IoT-FBAC: Function-based access control scheme using identity-based encryption in IoT

Hongyang Yan, Yu Wang, Chunfu Jia, Jin Li, Yang Xiang, Witold Pedrycz

PII: S0167-739X(18)30997-X
DOI: https://doi.org/10.1016/j.future.2018.12.061
Reference: FUTURE 4682

To appear in: *Future Generation Computer Systems*

Received date : 27 April 2018
Revised date : 7 September 2018
Accepted date : 29 December 2018

Please cite this article as: H. Yan, Y. Wang, C. Jia et al., IoT-FBAC: Function-based access control scheme using identity-based encryption in IoT, *Future Generation Computer Systems* (2019), https://doi.org/10.1016/j.future.2018.12.061

IoT-FBAC: Function-Based Access Control Scheme using Identity-Based Encryption in IoT

Hongyang Yan[1], Yu Wang[2], Chunfu Jia[1]*, Jin Li[2], Yang Xiang[3], Witold Pedrycz[4]

[1] *College of Computer and Control Engineering, Nankai University, China*

[2] *School of Computer Science, Guangzhou University, China*

[3] *School of Software and Electrical Engineering, Swinburne University of Technology, Australia*

[4] *Department of Electrical and Computer Engineering, University of Alberta, Canada*

Abstract

The Internet of Things (IoT) has become one of critical parts in our daily life. As a large number of smart things connecting to the Internet, terminals are vulnerable to various attacks. Thus the security of IoT becomes important before they are widely applied. Smart home, as an interesting application of IoT, has attracted more and more attention. However, most of the existing works have focused on the authentication between devices and the home gateway, which are only able to realize coarse-grained access control. In another word, once a device is authenticated, the user can access all the functions of the device. This leads to the over-privilege access behaviour. To solve this problem, we propose a Function-based Access Control scheme in IoT (IoT-FBAC), that uses an Identity-based Encryption (IBE) scheme. The proposed scheme provides fine-grained access control, prevents applications from accessing unauthorized functions. Meanwhile, the cost of each access operation is a constant in IoT-FBAC scheme. The security analysis indicates that the IoT-FBAC scheme is secure, which can prevent over-privilege access. The experiment results demonstrate that the proposed scheme is effective.

Keywords: Internet of Things, Identity-based encryption, Smart home, Privacy security

1. Introduction

In recent years, Internet of Things (IoT) has become a research hotspot as an emerging technology. Naturally, with the increasing of the scale of its expansion, the more smart devices are connected to the Internet. The International Data Corporation (IDC) forecasts that there will be 28 billion IoT devices connecting to the Internet by 2020 [1]. To seize the IoT market, major companies such as Google, Samsung and Apple are establishing their frameworks for connecting various devices [2]. At present, IoT devices have penetrated every aspect of life, and provided convenience for our daily life, such as biological feature recognition [3, 4] and context monitoring [5, 6].

Smart home is a kind of general application scenarios of IoT. Applying various smart devices in smart homes provides convenience for us. For example, when you are not at home, you can remotely monitor the condition of a child or an elderly person; when you arrive at home, the door opens and the air conditioner, television and indoor lights automatically turn on; when you leave home, the security system will arm automatically, the gardening system will automatically water and fertilize and so on. With the development of IoT technologies, increasingly more families are choosing smart home services. Some existing system

*Corresponding author.

Email addresses: hyang.yan@foxmail.com (Hongyang Yan[1]), yuwang@gzhu.edu.cn (Yu Wang[2]), cfjia@nankai.edu.cn (Chunfu Jia[1]), jinli71@gmail.com (Jin Li[2]), yxiang@swin.edu.au (Yang Xiang[3]), wpedrycz@ualberta.ca (Witold Pedrycz[4]) frameworks have recently been developed for users to manage smart things. These frameworks can support a third party to build apps that run on mobile phones to control the connected devices. In this way, users can control smart things more conveniently. Many such frameworks, such as Apple HomeKit [7], Samsung SmartThings [8] and Google Weave/Brillo [9], have been widely used.

However, security is one of the most important problems to be solved in IoT. Because of the limited computing power and storage capability, the use of IoT devices in practice has difficulty supporting the existing security mechanisms. Due to their diverse features and scalability issues, IoT devices are vulnerable by hackers [10]. The various means of attack make the security of IoT more complex [11, 12, 13, 14, 15, 16]. Attackers can manipulate smart devices to obtain sensitive information of users, and even cause physical and financial harm to users. For example, the Dyn DNS company took control of the millions of web cameras produced by a Chinese manufacturer with utilizing common attack methods such as default passwords and the outdated TELNET service [17], this attack leads to the behavior of users of such web cameras to be monitored. Another attack example is that burglars can determine the locations of lucrative homes via the distinctive features of expensive devices and access homes that have vulnerable smart locks [18]. These examples highlight the urgent need to study the security of IoT devices.

1.1. Problem Presentation

Because secure access control while maintaining privacy is a challenging problem, some works have studied this issue from other aspects [19, 20, 21]. In this paper, we focus on studying secure access control in IoT. With apps from the third-party becoming the most popular trend, many researchers have reported that such apps are not safe due to over-privilege access [22]. Felt *et al.* conducted an analysis for Android app market and observed that more than one-third of 940 apps had over-privileged access behavior [23]. Fernandes *et al.* also mentioned this problem when they summarized the security analysis of smart home in [24].

Over-privilege means that when an app performs data access to one authorized function of the device, it will also access the data of other unauthorized functions. This behavior is called over-privilege access. For example, as shown in Figure 1, a smart door lock has three functions: 1) lock/unlock (two actions); 2) door status monitoring (open or closed); and 3) door battery monitoring. The user only authorizes the app to access the battery of the smart door lock. Unfortunately, the app could exploits the other two functions to open the door if the app is malicious. This leads to a serious privacy leakage.

Why does the over-privilege access problem arise? There are two reasons for this problem: *coarse-grained access control* and *coarse app-device binding*. For the first reason, most of the existing IoT frameworks are based on the device-centric approach, which provides an app with either all or no permissions to access devices. This approach is useful for devices that have only a single function. However, with the increase of the number of functions of a device, the app will obtain over-privilege according to this approach. Thus, this method is no longer suitable. For the second reason, due to the design flaws of the existing frameworks, developers define capabilities and commands that are too coarse for users to control. For example, smart-door.lock represents both the status and command of the door. If an app is authorized to access the status of the door, it can also misused the lock command.

Aiming at solving the over-privilege access, Lee *et al.* [25] proposed a FACT scheme which is based on functionality-centric approach to prevent over-privilege access. Their scheme achieves two design goals, one is the principle of least privilege and the other one is availability in terms of device functionalities. However, in essence, their scheme uses the access control list (ACL). It is a bottleneck with the increase in devices because during each access, it requires the cost time at least $O(\mathsf{log}n)$ to search the ACL. Therefore, it is urgent to propose a new access control method to meet the trend of multi-functional devices.

