SecHome: A Secure Large-Scale Smart Home System Using Hierarchical Identity Based Encryption

Yu Li[1], Yazhe Wang[1(✉)], and Yuan Zhang[2]

[1] State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences,
Beijing 100093, China
{liyu,wangyazhe}@iie.ac.cn
[2] State Key Laboratory for Novel Software Technology,
Computer Science and Technology Department, Nanjing University,
Nanjing 210023, China
zhangyuan05@gmail.com

Abstract. With the rapid development of Cyber-Physical Systems, there has been a growing trend among smart devices to connect networks via different wireless protocols. In particular, smart home devices are becoming more and more prevalent. However, security issues on how to control and prevent unauthorized access to smart devices connected to the cloud still need to be considered and solved. Hierarchical Identity-Based Encryption is a well-known access control model which enables parent nodes to decrypt the data from descendant nodes. In this paper, we present SecHome, a large-scale smart home system using hierarchical identity based encryption protocol. SecHome applies the protocol by using efficient pairing based cryptography to enforce an access control policy, so parent nodes at the top of the hierarchy can monitor their descendant nodes. In practice, we have implemented our SecHome system on both smart phone and smart device sides, and the final evaluations demonstrate that our system is proved to be of practicality and with high efficiency.

Keywords: Smart device · Security · Privacy
Hierarchical Identity-Based Encryption

1 Introduction

In recent years, smart home devices have received much attention due to their potential applications and the proliferation of Internet of Things. As a result, users who would like to set up different access controls to different people and devices are driven to use an access manager. Some access managers are provided by smart home vendors as part of smart home ecosystem, and some are provided by third-party cloud services. There are several different types of

S. Qing et al. (Eds.): ICICS 2017, LNCS 10631, pp. 339–351, 2018.
https://doi.org/10.1007/978-3-319-89500-0_30

Internet of Things access methods with a smart home system, which include IEEE 802.11(Wi-Fi), Bluetooth and ZigBee. All these technologies are helping smart devices connecting to a cloud center. The owner of the devices can utilize her smart phone to control and monitor them. Most of them are wireless-network based where passwords are backed up to cloud and synced across the smart home devices.

However, numerous recent surveys show that smart home systems are vulnerable to hacking because of the weak authentication and authorization. The security and privacy issues are highly concerned for smart home adoption since the data generated by living environment is usually sensitive [1]. So far, we can identify two types of issues for smart home system: security issue and privacy issue. By saying security issue, we refer to the broad class of adversaries that intentionally attack the system. The problems of security issue need to be addressed, for example, using authentication and encryption scheme to avoid interference over the communication channel. Although the authentication between users and cloud is important in previous home automation researches, the focus of this paper is on privacy, which we expect will be the dominant concern of users in a smart home system. Specifically, privacy issue is concerned by home owners who are afraid that cloud service providers will get sensitive information and affect the security of the living environment. For example, sensitive information like smart lock will indicate whether there are any people at home. A cloud service provider will need a secure protocol to guarantee users' private information.

Current existing smart home systems use Transport Layer Security (TLS) or HTTPS protocol to provide authentication as well as encryption. However, such smart home systems may have several issues. First, there is no guarantee of third party security and privacy protection. The cloud server of smart home systems would master all sensitive data and it is difficult to say that the data won't harm users' privacy. Second, the smart home owner cannot issue keys to her home members dynamically so that they must interact with the cloud server which may have many potential dangers. Third, compared with the traditional security system, smart devices are more vulnerable since the low-cost embedded systems are used. Therefore, we cannot just rely on the traditional security system to provide a strong security guarantee in Cyber-Physical Systems, especially in smart home system.

In this paper, we propose a secure large-scale smart home system using hierarchical identity based encryption. Generally speaking, the basic scheme of our system can be described as follows. When a home owner starts setting a smart home, she issues a secret key to the home members based on the hierarchy of the home. Later, when any home member buys a smart device, the owner issues a private key for it and it connects to the presetting private cloud. The private cloud then communicates with the public cloud by using the public ID to encrypt the sensitive data. In order to allow users to control the access to the smart home devices, suitable hierarchy and authentication as well as encryption are required.

