_{sk}(p||T)||T\}. \tag{10}$$

5.3 Spoofing attack

The malicious IoT devices can spoof their attributes or network information such as a Media Access Control (MAC) or an IP address in Attribute Exchange phase. The IoT Application, which requested attributes and ID, allows them as correct information and estimate the final attributes specified to cipher text without any sense of spoofed information. At the next step, the IoT application encrypts a key of AES with attributes, which is estimated with the spoofed information, by ABE and sends the IoT broker cipher payload. However, once malicious IoT devices decrypt the cipher payload, they cannot obtain any meaningful information from the encrypted payload because although they can spoof their attributes in attribute exchange phase, they cannot modify their practical attributes in their certificate. The entire certificate can be managed only by the IoT CA.

5.4 Replay attack

The replay attack is sending previous message to IoT devices again. In this scheme, an attacker can replay to send the same message to the same device. For this attack, our proposed scheme uses time stamp to ensure a freshness of messages. To do the replay attack, the attacker should modify the time stamp of payload, Eq. 11. Let the attacker change the time stamp T to new one T'. The payload will be as Eq. 11.

$$payload' = \{ABE_{cert_s}(sk, attr)||AES_{sk}(p||T)||T'\} \tag{11}$$

Upon receiving this payload, the IoT device extracts sk and T. And before decryption of AES, the device verifies the time stamp by Eq. 12.

$$\begin{aligned} T_d - T' &< \Delta T \\ sk &\leftarrow ABE_{cert_r}(ep_{abe}) \\ p, T &\leftarrow AES_{sk}(ep_{aes}) \\ T_d - T &< \Delta T \end{aligned} \tag{12}$$

As you can see in Eq. 12, although the attacker change past time stamp T to the valid one, he/she cannot modify encrypted time stamp T. In other words, encrypted time stamp ensures integrity of the time stamp because it can be changed by who an entity has the same certificate from used certificate encrypting the payload.

6 Conclusion and future works

The interest of the security element was increased according to the increase in the research on IoT. There are three entities of IoT environments—IoT application, IoT broker and IoT devices. Due to the high-performance computing and high-speed communication capability, the IoT application has rare constrained elements. The IoT broker also could have high-performance computing and high-speed communication capability. Therefore, REST protocols supporting various features such as HTTP and SOAP used for network communication already could be used between the IoT application and the IoT broker. However, since IoT devices have low computing performance and communicating capability, HTTP and SOAP protocols are not properly used. Thus, many research about lightweight protocols was processed to supplement this limited circumstances. CoAP and MQTT are representative lightweight protocols. Whereas HTTP protocol used between the IoT application and the IoT broker is TCP-based communication, CoAP and MQTT protocols are based on UDP communication used between the IoT broker and IoT devices. This difference restrains providing the End-to-End security from the IoT application to IoT devices.

In this paper, we defined the IoT framework to offer the End-to-End security under IoT environment and designed the architecture of IoT broker. In addition, we used CP-ABE and AES to encrypt payload and keys to communicate between the IoT application and IoT devices without decryption at the IoT broker. In our proposed model, we divided a proxy layer and have performed mutual action with the security layer at the IoT broker. And we considered the End-to-End security on whole IoT framework by including ABE in the security layer between IoT brokers. In our framework, we introduced IoT CA as a managing and issuing institution to use ABE use an attribute certificate. As an experiment result for this model, we used a PC as the IoT application, an ODROID XU3 as an IoT broker and a Raspberry Pi Model B as an IoT device to measure the process time during the encryption using AES and ABE in the IoT application and decryption in the IoT device. When we compared response time about our model and other model without encryption between the IoT application and IoT devices, there was a little delay. But our model provides safeties from eavesdropping and a malicious IoT broker.

To serve commercial services, the End-to-End security between the IoT application and the IoT devices must be provided and need to reduce response time to provided large services by researching lightweight ABE. Further research about efficient caching and distribution techniques between the IoT device and the IoT broker is needed to assure real-time services because of the IoT device under the highly restricted condition.