1.2. Our Contributions

To solve the over-privilege access problem, we propose a novel Function-based Access Control scheme in IoT, referred to as the IoT-FBAC scheme. The basic unit is *functionality* in Our scheme. We formalize the contributions of the study as follows:

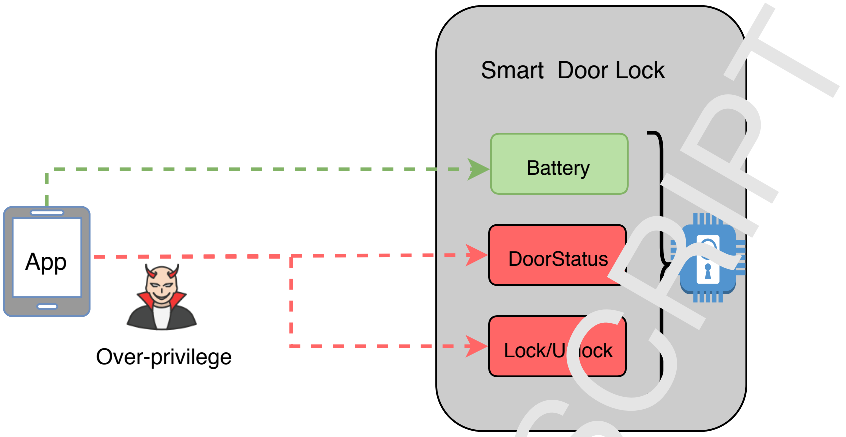


Figure 1: The over-privilege access problem in IoT frameworks

- We present a Function-Based Access Control scheme in IoT, which is called IoT-FBAC. This scheme uses an identity-based encryption (IBE) scheme to encrypt the data generated by devices. In contrast to Lee's scheme, where they used engineering skills, we are the first to solve the over-privilege problem with using a cryptographic technique to our knowledge.

- IoT-FBAC scheme provides fine-grained access control. It prevents the app from accessing unauthorized functions. Compared to the other traditional schemes, IoT-FBAC scheme meets the trend of multi-functional devices.

- By using IBE scheme to encrypt data before uploading to the cloud server, unauthorized app and cloud server provider could not obtain the sensitive data. Thus, data in IoT-FBAC scheme is secure.

Organization. The rest of paper is organized as follows. In section II, we present an overview of existing IoT frameworks and identify their flaws. In section III, we introduce preliminaries of our IoT-FBAC scheme. The smart home framework, including the system model and security model, is defined in section IV. In section V, we present the proposed IoT-FBAC scheme, which uses IBE to prevent over-privilege access. Then, we analyze the security of IoT-FBAC scheme in section VI. The experiment results in section VII demonstrates that our scheme is efficient and practical. Section VIII presents the related work about IoT security. Finally, we conclude this paper and talk about future works in section IX.

2. Overview of Existing IoT Frameworks

In this section, we give two instances of the existing IoT frameworks such as SmartThings and IoTivity. The basic overview and their design flaws are presented as follows.

2.1. SmartThings

SmartThings is the most popular IoT framework which is developed by Samsung [8] in the market. This framework integrates heterogeneous IoT ecosystems and supports approximately 170 IoT devices. A simple SmartThings architecture generally consists of three parts: *The hub, SmartApp,* and *Smart devices.*

- The hub: Because of the diversity of devices, there may be different communication protocols such as Zigbee, Z-Wave between different devices. The hub helps these devices connect to the home network.

- SmartApp: The apps are installed on the user's smart phone, they provide interface for the user to control the devices.

- Smart devices: The devices are connected to the Internet through the hub. They generate data and transfer them to the hub for processing due to the limited storage and computing power.

The basic authorization unit in SmartThings is a **capability**, which consists of two elements: *attributes* and *commands*. The former represents the properties of a device. The latter represents the way that a user controls the devices through SmartApp. For example, in Figure 1, door status and battery are the attributes, and lock/unlock are the commands. SmartThings enables users to control devices by creating an **account**. Within the account, users can add devices by scanning the nearby devices for binding. Under the account, users can divide the devices in different rooms, which can present the physical spaces.

2.2. IoTivity

IoTivity [26] is an open-source IoT framework. This framework is based on the device-to-device connectivity for the IoT devices. There are three parts including *servers, clients,* and *resource hosting devices.*

- Server: It aggregates the data generated by the connected devices.
- Client: It represents a user that attempts to access the device.
- Resource hosting device: It monitors the status of the connected devices, and it helps a user find the addresses of servers.

The basic access unit in IoTivity is **resource**, which has three elements: *identity, property* and *attribute.* Identity is a uniform identifier and includes the address and path of each device. Property includes the resource type and interface of each device. Attribute is the data value of the device functionalities. The access control in IoTivity is determined by an **ACL**, which is maintained in a server. Each access control entry has three items: *subject ID, resource* and *permission.* Subject ID represents the unique identifier of a device. Resource is the resource type of a server. Permission is the privilege type of a client. When a client attempts to access a device, it sends a request to the server, and the server looks up the ACL and search the entry with this request. Then, it allows or denies access according to the permission.

2.3. Design Flaws in Frameworks

Some researchers have discussed the design flaws in the above IoT frameworks.

SmartThings. Fernandes *et al.* analyzed the security problems about smart home in [24]. Here, we concentrate on the two problems about over-privilege in SmartThings.

- Coarse-Grained Capabilities: The existing capabilities that are defined too coarse grained. For an instance, `capability.lock` represents both the lock status attribute and the lock command. It will cause a vulnerable attack if the lock command is misused.
- Coarse Authorization: When an app is authorized by user, it obtains all the data and commands of a device.

IoTivity. There are several access control security flaws in IoTivity according to Sanghak *et al.* in [25].

- All attribute data are stored in the same process or file system.
- Due to the basic control unit is resource, it can not achieve fine-grained access control.

These flaws can cause over-privilege access. Thus, in order to alleviate this shortcoming, we present our IoT-FBAC framework to resist this attack.

3. Preliminaries

For the sake of completeness of the study, we briefly recall the basic concepts that are of interest in this work. Here, we give the basic complexity assumption and IBE scheme.

3.1. Complexity Assumption

Bilinear Map: For a cyclic group $\mathbb{G}$ of prime order p, there is a map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$. This map has the following properties:

- Bilinear: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
- Non-degenerate: $e(g, g) \neq 1$.
- Computability: there is an efficient bilinear map e as shown above. Note that $e(,)$ is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

q-Bilinear Diffie-Hellman Inversion Assumption: The q-BDHI problem in the group $\mathbb{G}$ is defined as follows: given the $(q + 1)$-tuple $(g, g^x, g^{x^2}, ..., g^{x^q}) \in (\mathbb{G}^*)^{q+1}$ as input, then output $e(g, g)^{1/x} \in \mathbb{G}_1^*$. An algorithm $\mathcal{A}$ has advantage ϵ in solving the q-BDHI problem in $\mathbb{G}$ if

$$Pr[\mathcal{A}(g, g^x, g^{x^2}, ..., g^{x^q}) = e(g, g)^{1/x}] \geq \epsilon$$

Similarly, an algorithm $\mathcal{B}$ that outputs $b \in \{0, 1\}$ has advantage ϵ in solving the decision q-BDHI problem in $\mathbb{G}$ if

$$|Pr[\mathcal{B}(g, g^x, g^{x^2}, ..., g^{x^q}, e(g, g)^{1/x}) = 0] - Pr[\mathcal{B}(g, g^x, g^{x^2}, ..., g^{x^q}, T) = 0]| \geq \epsilon$$

Definition 2.1 *The decision (t, q, ϵ)-BDHI assumption holds in $\mathbb{G}$ if there is no t-time algorithm that has an advantage of at least ϵ in solving the decision q-BDHI problem in $\mathbb{G}$.*

3.2. IBE Scheme

IBE is an important primitive of public key encryption cryptography. In the IBE scheme, the public key of a user is the unique information such as identity. The system will initiate and generate secret keys for every user. When a sender sends a message to a receiver, he will encrypt the message using the identity of the receiver. The receiver will obtain the message by using the secret key. Our IoT-FBAC scheme is proposed based on the IBE scheme [27], and has the following four algorithms: *setup, extract, encrypt, and decrypt.*

- Setup: Taking a security parameter k as input, returning system parameters $params$ and secret master key MSK. The system parameters $params$ will be publicly known, whereas the master key MSK is secret.
- Extract: Taking $params, MSK$, and an arbitrary $ID \in \{0, 1\}^*$ as input, it will return a private key SK. Here, ID is an arbitrary string that will be used as a public key, and SK is the corresponding private decryption key. The extract algorithm extracts a private key from the given public key.
- Encrypt: Taking $params, ID$ and message M as input, where $M \in \mathcal{M}$, it will return a ciphertext $C \in \mathcal{C}$.
- Decrypt: Taking $params, SK$ and $C \in \mathcal{C}$ as input, it will return $M \in \mathcal{M}$.