There are two main issues in our system design that need to be addressed. First, since a malicious user may attempt to impersonate a normal user and control her smart devices, every device needs to be authenticated by smart home cloud server to ensure that device qualifies for connecting. Second, different home members should have different access rights to monitor and control smart devices. For instance, the smart home owner can control and issue access rights to her home members.
Our contributions can be summarized as follows:

- We are the first to study privacy protection in smart home system by using hybrid cloud architecture and to propose a hierarchical key management scheme.
- To the best of our knowledge, we are also the first to study a privacy protection in smart home using Hierarchical Identity Based Encryption.
- In terms of privacy, our algorithm leaks no knowledge about each party's data.

The rest of this paper is organized as follows. In Sect. 2, we discuss the related works. In Sect. 3, we present the overall architecture and intuitions behind our design. We then give the full specification of our system and analyze the security in Sect. 4. In Sect. 5, we present evaluations of our solution. Our conclusion and future work are shown in Sect. 6.

2 Related Work

According to [2,3], this novel paradigm, named "Internet of Things", is rapidly gaining ground as the modern wireless networks technology. However, in [4], the IoT has a great impact on personal privacy and security. Therefore, a high degree of reliability is needed which includes data authentication, access control and clients' privacy. Unlike other Cyber-Physical Systems devices, smart home devices have a direct influence on people's daily life. Therefore, the design of the access control scheme in smart home system is extremely important and the encrypted data generated by the system should only be viewed by the correct home member. To the best of our knowledge, there is no such scheme that can only let smart home owner monitor the data. Each of the existing smart home systems relies on a reliable third party cloud server.

Identity-Based Encryption(IBE) which was proposed by Shamir in [5] can simplify key management in a Public Key Infrastructure(PKI) by using objects' identities(e.g., unique mobile phone number, email address, product serial number, etc.) as public keys. After that the first secure IBE scheme was proposed by Boneh and Franklin [6] from the bilinear pairings. They have also proved that the IBE scheme is semantically secure against adaptive chosen-ciphertext attack under the DBDH assumption in the random oracle model. Moreover, a handful of researches on constructing provable secure IBE scheme were proposed

in [7–10]. Many alternative approaches are derived from IBE with the development of cloud computing, for example, Role Based Encryption (RBE) [11] and Attribute Based Encryption (ABE)[12–16]. Another approach which can enforce access control policies and data encryption is to apply the Hierarchical ID-based Encryption (HIBE)[17–19] to Internet of Things.

The design of our secure smart home system is motivated by [20], which is an emerging active research area in the intersection of computer security and Internet of Things. In HIBE, the length of the identity becomes longer with the growth in the depth of hierarchy, which is suitable for a home's structure since the depth of the hierarchy would not be too large in a family. However, as far as we know, there is no previous proposed security and privacy protection scheme for smart home systems. Furthermore, although our design is motivated by HIBE proposed in [20], our problem does not fit exactly into the specific cryptographic-design. For example, in our system, we consider various hardware securities, such as secure boot and data isolation [21].

3 Architecture

In this section, we present the architecture of our secure hierarchical smart home system. In our system, there are five components which include public cloud, private cloud, home owner, home members and smart devices. The architecture is shown in Fig. 1. Public cloud is utilized to store and transfer the encrypted sensitive data. Due to the limited computation capacity, a private cloud is used to help encrypting and decrypting data generated by the smart device sensors. Home owner and home members use a smart phone to control and monitor data generated by the smart devices. Home owner, home members and smart devices are arranged in a form of a hierarchy which can be visualized as a pyramid. The hierarchy will be described in Sect. 5.2.