References

1. Bethencourt J, Sahai A, Waters B (2007) Ciphertext-policy attribute-based encryption. In: IEEE symposium on security and privacy, SP'07. IEEE, pp 321–334 (2007)
2. Chase M (2007) Multi-authority attribute based encryption. In: Theory of cryptography. Springer, pp 515–534
3. Chuankun W (2010) A preliminary investigation on the security architecture of the internet of things [j]. Bull. Chin. Acad. Sci. 4:009
4. Cirani S, Picone M, Veltri L (2015) Mjcoap: an open-source lightweight java coap library for internet of things applications. In: Interoperability and open-source solutions for the internet of things. Springer, pp 118–133
5. Dierks T (2008) The transport layer security (tls) protocol version 1.2. Technical report, RFC 5246, July 2008
6. Gerdes S, Bergmann O, Bormann C (2014) Delegated coap authentication and authorization framework (dcaf). IETF draftgerdes-core-dcaf-authorize-02
7. Goyal V, Jain A, Pandey O, Sahai A (2008) Bounded ciphertext policy attribute based encryption. In: Automata, languages and programming. Springer, pp 579–591
8. Goyal V, Pandey O, Sahai A, Waters B (2006) Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM conference on computer and communications security. ACM, pp 89–98
9. Heer T, Garcia-Morchon O, Hummen R, Keoh SL, Kumar SS, Wehrle K (2011) Security challenges in the ip-based internet of things. Wirel Pers Commun 61(3):527–542
10. Hunkeler U, Truong HL, Stanford-Clark A (2008) Mqtt-s—a publish/subscribe protocol for wireless sensor networks. In: 3rd international conference on communication systems software and middleware and workshops, 2008. comsware 2008. IEEE, pp 791–798
11. Liu Y, Hu W, Du J (2011) Network Information Security Architecture Based on Internet of Things. ZTE Technol J. 17(1):17–20
12. Khoo B (2011) Rfid as an enabler of the internet of things: issues of security and privacy. In: Internet of things (iThings/CPSCom), 2011 international conference on and 4th international conference on cyber, physical and social computing. IEEE, pp 709–712
13. Lee JY, Lin WC, Huang YH (2014) A lightweight authentication protocol for internet of things. In: 2014 international symposium on next-generation electronics (ISNE). IEEE, pp 1–2
14. Li X, Xuan Z, Wen L (2011) Research on the architecture of trusted security system based on the internet of things. In: 2011 international conference on Intelligent computation technology and automation (ICICTA), vol. 2. IEEE, pp 1172–1175
15. Banks A, Gupta R (2014) MQTT version 3.1.1. OASIS standard
16. McGrew D, Bailey D (2012) Aes-ccm cipher suites for transport layer security (tls). Technical report, RFC 6655, Aug 2012
17. McGrew D, Rescorla E (2010) Datagram transport layer security (dtls) extension to establish keys for secure real-time transport protocol (srtp). Technical report, RFC 5764, May 2010
18. Medaglia CM, Serbanati A (2010) An overview of privacy and security issues in the internet of things. In: The internet of things. Springer, pp 389–395
19. Ning H, Liu H et al (2012) Cyber-physical-social based security architecture for future internet of things. Adv Internet Things 2(01):1

20. Pesonen LI, Eyers DM, Bacon J (2007) Encryption-enforced access control in dynamic multi-domain publish/subscribe networks. In: Proceedings of the 2007 inaugural international conference on distributed event-based systems. ACM, pp 104–115
21. Raza S, Voigt T, Jutvik V (2012) Lightweight ikev2: a key management solution for both the compressed ipsec and the ieee 802.15. 4 security. In: Proceedings of the IETF workshop on smart object security
22. Riahi A, Challal Y, Natalizio E, Chtourou Z, Bouabdallah A (2013) A systemic approach for iot security. In: 2013 IEEE international conference on distributed computing in sensor systems (DCOSS). IEEE, pp 351–355
23. Shelby Z, Hartke K, Bormann C (2014) The constrained application protocol (coap). Technical report, RFC 7252, June 2014
24. Wang G, Liu Q, Wu J (2010) Hierarchical attribute-based encryption for fine-grained access control in cloud storage services. In: Proceedings of the 17th ACM conference on computer and communications security. ACM, pp 735–737
25. Waters B (2011) Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Public key cryptography-PKC 2011. Springer, pp 53–70
26. Weber RH (2010) Internet of things-new security and privacy challenges. Comput Law Secur Rev 26(1):23–30
27. Wei R (2012) A study of security architecture and technical approaches in internet of things. Netinfo Secur 5:025
28. Zhao K, Ge L (2013) A survey on the internet of things security. In: 2013 9th international conference on computational intelligence and security (CIS). IEEE, pp 663–667