The above four algorithms must satisfy the standard consistency constraint, that is, when SK is the private key generated by the extract algorithm under the given public key ID, then

$$\forall M \in \mathcal{M} : Decrypt(params, C, SK) = M$$

where

$$C = Encrypt(params, ID, M)$$

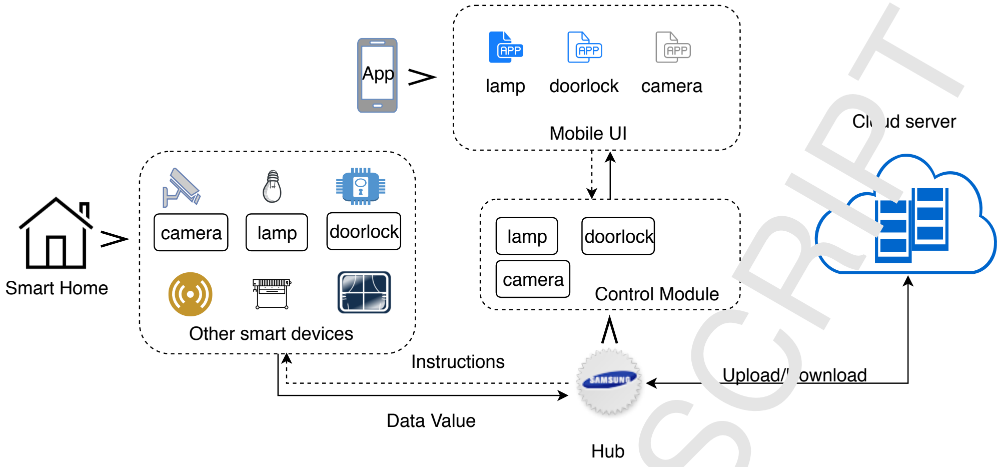


Figure 2: System framework

4. The Smart Home Framework

We formalize the smart home framework in this section. First, we present the system model and the description of a participant in our system model. Second, we provide the security model, including the threat model and security definition.

4.1. System Model

4.1.1. Home area network

Smart home as one of the important applications of IoT, has formed a tiny intelligent world to provide services for people. In this section, we present the IoT-FBAC system model as Figure 2, which consists of different types of smart *devices* (e.g., lamp, door lock, and camera), *a hub*, the *smart apps* on the mobile phone and the *cloud server.*

- **Devices:** A smart home includes various devices such as lamp, door lock and camera. These devices are connected to the home network with the help of the hub. When the devices are working, they generate corresponding data related to the user. Due to the restricted computational power, storage and bandwidth, the complexity operations are not conducted in the devices' side.
- **Hub:** There will be at least one hub in a smart home. The hub is important for users because it acts as a gateway in the smart home, similar to a bridge between devices and app controllers. The hub in smart home has three functions: *aggregation*, *computation* and *transmission*. First, data from devices are aggregated by the hub; second, it helps users to compute the data, we assume that the hub has a stronger higher computational power; third, the hub transmit the data to the cloud server for storage.
- **SmartApp:** Smart apps are installed on the user's mobile phone to provide the interface for users. The user can monitor or command the connected devices through the apps. The apps in the IoT-FBAC scheme are unauthentic because we assume that the attacker may use the third-party app to steal the sensitive data about users.
- **Cloud Server:** It provides storage services for users. Generally, the cloud storage server is provided by a third party, it attempts to obtain information about users when they store data in the cloud. Thus, before uploading the data to the cloud, the hub first encrypts the sensitive data.

Functionality. The basic unit of authorization in our IOT-FBAC scheme is functionality. It consists of two elements: *data value* and *instructions*. Data value represents the data generated by connected devices (e.g., temperature and heart rate). Instruction represents an action command that an app provides that the device will follow (e.g., `open()` and `change_color()`).

4.1.2. Communication model

Due to the diversity of devices, the communication protocols are different. In a general way, there are two main communication protocols in smart home. One is a short-range wireless interface such as IEEE 802.15.4; another one is a long-range wireless interface such as WiFi. The former is used to connect the internal devices and the latter is used to connect the outside world via the Internet.

4.2. Security model

Threat Model. In this paper, we consider untrusted third-party app attacks on IoT platforms. Such apps attempt to access the sensitive data of users or execute unauthorized functionality. The adversary can attack through the following two ways: (1) *malware*, the malicious logic is embedded in the app when it was installed on the phone; (2) *vulnerable apps*, there will be design flaws in these apps that can be utilized by a malicious attacker to escalate the apps' privilege to access the unauthorized functionality and steal sensitive data.

Security Definition. Before defining the security of our system, we present the definition of the secure selective identity IBE scheme. Boneh *et al.* [27] defined the secure selective identity IBE scheme by using a game. Here, we provide a brief overview of this game.

$Game_{\varepsilon,\mathcal{A}}$:

Init: The adversary selects ID^* to be challenged.

Setup: $(params, mk) \leftarrow Setup(1^k)$

The challenger runs this algorithm, provides public parameters *params* to the adversary and keeps the master key *mk* to itself.

Phase 1: The adversary issues queries $q_1, q_2, ..., q_m$, where q_i is the private key query (ID_i). In every query, let $ID_i \neq ID^*$. Then, the challenger runs algorithm *KeyGen* to generate private key d_i corresponding to the public key ID_i. The challenger sends d_i to the adversary.

Note that these queries are adaptive because each query q_i depends on the former queries.

Challenge: Once the adversary decides that Phase 1 is over, it outputs two equal-length plaintexts $M_0, M_1 \in \mathcal{M}$ as the challenged plaintexts. Then, the challenger selects a random bit $b \in \{0, 1\}$ and encrypts it $C = Encrypt(params, ID^*, M_b)$ as the challenge ciphertext. Finally, the challenger sends C to the adversary.

Phase 2: The adversary issues queries $q_{m+1}, q_{m+2}, ..., q_n$, where q_i is the private key query ID_i, $ID_i \neq ID^*$. The challenger responds as in Phase 1.

Note that these queries are adaptive as in Phase 1.

Guess: The adversary outputs a guess $b' \in \{0, 1\}$.

Figure 3: IND-sID-CPA Secure Game

The adversary $\mathcal{A}$ has advantage $Adv_{\varepsilon,\mathcal{A}}$ in attacking the scheme ε, where

$$Adv_{\varepsilon,\mathcal{A}} = |Pr[b = b'] - 1/2|$$

Definition 3.1 *An IBE system ε is (t, q_{ID}, ϵ) IND-sID-CPA secure for any t-time IND-sID-CPA adversary $\mathcal{A}$ that makes at most q_{ID} chosen private key queries that $Adv_{\varepsilon,\mathcal{A}} < \epsilon$.*

As described above, the attacker may select a function ID, which is not authorized by the user. Thus, we define the security as follows:

Definition 3.2 *We say that our IoT-FBAC system ε is (t, q_{ID}, ϵ) IND-sID-CPA secure if the IBE scheme in our system is IND-sID-CPA secure.*

5. IoT-FBAC scheme

We propose a function-based access control system using the IBE scheme, which supports an independent function access of devices.