In our secure hierarchical smart home architecture, we assume that a private cloud has been set up in the home. Then we let the owner of the home be at the depth 1 in the home hierarchy. After that, the home owner can set keys for her children nodes as well as the sub-structure as shown in steps 1 and 2. Home members also set keys for their smart devices in steps 3 and 4. Smart devices

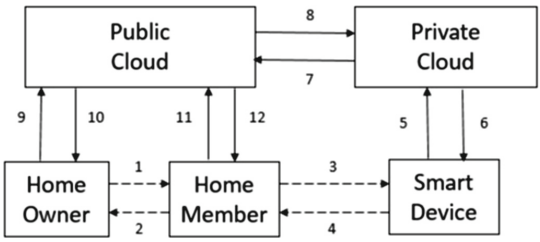


Fig. 1. Our HIBE system architecture

transfer the encrypted data to private cloud using symmetric key encryption as shown in steps 5 and 6. The private cloud can only be accessed by the home owner and the data must be encrypted by HIBE encryption when they leave the private cloud as shown in step 7 and 8. A public cloud is a third-party cloud which is employed to store and transfer the encrypted data. Home owner and home members who equip with a smart phone can send HIBE encrypted data and command public cloud in step 9–12. Therefore, the top of the home hierarchy can monitor the huge data generated by their descendant nodes and smart devices through the public cloud. The step details of how to set up keys and encrypt data for home members and smart devices will be described in Sect. 5.2.

4 SecHome Using HIBE: Specification

4.1 Overview of Our System

In this section, we present the overview of our system and the intuitions behind our design. Boneh et al. proposed a hierarchical identity based encryption with constant size ciphertext in [20] which can apply to a number of applications. However, since the encryption scheme they proposed is only a general approach for constructing HIBE using pairing based encryption, we will apply the HIBE that matches the properties of smart home system. Moreover, we also provide a revocation mechanism to keep our system more secure. At a high level, smart home device usually contains an embedded chip with a real time operating system running on it. But this kind of systems is more vulnerable because of the design of the system architecture [22,23]. Therefore, our system also considers other aspects of security that would protect the data stored in the device. Intuitively, we provide a hardware-assisted dynamic root of trust which allows secure task loading at the runtime.

4.2 Design of Our SecHome System

Our goal is to maintain the confidentiality and integrity of users' data running on a network of hosts potentially under the control of an adversary. This section outlines our design to achieve this with good performance and through keeping adversary's attack out of the SecHome system.

Algorithm 1 shows the overall scheme of our smart home system, which consists of the initialization, adds and revokes nodes in the hierarchy. Specifically, people nodes in this hierarchy are equipped with a mobile device that allows them to generate keys for children nodes and to send as well as receive messages through a wireless network.

Algorithm 1. Algorithm for Smart Home Keys Generation and Encryption

Initialization:

For each home owner who joins in the smart home system, the owner generates public parameters $(g, g_1, g_2, g_3, h_1, h_2, \cdots h_L)$ and master key $mk = g_2^{\alpha}$.

Input:

For each member and smart device in depth 1, we denote it as ID_1. Home owner randomly chooses r, generates $d_{ID_1} = (g_2^{\alpha} \cdot (h_1^{H(ID_1)} \cdot g_3)^r, g^r, h_2^r, \cdots h_L^r)$ and sends it via a secret channel described in Algorithm 2.

Add an element in the hierarchy:

When a new smart device joins in this hierarchy at depth 2,

ID_1 randomly chooses t and generates $d_{H(ID_2)} = (a_0 \cdot b_2^{H(ID_2)} \cdot (h_1^{H(ID_1)} \cdots h_2^{H(ID_2)} \cdot g_3)^t, a_1 \cdot g^t, b_3 \cdot h_3^t \cdot b_L \cdot h_L^t)$ and sends it via a secret channel described in Algorithm 2.

ID_1 encrypts t using ID_0's ID and sends it to ID_0. ID_0 gets $r + t$ which can decrypt device in depth 2.

When a new home member joins in this hierarchy in depth 1 and controls a device in depth 2,

Home owner only needs to generate a d_{ID_1} for the member and sends the encrypted device's r to him via a secret channel using Algorithm 2. The member can use device's r to control or get the data from it.

Recursively we can set up keys for all nodes in the hierarchy.

Update an element in the hierarchy:

When a node in this hierarchy at depth 2 needs to update decryption key,

The parent ID_1 randomly chooses a new t and generates $d_{H(ID_2)} = (a_0 \cdot b_2^{H(ID_2)} \cdot (h_1^{H(ID_1)} \cdots h_2^{H(ID_2)} \cdot g_3)^t, a_1 \cdot g^t, b_3 \cdot h_3^t \cdot b_L \cdot h_L^t)$ and sends it via a secret channel described in Algorithm 2.

ID_1 encrypts t using ID_0's ID and sends it to ID_0. ID_0 gets $r + t$ which can decrypt device in depth 2.

We can update nodes'keys in the list by recursively using $RL_{ID|k}$ and $d_{ID|k-1}$.

Revoke an element in the hierarchy:

When a node in this hierarchy at depth 2 needs to be revoked,

The parent ID_1 randomly chooses t and generates $d_{H(ID_2)} = (a_0 \cdot b_2^{H(ID_2)} \cdot (h_1^{H(ID_1)} \cdots h_2^{H(ID_2)} \cdot g_3)^t, a_1 \cdot g^t, b_3 \cdot h_3^t \cdot b_L \cdot h_L^t)$, and sends it via a secret channel described in Algorithm 2.

ID_1 encrypts t using ID_0's ID and sends it to ID_0. ID_0 gets $r + t$ which can decrypt device in depth 2.

The node will have been successfully revoked by recursively doing this until all sub-nodes getting new keys.

When a home owner generates a key for her children nodes, for example, we use a general HIBE structure in Fig. 2 to illustrate our scheme and explain how the proposed scheme adds and revokes a home member or device in the hierarchy. Start by assuming a home owner's id is Pid_1. The owner has two

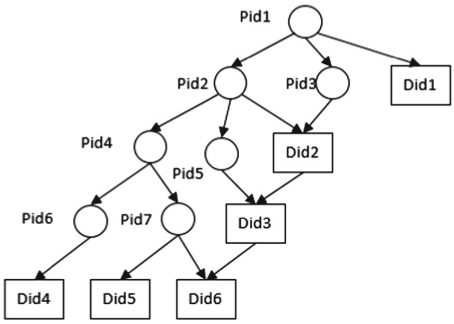


Fig. 2. Our HIBE example

children nodes Pid_2 and Pid_3. A smart device, with ID number Did_1 can only be controlled by Pid_1. Home owner Pid_1 generates public parameters and master key using Setup mentioned above. For the children nodes Pid_2 and Pid_3, home owner generates two tuples (Pid_1, Pid_2) and (Pid_1, Pid_3). Then the owner uses KeyGen to produce two private keys for the two children nodes. In order to send the keys to the children nodes securely, we choose the broadcast encryption scheme proposed in Algorithm 2 which needs a presetting secret key sk_u for each node. The parent node first sets up an identity space ID, and then produces a message (Hdr, c) for each user and broadcast it to children node group. The children nodes decrypt the message using their presetting secret key sk_u to get the private key d_I. For the device Did_1, the key distribution process is as same as the children nodes. However, there is another scenario that another person may also need to control it after the key distribution. Therefore, we are required to add an element between Pid_1 and Did_1. To solve this problem, Pid_1 can issue a private key to the new element and send the encrypted d_{Did_1} to the owner. Then the new element can be inserted into the hierarchy successfully.

In another scenario, child node Pid_3 buys a smart device Did_2 and wants to set it up within this hierarchy. First, Pid_2 generates private keys for the device using public parameters of the parent node. Then Pid_2 encrypts the devices' private r using Encrypt in Algorithm 1 and sends back to the parent node. The parent node can decrypt it using Decrypt and get r in order to obtain the private key of the device. Therefore, Pid_1 can control and gain the data generated by device Did_2.

Sometimes, a node in the hierarchy needs to update the decryption, or even the node should be revoked in this hierarchy. For example, the node Pid_4 needs to be revoked in the hierarchy. We first generate a subtree that contains Pid_4 from the root node to the leaf nodes. In this hierarchy, the node Pid_4 has a parent node Pid_2 and two children nodes Pid_6 and Pid_7. Therefore, Pid_2 generates two private keys for Pid_6 and Pid_7 and recursively the descendant nodes of Pid_4 get new keys and send the encrypted r using ID to all ancestor nodes. Then the node Pid_4 has been successfully revoked.