5.1. The Basic Scheme

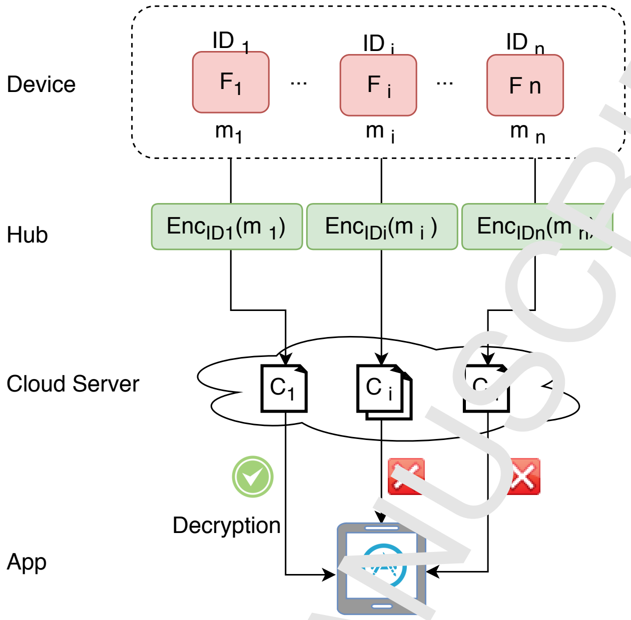


Figure 4: Function-based access control scheme using IBE

In Figure 4, we present an instance showing how the IoT-FBAC scheme works. We suppose that there is a device with n functions in the smart home and that each function has a unique ID. The different functions generate the corresponding data m_i, where $1 \leq i \leq n$. The hub collects all the data and encrypts the data under its ID. After encryption, the ciphertexts C_i will be stored in the cloud server. If an app has the privilege to access one function's data, for example, in Figure 4, the app is authorized with the privilege to access function 1, then it can obtain the C_1 and decrypt the ciphertext exactly. Otherwise, the hub will refuse the request and the app can not obtain anything.

In our IoT-FBAC scheme, we design three stages: *preparation, registration,* and *access.* We now present the basic scheme. Figure 5 shows the process of the proposed IoT-FBAC scheme.

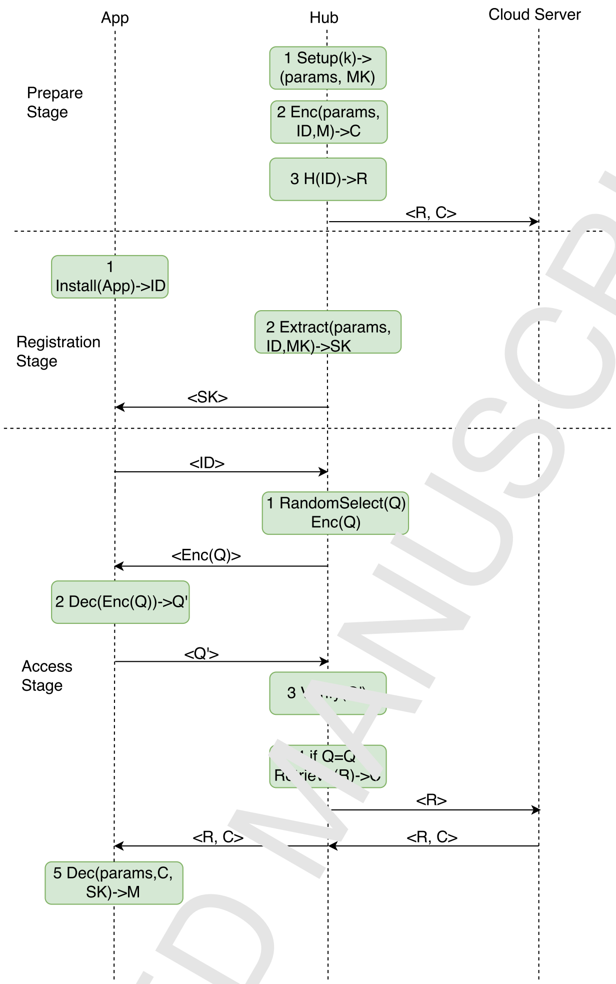


Figure 5: The concrete scheme

A Preparation Stage

This stage aims to prepare some information for the connected devices in the smart home. In the preparation stage, there are three steps: **setup**, **encrypt**, and **ID-hidden**. Algorithm 1 shows the algorithms of the preparation stage.

Algorithm 1 Preparation Stage

1 **Setup**$(k) \to (params, MK)$**.** The hub issues the setup of the IBE scheme, it takes a security parameter k as input, and it outputs the public parameters $params$ and the secret master key MK.

2 **Encrypt** $(params, ID, M) \to C$**:** The hub collects the data M from the connected device in the smart home, and it encrypts them into ciphertexts C under the corresponding ID. It takes $params, ID$ and M as input, and it outputs ciphertext C.

3 **ID-Hidden** $H(ID) \to R$**:** After encrypting the data, the hub chooses a collision-resistant hash function H to compute $H(ID)$, which is used as a tag of a device function. Then, the hub uploads $< C, R >$ to the cloud server.

B Registration Stage

This stage describes the process of installing an app on a user's mobile phone as shown in Algorithm 2. During the registration stage, the user can authorize the privilege of the device function that the app can access. Then, the hub will generate the corresponding private key SK according to the chosen function ID and send the key to the app.

Algorithm 2 Registration Stage

1 **Install** $(App) \rightarrow ID$**:** When an app is installed on the phone, the user can define the smart device function ID that can be accessed.
2 **Extract** $(params, ID, MK) \rightarrow SK$**:** The hub generates the secret key SK according to the function ID selected by the user. It takes $params, ID$ and master key MK as input, and it outputs the SK under the corresponding ID.

C Access Stage

After installing an app, the process of accessing is presented in Algorithm 3. When the hub receives a request from an app, it first verifies the privilege. If the app's access is legal, the hub will ask the cloud server to retrieve the ciphertext and return back to the app. Finally, the app can decrypt it to obtain the data.

Algorithm 3 Access Stage

1 **Encrypt** $(Q) \rightarrow C'$**:** When the hub receives a request ID from an app, it selects a random Q, encrypts Q as a verification ciphertext and sends it to the app.
2 **Decrypt** $(C') \rightarrow Q'$**:** The app decrypts the verification ciphertext to obtain Q' and returns it to the hub.
3 **Verify:** The hub verifies the Q and Q'. If the $Q \neq Q'$, then the hub will refuse this request. Otherwise, it will proceed to the next step.
4 **Retrieve** $(R) \rightarrow C$**:** If $Q = Q'$, then it means that the app has privilege to access this function, and the hub will ask the cloud server to retrieve the ciphertext and return it to the app.
5 **Decrypt** $(params, C, SK) \rightarrow M$**:** The app obtains the data M by running the decryption algorithm. It takes $params$, ciphertext C and private key SK as input, and it outputs the data M.

5.2. The Concrete Scheme

We presented the basic protocol in section 5.1; next, we will construct the concrete IoT-FBAC scheme below.

A Preparation Stage Let k be the security parameter, let $\mathbb{G}$ be a bilinear group of prime order p, and let g be a generator of $\mathbb{G}$. There is an admissible bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$. The public keys (ID) are elements in $\mathbb{Z}_p^*$.

Step 1: $\mathsf{Setup}(k) \rightarrow (params, MK)$. The user runs the setup algorithm through the hub. The hub takes a security parameter k as input, and it obtains $params$ and MK as output. The $params$ are public, and MK is secret. The hub selects random elements $x, y \in \mathbb{Z}_p^*$. Let $X = g^x$ and $Y = g^y$. The public $params$ and the secret master key MK are

$$params = (g, X, Y), MK = (x, y)$$

Step 2: $\mathsf{Encrypt}$ $(params, ID, M) \rightarrow C$. In this step, M denotes the data of the connected devices, and ID denotes the function identity of a device. The hub collects and encrypts the data from the connecting devices. It takes $params, ID$ and M as input, and it returns ciphertext C. To encrypt a data message $M \in \mathbb{G}_1$ under the function $ID \in \mathbb{Z}_p^*$, the hub selects a random $s \in \mathbb{Z}_p^*$ and performs the following calculations:

$$A = g^{s \cdot ID} X^s$$

$$B = Y^s$$

$$D = e(g,g)^s \cdot M$$

Thus, the ciphertext is $C =< A, B, D >$.