Algorithm 2. Algorithm for Broadcasting Smart Home Keys

Initialization:
Home owner sets up a broadcast scheme for identity space ID. It outputs public parameters as well as a master secret key.

KeyGen:
Takes the master key and a user $u \in ID$ and outputs a secret key d_I

Encryption:
takes the public parameters and a subset $S \in ID$, and produces a (Hdr, c). c is encrypted by using device ID.

Decryption:
takes the header Hdr and presetting secret key sk_u and outputs the key d_I

Algorithm 2 gives a broadcast encryption scheme for distributing the key for each element in the hierarchy. In order to achieve the broadcast encryption scheme, the manufactory of the smart device will preset a secret key sk_u into the smart device. When a home member issues a private key d_I to the child node, she generates a tuple (Hdr, c), where c is encrypted by using ID. The child node can decrypt c and get the private key d_I after receiving the tuple that is indicated to her. By executing this scheme, the child node can get data like SSID and Wi-Fi password in a secure way and connect to the gateway securely.

Besides the above HIBE scheme, we also provide some security enhancements in our SecHome system which are described in the following.

Initialization Key. The SecHome's smart phone and device hardware platforms come with an initialization key. Access to this key is controlled by the MPU and only trusted software components are given access to it. HIBE keys can be derived from the key.

Memory Protection Unit (MPU). SecHome's smart device side is based on MPU provided by [21]. The MPU provides secure initialization and configuration which can act as a root of trust for the HIBE scheme.

Secure Boot. SecHome's trusted software components are loaded with secure boot and isolated from the rest of the system by the MPU to ensure their integrity.

Cryptographic Hash. We rely on a keyed pseudo-random function and a collision-resistant cryptographic hash function. Our implementation uses HMAC and SHA-256.

Authenticated Encryption. For HIBE encryption, we use a stream cipher that provides authenticated encryption with associated data. The associated data is authenticated, but not included in the ciphertext. In this way, it is difficult to forge any ciphertext and the security against chosen ciphertext attack can be provided.

4.3 Security Considerations of Our System

We now discuss several attack scenarios on SecHome which are partly outside the adversary model.

Server Spoofing Attack Resistance. Compared to other classical remote authentications, our system also allows users to verify the server side in order to avoid server spoofing attacks.

Replay Attacks Resistance. The adversary may try to attack in various ways by replaying a prior command. In order to prevent this kind of attacks, our scheme provides online key exchange protocol, so that the users can simply refuse to give d_{ID} a second time to any other nodes.

Masquerade Attack Resistance. Our system is session based and in each session a receiver will be assigned a dynamic salt. A dynamic salt can hide the real ID from eavesdropping, and only be valid in a certain session. Whenever an expired ID is received, the receiver can simply discard those requests.

Data Isolation. Many low-end platforms do not have multiprocessing and virtual memory, but the MPU used in our SecHome system can achieve data isolation with memory access control.

Overall, SecHome achieves all security and functional requirements better than previously proposed solutions in smart home system.

5 Experiments and Results

5.1 Experiment Setup

We have implemented the above architecture of the secure smart home system. The system consists of three parties which include public cloud server, smart phones and smart home devices. The public cloud is implemented in Java. The interfaces of the cloud are exposed as web services, and the web services are hosted in Apache Tomcat. The clouds use MySQL database which can be easily replaced by other databases for server side data storage. The smart phone side is written with Android which can run in any smart phone with Android platform support. The smart device is written with C programming language which deploys on Raspberry Pi2. To ensure that the smart devices side gets the valid secret key, these keys are embedded in the smart devices, and the smart devices are signed by the key generated by the trusted certificate authority when the devices are initialized.

Our HIBE scheme uses asymmetric bilinear groups, and the bilinear map takes inputs from an isomorphic groups G. In our implementation, we use jPBC [24] and PBC [25] as our pairing-based cryptography library. We use ChaCha20 as the symmetric encryption algorithm. The reason for our choice is that ChaCha20 has better performance on mobiles and smart devices with ARM platform. We consider ChaCha20 as a secure symmetric encryption algorithm since ChaCha20 is designed to meet the standard notions of privacy and

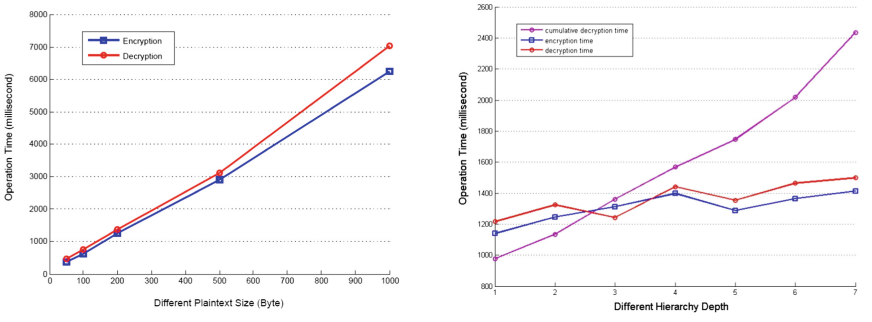
authenticity and ChaCha20 can provide a 256-bit security level [26]. We have performed our experiments on a cluster server with two 6-core Intel(R) Xeon(R) CPU 1.90 GHz processors, 16 GB of RAM, and 6 TB 7200 RPM hard disks, which are connected by gigabit switched Ethernet. On the smart phone side, we performed our experiments on a smart phone equipped with a Samsung Exynos4412 CPU 1.5 GHz processor, 1 GB of RAM and 8 GB flash disk. On the smart device side, we have performed our experiments on Raspberry Pi2 equipped with a ARM Cortex-A7 based BCM2836 CPU 900 MHz processor, and 1 GB of RAM.

5.2 Results on Effectiveness

Encryption and decryption are the most frequently used operations in the system. Since we have smart phone side and smart device side that need to encrypt and decrypt their data, we first measure the time taken at the smart phone for performing encryption and decryption. The time for smart phone decryption is measured from the time the smart phone receives the encrypted data from the private cloud, to the time the smart phone starts to display the data to the home member.

Figure 3(a) shows the time that the smart phone has spent in executing the encryption and decryption algorithm on different sizes of data. In this case, increasing the size of the plaintext also increases the decryption time; increasing the number of ancestor nodes has the same influence on the encryption and decryption time. However it is important to note that the number of ancestors is usually much smaller than the increasing of the plaintext data.

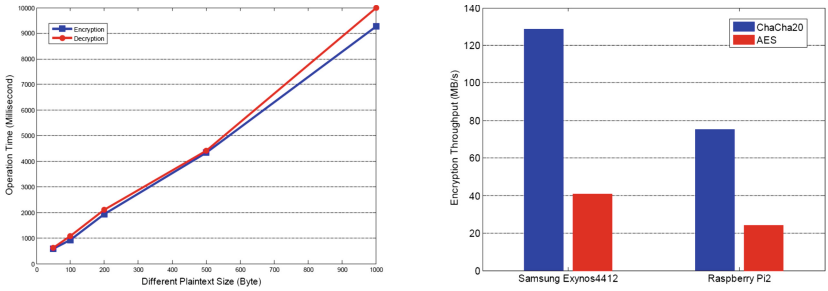
Figure 3(b) shows the time that the smart phone has spent on different depth of the hierarchy. In this experiment, we have created a hierarchy with depth of 7. From the result we can get the depth has a minor influence on each node's encryption and decryption performance.



(a) Different Plaintext Size

(b) Different Hierarchy Depth

Fig. 3. Smart phone operation time



(a) Different Plaintext Size

(b) Different Hierarchy Depth

Fig. 4. Smart device operation time

Next we look at the smart device side operation time. Figure 4(a) shows the time for encrypting and decrypting files of different sizes on the smart device side. In this experiment, we have created a hierarchy with depth of 5. In our measurements, the encryption time is measured from the time when a sensor starts to collect the data to the time the smart device starts sending the encrypted data to the cloud. The decryption time is measured from the time the cloud starts sending encrypted data to the time the smart device starts executing the command. From the result, we can believe that SecHome has the potential to be used in many commercial situation.