Step 3: $H(ID) \to R$. To retrieve the corresponding ciphertext according to the function ID, the hub chooses a collision-resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_p^*$ to compute $H(ID) \to R$, which is used as an index. The hub uploads $< C, R >$ to the cloud server for storage.

B Registration Stage When an app is installed on a user's phone, the user has to define the access privilege of this app. It can access the function's data only if the device function is authorized; otherwise, it cannot succeed.

Step 1: Install $(App) \to ID$. As described in the basic scheme, while installing an app on the smart phone, the user authorizes the device functions ID to the app. For the authorized ID, the app has the privilege to access the data. For non-selective function ID, the app has no privilege to access the data.

Step 2: Extract $(params, ID, MK) \to SK$. After defining the access privilege of the app, the hub will generate the private key SK according to the function ID. It takes $params, ID$ and master key MK as input, and it outputs SK. To generate the private key SK under function $ID \in \mathbb{Z}_p^*$, the hub chooses a random $r \in \mathbb{Z}_p$, which makes $x + ry + ID \neq 0 \pmod{p}$, and computes

$$K = g^{1/(ID+x+ry)} \in \mathbb{G}$$

Thus, the private key is $SK = (r, K)$, the hub sends the private key SK to the app.

C Access Stage

Step 1: Encrypt $(Q) \to C'$. When the hub receives a function ID request from an app, it has to verify whether this request is legal. The hub selects a random $Q \in G_1$ and a random $\tilde{s} \in \mathbb{Z}_p^*$, and then it carries out the following computing:

$$U = g^{\tilde{s} \cdot ID} X^{\tilde{s}}$$

$$V = Y^{\tilde{s}}$$

$$W = e(g,g)^{\tilde{s}} \cdot Q$$

The verification ciphertext is $C' = \{U, V, W\}$, which is sent to the app.

Step 2: Decrypt $(C') \to Q'$. The app decrypts the verification ciphertext with private key SK, the following computing is realized:

$$e(UV^r, K) = e(g,g)^{\tilde{s}}$$

$$\frac{W}{e(g,g)^{\tilde{s}}} = Q'$$

The app sends Q' to the hub.

Step 3: Verify. The hub receives Q' and verifies it. There are two situations: (1) If $Q = Q'$, then it means that the app has privilege to access this function, and the hub will proceed to the next step. (2) If $Q \neq Q'$, then it means that the app attempts over-privilege access. The hub will refuse this request from the app.

Step 4: Retrieve $(R) \to C$. As described above, suppose that ID is the device function that the app wants to access. The app sends an access request for function ID to the hub, and the hub computes $H(ID) \to R$, where R is the index for retrieving the ciphertext in the cloud server. Then, the cloud server returns $< R, C >$ to the hub. Finally, the hub sends ciphertext C to the app.

Step 5: Decrypt $(params, C, SK) \to M$. When the app receives the ciphertext, it can decrypt it. The app runs the decrypt algorithm, which takes the ciphertext C, secret key SK, and $params$ as input, and it will acquire data M. For the detailed decryption process of using the private $SK = (r, K)$ to decrypt the ciphertext $C = (A, B, D)$, we have the following calculation:

$$\frac{D}{e(AB^r, K)} \quad = \quad \frac{e(g,g)^s M}{e(g^{s \cdot ID} X^s Y^{sr}), g^{1/(ID+x+ry)})}$$

$$
\begin{aligned}
&= \frac{e(g,g)^s M}{e(g^{s\cdot ID}\cdot g^{xs}\cdot g^{sry}, g^{1/(ID+x+ry)})} \\
&= \frac{e(g,g)^s M}{e(g^{s(ID+x+ry)}, g^{1/(ID+x+ry))})} \\
&= \frac{e(g,g)^s M}{e(g,g)^s} \\
&= M
\end{aligned}
$$

Thus, the app can obtain the data of the authorized device function.

6. Security Analysis

We analyze the security of the proposed IoT-FBAC scheme in this section. We assume that the cloud is "semi-honest but curious", the hub is a trusted party, and the apps are not trusted. Our scheme aims to resist over-privilege access attacks. Therefore, we analyze how our scheme prevents over-privilege access by malicious apps.

Theorem 5.1: *Suppose that the IBE scheme is secure under the decision q-BDHI assumption; then, the IoT-FBAC scheme is also secure and can prevent over-privilege access.*

Proof. We prove the IoT-FBAC scheme is secure under the decision q-BDHI assumption and it can prevent over-privilege access from the following four lemmas.

Lemma 1. *If the IBE scheme is secure under the decision q-BDHI assumption, then data privacy is protected in our IoT-FBAC scheme.*

Proof. To protect data privacy, we use the IBE scheme to support our security model. From the security definition in section III, the IBE scheme is IND-sID-CPA secure under the decision q-BDHI assumption. The detailed proof is presented in [27]. In the IoT-FBAC scheme, we use the IBE scheme to encrypt the data of devices through the hub and send the encrypted data to the cloud server. Thus, an attacker cannot acquire sensitive information from the encrypted data. Data privacy is protected in the IoT-FBAC.

Lemma 2. *If the app follows the protocol of our IoT-FBAC scheme, then it can only obtain the private key of an authorized function ID.*

Proof. In the IoT-FBAC scheme, there are three stages: preparation stage, registration stage and access stage. Following our protocol, the hub will preprocess the data from the connected IoT devices, encrypt the data under the different function IDs and upload the encrypted data to the cloud server in the preparation stage. When an app is installed on a smart phone, the user will authorize the app privilege to access the designated function. At the same time, the hub extracts the corresponding private key according to the designated function ID. Then, the hub sends the private key to the app. In this way, we can see that the app can only obtain the private key of an authorized function ID.

Lemma 3. *If the malicious app attempts to obtain the data of function ID^*, then it cannot over-privilege access it.*

Proof. We suppose that for a device with two functions $\{ID, ID^*\}$, function ID is authorized to an app while function ID^* is not. Following our scheme, the hub will extract the private key SK under the function ID. Now, the app attempts to access the data of function ID^*, which is not authorized. The app will send a request to the hub, and the hub will select a random number Q and encrypt it with ID^*. The computation is as follows:

$$A^* = g^{s'\cdot ID^*}\cdot g^{xs'}$$

$$B^* = g^{ys'}$$

$$D^* = e(g,g)^{s'}\cdot Q$$

where s' is selected randomly in $\mathbb{Z}_p^*$. Then, the hub sends $C^* = (A^*, B^*, D^*)$ to the app. When the app receives C^*, it decrypts this by using the private key SK. We can see that $SK = (r, g^{1/(ID+x+ry)})$, where

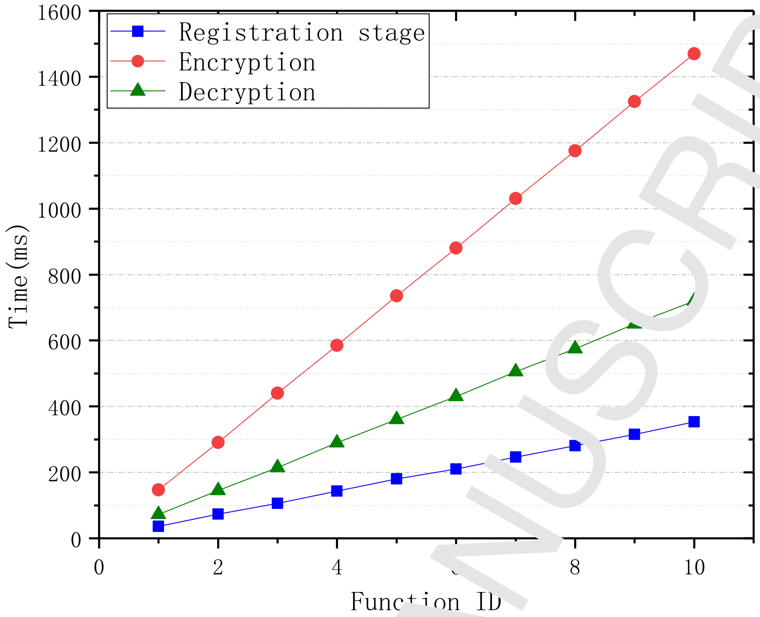


Figure 6: The time increase with the number of function ID.

r is a random element in $\mathbb{Z}_p$ and x, y are random elements in $\mathbb{Z}_p^*$. The decryption process is

$$
\begin{aligned}
\frac{D^*}{e(A^*B^{*r}, K)} &= \frac{e(g,g)^{s'}Q}{e(g^{s'\cdot ID^*}\cdot g^{xs'}\cdot g^{ys'r}, g^{1/(ID+x+ry)})} \\
&= \frac{e(g,g)^{s'}Q}{e(g^{s'(ID^*+x+ry)}, g^{1/(ID+x+ry)})} \\
&= \perp
\end{aligned}
$$

From the above decryption, the pairing computation in the denominator cannot be computed because (x, y) is the MSK kept by the hub, and the app does not know anything about it. Thus, the app cannot access the unauthorized function. Our scheme prevents over-privilege access.

Lemma 4. *If other attackers attempt to obtain data about connected devices, they cannot obtain any sensitive information about the IoT device function IDs.*

Proof. As described in step 3 of the preparation stage in our scheme, we choose a collision-resistant hash function H to hide the device function ID, which is $H(ID) \to R$. According to the security of the hash function, it is difficult to compute $H^{-1}(R) \to ID$. Moreover, for the given ID, it is also difficult to find ID^* to make $H(ID) = H(ID^*)$. Thus, other attackers cannot obtain any information about the IoT device function IDs.

The above analysis indicates that our IoT-FBAC scheme is secure and that it can prevent over-privilege access attacks.

7. Experimental Evaluation

Because IBE is the backbone of the IoT-FBAC scheme, we implemented algorithms of the developed scheme in Java based on the Java Pairing-based Cryptography (JPBC) library. All the algorithms were run on a PC with a 2.13 GHz CPU and 6 GB of RAM. The length of a message was 128 bytes.

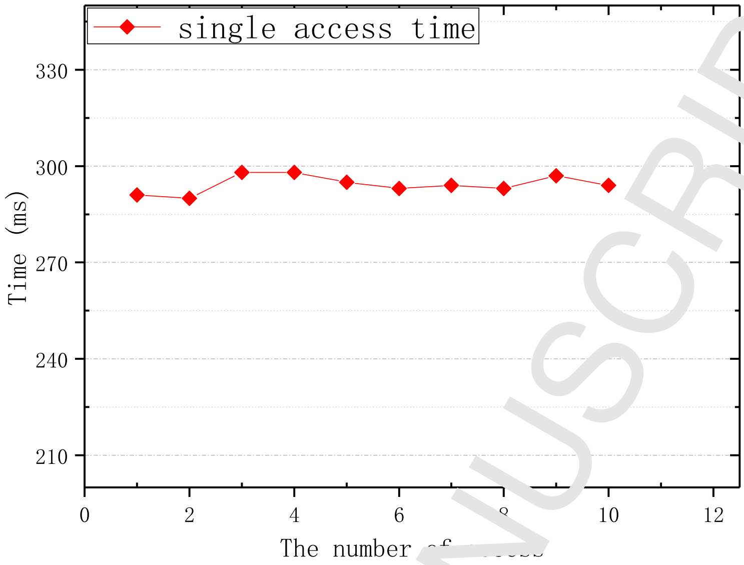


Figure 7: The time of accessing one function ID.

7.1. Comparison Analysis

In Table 1, we present the comparison between the FACT scheme and the IoT-FBAC scheme from six aspects: *object unit, fine-grained access, against over-privilege, ACL, data security* and *space saving.*

Table 1: The comparison between the FACT scheme and the IoT-FBAC scheme.

Framework	Object Unit	Fine-grained Access	Against Over-privilege	ACL	Data Security	Space Saving
FACT	Functionality	Yes	Yes	Require	No	No
IoT-FBAC	Functionality	Yes	Yes	Not Require	Yes	Yes

Both FACT and our scheme achieve fine-grained access control. The unit of access control that both schemes authorize is functionality. In FACT, Linux Containers (LXC) [28] are used to insulate the functionality. The access permission of apps is listed in an ACL, which is stored in the FACT system. If an app sends a request to FACT, it will look up the ACL to check whether the request is valid. In our scheme, an ACL table is not required. The hub serves this role when it handles the data and encrypts them under different IDs. The app with the secret key only accesses the corresponding functionality. Thus, the two schemes can prevent over-privilege attacks.

Additionally, to protect the sensitive data from the devices, we encrypt them and outsource the ciphertexts to the cloud server for storage in the IoT-FBAC scheme. It allows the proposed scheme to achieve data security while saving the local storage space.

7.2. Performance Evaluation

We test the time of all algorithms in the proposed IoT-FBAC scheme. Figure 6 shows the time of registration, encryption and decryption. During registration stage (the blue line), an app is installed on a smart phone, and the user authorizes the privilege to the app, which means that the user defines which functionality the app can access. If chosen, the hub will generate the corresponding secret keys for the app. The time increases with the number of function IDs. In practice, the number of functionalities is limited;

thus, the time will also be restricted. The red line illustrates the encryption time. The encryption operation is executed by the hub. The data generated by devices are transmitted to the hub, and the hub encrypts them under different function IDs. The greater the number of function IDs is, the more time that it will cost. When devices in a smart home are no longer added, the encryption time will achieve a balance. The green line shows the decryption time. It is operated by an app. When an app receives the ciphertext, it decrypts it with the secret key to obtain the data. The decryption time increases with the number of function IDs.

Figure 7 shows the time for an app to execute one access to a device function. We assume that there is only one functionality to request in each access process. The time of each access is a constant. Compared to the FACT scheme, the time is improved in our IoT-FBAC scheme because if an ACL contains N items, the search time complexity is $\lg N$ using the fastest search algorithm, whereas in our IoT-FBAC scheme, the hub does not require the ACL and outsources the search process to the server.

8. Related Work

Thus far, most works about IoT security have focused on three aspects: *authentication*, *data privacy*, and *access control*.

8.1. Authentication

Authentication is necessary when a new device connects to the Internet. Kothmayr *et al.* [29] introduced the first implemented two-way authentication security scheme for IoT devices, which used RSA cryptography. Their scheme is based on the most widely used protocol Datagram Transport Layer Security (TDLS) and suitable for the IoT platform. Zhang *et al.* [30] proposed a novel authentication scheme for IoT devices which is based on the proximity. Their scheme requires users to hold a smartphone and perform one of two hand gestures in front of devices. Meanwhile, it limits the distance between the user and the device. Kumar *et al.* [31] proposed an anonymous secure framework (ASF) that provides the authentication more efficient and unlinkability. Their scheme requires update session keys regularly, so when the number of devices is large enough, the update operations need consume much time. In [32], Xi *et al.* introduced a fast and error-free authentication and key agreement protocol which used channel state information (CSI) as the common secret key for the privileged devices. Similarly, their scheme also is impacted by the distance. Miettinen *et al.* [33] presented an approach for secure zero-interaction pairing for personal devices that is based on the context. This method requires no user involvement and devices can be paired automatically. Jian *et al.* [34] presented a cloud-aided lightweight certificateless authentication protocol with anonymity. Other device pairing works were presented in [35, 36, 37].

8.2. Data Privacy

There are also many works studying data privacy in IoT applications or frameworks. Fernandes *et al.* [38] presented a system called FlowFence that requires users to work with their sensitive data in sandboxes to protect the privacy. When a sensitive data flow in the sandbox, it will block all other undeclare flows. Hails [39] presented a web framework that uses MAC to confine untrusted apps. Though this method can constraints on apps' behaviour, the data privacy of users are transparent to the untrusted apps. Jana *et al.* [40] designed a practical privacy protection system for the context-aware application scenario in which an untrusted app is running on a trusted device. It integrates the OpenCV to realize the visual inputs. Yu *et al.* [41] proposed decentralized middle boxes to prevent unapproved communication between IoT devices, while Simpson *et al.* [42] proposed a central security manager to control the traffic of devices, which aims to reduce the security risk. This central security manager can intercept all the traffic to and from devices, detect the status of the devices and report vulnerabilities.

8.3. Access Control

The most common access control system in IoT is based on the ACL [43], in which access rights are centrally specified. However, with the increasing number of IoT devices, it is a burden to manage the access control of these devices. Ardagna *et al.* [44] proposed a credential-based access control system that uses a combination of various attributes concerning a requested subject to generate dynamic access policies. Some works proposed capability-based access control models, such as those proposed in [45, 46, 47]. But none of these schemes took into account the over-privilege access behaviour, which can lead to the sensitive data leak. Anggorojati *et al.* presented a vision-based capability to handle authority delegation in cross-domain IoT environments [45]. Hernndez-Ramos *et al.* provided a model in a distributed IoT environment where capability issuance and authorization occur without intermediate entities implementing access control logic [46]. Gusmeroli *et al.* designed a proposal in which users can manage access control processes for their own IoT devices by generating electronic capability tokens [47]. Some other related works are presented in [48, 49, 50].

9. Conclusion and Future Work

In order to prevent over-privilege access, we propose the Function-based Access Control scheme in IoT. With using the identity-based encryption scheme, the proposed scheme can obtain fine-grained access control because the basic unit object is functionality. Meanwhile, data from devices are encrypted before being uploaded to the server cloud, which guarantees data privacy. The experimental results indicate that the IoT-FBAC scheme is practical and efficient. In the future work, the interesting directions include studying the dynamic access control scheme to prevent over-privilege access and designing a secure efficient solution to verify the validity of the connected devices in IoT scenario.

Acknowledgments

This work is supported by National Key Basic Research Development Program (No.2013CB834204), National Natural Science Foundation of China (No.61772291), Tianjin Natural Science Foundation (No.17JCZD JC30500), and Open Topic of China Civil Aviation University Information Security Evaluation Center (No.CAAC-ISECCA-201702).

References

[1] V. Turner, J. F. Gantz, D. Reinsel, S. Minton, The digital universe of opportunities: Rich data and the increasing value of the internet of things, IDC Analyze the Future (2014) 1–5.

[2] C. Neagle, A guide to the confusing internet of things standards world, Network World, (2014).

[3] Z. Liu, Z. Wu, T. Li, J. Li, C. Shen, Gmm and cnn hybrid method for short utterance speaker recognition, IEEE Transactions on Industrial Informatics 14 (7) (2018) 3244–3252.

[4] Y. Li, G. Wang, L. Nie, Q. Wang, W. Tan, Distance metric optimization driven convolutional neural network for age invariant face recognition, Pattern Recognition 75 (2017) 51–62.

[5] M. Z. A. Bhuiyan, J. Wu, G. Wang, J. Cao, Sensing and decision making in cyber-physical systems: The case of structural event monitoring, IEEE Transactions on Industrial Informatics 12 (6) (2016) 2103–2114.

[6] M. Z. A. Bhuiyan, J. Wu, G. Wang, Z. Chen, J. Chen, T. Wang, Quality-guaranteed event-sensitive data collection and monitoring in vibration sensor networks, IEEE Transactions on Industrial Informatics 13 (2) (2017) 572–583.

[7] Apple homekit, https://developer.apple.com/homekit/.

[8] Samsung smartthings, https://www.smartthings.com.

[9] Google weave project, https://developers.google.com/weave/.

[10] N. Dhanjani, Abusing the internet of things: Blackouts, freakouts, and stakeouts, https://www.abbeys.com.au/book/abusing-the-internet-of-things-blackouts-freakouts-and-stakeouts.do.

[11] L. Fan, X. Lei, N. Yang, T. Q. Duong, G. K. Karagiannidis, Secure multiple amplify-and-forward relaying with cochannel interference, IEEE Journal of Selected Topics in Signal Processing 10 (8) (2016) 1494–1505.

[12] L. Fan, X. Lei, N. Yang, T. Q. Duong, G. K. Karagiannidis, Secrecy cooperative networks with outdated relay selection over correlated fading channels, IEEE Transactions on Vehicular Technology 66 (8) (2017) 7599–7603.

[13] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-An, H. Ye, Significant permission identification for machine learning based android malware detection, IEEE Transactions on Industrial Informatics 14 (7) (2018) 3216–3225.

[14] Z. Huang, J. Lai, W. Chen, T. Li, Y. Xiang, Data security against receiver corruptions: Soa security for receivers from simulatable dems, Information Science (2018)doi:10.1016/j.ins.2018.08.059.

[15] C. Gao, S. Lv, Y. Wei, Z. Wang, Z. Liu, X. Cheng, M-sse: An effective searchable symmetric encryption with enhanced security for mobile devices, IEEE Access 6 (2018) 38860–38869.

[16] K. Riad, L. Ke, Roughdroid: Operative scheme for functional android malware detection, Security and Communication Networks (2018)doi:10.1155/2018/80873032018.

[17] B. Krebs, Hacked cameras, dvrs powered todays massive internet outage, https://krebsonsecurity.com/2016/10/hacked-cameras-dvrs-powered-todays-massive-internet-outage/.

[18] T. Denning, T. Kohno, H. M. Levy, Computer security and the modern home, Communications of the ACM 56 (1) (2013) 94–103.

[19] J. Li, Z. Liu, X. Chen, F. Xhafa, X. Tan, D. S. Wong, L-encdb: A lightweight framework for privacy-preserving data queries in cloud computing, Knowledge-Based Systems 79 (2015) 18–26.

[20] Z. Liu, Y. Huang, J. Li, X. Cheng, C. Shen, Divoram: Towards a practical oblivious ram with variable block size, Information Sciences 447 (2018) 1–11.

[21] T. Li, J. Li, Z. Liu, P. Li, C. Jia, Differentially private naive bayes learning over multiple data sources, Information Sciences 444 (2018) 89–104.

[22] P. H. Chia, Y. Yamamoto, N. Asokan, Is this app safe?: a large scale study on application permissions and risk signals, Proceedings of the 21st international conference on World Wide Web (2012) 311–320.

[23] A. P. Felt, E. Chin, S. Hanna, D. Song, D. Wagner, Android permissions demystified, Proceedings of the 18th ACM conference on Computer and communications security (2011) 627–638.

[24] E. Fernandes, J. Jung, A. Prakash, Security analysis of emerging smart home applications, Security & Privacy (SP) (2016) 636–654.

[25] S. Lee, J. Choi, J. Kim, B. Cho, S. Lee, H. Kim, J. Kim, Fact: Functionality-centric access control system for iot programming frameworks, Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies (2017) 43–54.

[26] Iotivity, https://iotivity.org/.

[27] D. Boneh, X. Boyen, Efficient selective-id secure identity-based encryption without random oracles, International Conference on the Theory and Applications of Cryptographic Techniques (2004) 223–238.

[28] M. Helsley, Lxc: Linux container tools, IBM devloperWorks Technical Library 11.

[29] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, G. Carle, A dtls based end-to-end security architecture for the internet of things with two-way authentication, Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference on (2012) 956–963.

[30] J. Zhang, Z. Wang, Z. Yang, Q. Zhang, Proximity based iot device authentication, INFOCOM 2017-IEEE Conference on Computer Communications, IEEE (2017) 1–9.

[31] P. Kumar, A. Braeken, A. Gurtov, J. Iinatti, P. H. Ha, Anonymous secure framework in connected smart home environments, IEEE Transactions on Information Forensics and Security 12 (4) (2017) 968–979.

[32] W. Xi, C. Qian, J. Han, K. Zhao, S. Zhong, X.-Y. Li, J. Zhao, Instant and robust authentication and key agreement among mobile devices, Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (2016) 616–627.

[33] M. Miettinen, N. Asokan, T. D. Nguyen, A.-R. Sadeghi, M. Sobhani, Context-based zero-interaction pairing and key evolution for advanced personal devices, Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (2014) 880–891.

[34] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, Y. Tang, Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks, Journal of Network & Computer Applications 106 (2018) 117–123.

[35] H. Chan, A. Perrig, D. Song, Random key predistribution schemes for sensor networks, Security & Privacy (2003) 197–213.

[36] L. Eschenauer, V. D. Gligor, A key-management scheme for distributed sensor networks, Proceedings of the 9th ACM Conference on Computer and Communications Security (2002) 41–47.

[37] D. Liu, P. Ning, R. Li, Establishing pairwise keys in distributed sensor networks, ACM Transactions on Information and System Security (TISSEC) 8 (1) (2005) 41–77.

[38] E. Fernandes, J. Paupore, A. Rahmati, D. Simionato, M. Conti, A. Prakash, Flowfence: Practical data protection for emerging iot application frameworks., USENIX Security Symposium (2016) 531–548.

[39] D. B. Giffin, A. Levy, D. Stefan, D. Terei, D. Mazières, J. C. Mitchell, A. Russo, Hails: Protecting data privacy in untrusted web applications, Usenix Conference on Operating Systems Design and Implementation (2012) 47–60.

[40] S. Jana, A. Narayanan, V. Shmatikov, A scanner darkly: Protecting user privacy from perceptual applications, Security & Privacy (SP) (2013) 349–363.

[41] T. Yu, V. Sekar, S. Seshan, Y. Agarwal, C. Xu, Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things, Proceedings of the 14th ACM Workshop on Hot Topics in Networks (2015) 5.

[42] A. K. Simpson, F. Roesner, T. Kohno, Securing vulnerable home iot devices with an in-hub security manager, Pervasive Computing and Communications Workshops (PerCom Workshops) (2017) 551–556.

[43] J. Qian, S. Hinrichs, K. Nahrstedt, Acla: A framework for access control list (acl) analysis and optimization, Communications and Multimedia Security Issues of the New Century (2001) 197–211.

[44] C. A. Ardagna, S. D. C. di Vimercati, G. Neven, S. Paraboschi, F.-S. Preiss, P. Samarati, M. Verdicchio, Enabling privacy-preserving credential-based access control with xacml and saml, Computer and Information Technology (CIT) (2010) 1090–1095.

[45] B. Anggorojati, P. N. Mahalle, N. R. Prasad, R. Prasad, Capability-based access control delegation model on the federated iot network, Wireless Personal Multimedia Communications (WPMC) (2012) 604–608.

[46] J. L. Hernandez-Ramos, A. J. Jara, L. Marin, A. F. Skarmeta, Distributed capability-based access control for the internet of things, Journal of Internet Services and Information Security (JISIS) 3 (3/4) (2013) 1–16.

[47] S. Gusmeroli, S. Piccione, D. Rotondi, A capability-based security approach to manage access control in the internet of things, Mathematical and Computer Modelling 58 (5-6) (2013) 1189–1205.

[48] D. He, N. Kumar, H. Wang, L. Wang, K. K. R. Choo, A. Vinel, A provably-secure cross-domain handshake scheme with

symptoms-matching for mobile healthcare social network, IEEE Transactions on Dependable & Secure Computing 15 (4) (2018) 633–645.

[49] D. He, S. Zeadally, L. Wu, Certificateless public auditing scheme for cloud-assisted wireless body area networks, IEEE Systems Journal 12 (1) (2018) 64–73.

[50] D. He, N. Kumar, H. Shen, J. H. Lee, One-to-many authentication for access control in mobile pay-tv systems, Science China Information Sciences 59 (5) (2016) 1–15.

Hongyang Yan received her M.S. degrees from School of Mathematics and Information Science, Guangzhou University in 2016. Now she is a Ph.D. student in Nankai University. Her research interests include secure access control such as attribute-based cryptography and identity-based cryptography, IoT secure.

Yu Wang received his PhD in computer science from Deakin University, Australia. He is currently an associate professor with the School of Computer Science, Guangzhou University, China. His research interests include network traffic analysis, mobile networks, social networks, and cyber security.

Chunfu Jia is a professor and a PhD supervisor. His main research interests include network and system security, cryptography application and malware analysis.

Jin Li received the B.S. degree in mathematics from Southwest University, Chongqing, China, in 2002, the M.S. degree in mathematics from Sun Yat-sen University, Guangzhou, China, in 2004, and the Ph.D. degree in information security from Sun Yat-sen University, in 2007. He served as a Senior Research Associate with the Korea Advanced Institute of Technology, Daejeon, Korea, and the Illinois Institute of Technology, Chicago, IL, USA, from 2008 to 2010. He is currently a Professor with Guangzhou University, Guangzhou, China. He has authored over 40 papers in international conferences and journals. His research interests include design of secure protocols in cloud computing (secure cloud storage, encrypted keyword search, and outsourcing computation) and cryptographic protocols.

Yang Xiang received his PhD in Computer Science from Deakin University, Australia. He is currently a full professor at School of Information Technology, Deakin University. He is the Director of the Network Security and Computing Lab (NSCLab) and the Associate Head of School (Industry Engagement). His research interests include network and system security, distributed systems, and networking. In particular, he is currently leading his team developing active defense systems against large-scale distributed network attacks.

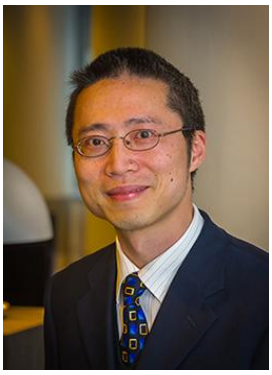
Witold Pedrycz is a Professor and Canada Research Chair (CRC - Computational Intelligence) in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada. He is also with the Systems Research Institute of the Polish Academy of Sciences, Warsaw, Poland. In 2009 Dr. Pedrycz was elected a foreign member of the Polish Academy of Sciences. His main research directions involve Computational Intelligence, fuzzy modeling and Granular Computing, knowledge discovery and data mining, fuzzy control, pattern recognition, knowledge-based neural networks, relational computing, and Software Engineering.

Hongyang Yan

Yu Wang

Chunfu Jia

Jin Li



Yang Xiang

Witold Pedrycz

Highlights

- We provide a function-based access control system for smart homes
- Our scheme prevents the app from accessing unauthorized functions.
- Each access operation in our scheme costs a constant time.
- Data privacy in our scheme is secure.