Since we use ChaCha20 as our symmetric encryption algorithm, the encryption and decryption can happen while the data is being transferred between smart phones, smart devices and cloud. Figure 4(b) shows the comparison of ChaCha20 and AES which demonstrates that ChaCha20 is more suitable for smart phones and devices.

6 Conclusion

In this paper, we present SecHome, a system to provide a secure hierarchical identity encryption for smart home to protect users' security and privacy. Our system protects users' data privacy and security from a cryptography perspective, and we show that our system enables each node to monitor her descendant node data in a secure way successfully. We also give a security analysis of our system. As far as we know, this is the first solution that has privacy considerations for smart home's hierarchy properties. Our main result works for smart home encryption, and we extend it for authentication and initialization protection as well. We also implement a prototype of our system and show that the overhead of our system is insignificant. Evaluations of the security and complexity show that the nodes can be protected and monitored, unless the computing resource of the nodes is extremely low.

Acknowledgement. We would like to thank the anonymous reviewers for their insight and detailed feedback. Our work was supported by The National Key Research and Development Program of China NO.2017YFB0801900 and Youth Innovation Promotion Association of CAS.

References

1. Brush, A., Lee, B., Mahajan, R., Agarwal, S., Saroiu, S., Dixon, C.: Home automation in the wild: challenges and opportunities. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2115–2124. ACM (2011)
2. Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)
3. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of things (IoT): a vision, architectural elements, and future directions. Future Gener. Comput. Syst. **29**(7), 1645–1660 (2013)
4. Weber, R.H.: Internet of things-new security and privacy challenges. Comput. Law Secur. Rev. **26**(1), 23–30 (2010)
5. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_5
6. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_13
7. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_14
8. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_7
9. Waters, B.: Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03356-8_36
10. Liang, K., Liu, J.K., Wong, D.S., Susilo, W.: An efficient cloud-based revocable identity-based proxy re-encryption scheme for public clouds data sharing. In: Kutyłowski, M., Vaidya, J. (eds.) ESORICS 2014. LNCS, vol. 8712, pp. 257–272. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11203-9_15
11. Zhou, L., Varadharajan, V., Hitchens, M.: Achieving secure role-based access control on encrypted data in cloud storage. IEEE Trans. Inf. Forensics Secur. **8**(12), 1947–1960 (2013)
12. Goyal, V., Pandey, O., Sahai, A., Waters, B.: 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 (2006)
13. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, SP 2007, pp. 321–334. IEEE (2007)
14. Li, M., Yu, S., Zheng, Y., Ren, K., Lou, W.: Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE Trans. Parallel Distrib. Syst. **24**(1), 131–143 (2013)

15. Wan, Z., Liu, J.E., Deng, R.H.: Hasbe: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing. IEEE Trans. Inf. Forensics Secur. **7**(2), 743–754 (2012)
16. Jung, T., Li, X.-Y., Wan, Z., Wan, M.: Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption. IEEE Trans. Inf. Forensics Secur. **10**(1), 190–199 (2015)
17. Horwitz, J., Lynn, B.: Toward hierarchical identity-based encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466–481. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_31
18. Shao, J., Cao, Z.: Multi-use unidirectional identity-based proxy re-encryption from hierarchical identity-based encryption. Inf. Sci. **206**, 83–95 (2012)
19. Blazy, O., Kiltz, E., Pan, J.: (Hierarchical) identity-based encryption from affine message authentication. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 408–425. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2_23
20. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_26
21. Koeberl, P., Schulz, S., Sadeghi, A.-R., Varadharajan, V.: Trustlite: a security architecture for tiny embedded devices. In: Proceedings of the Ninth European Conference on Computer Systems, Article no. 10, p. 1. ACM (2014)
22. Costin, A., Zaddach, J., Francillon, A., Balzarotti, D., Antipolis, S.: A large-scale analysis of the security of embedded firmwares. In: USENIX Security Symposium (2014)
23. Cui, A., Stolfo, S. J.: A quantitative analysis of the insecurity of embedded network devices: results of a wide-area scan. In: Proceedings of the 26th Annual Computer Security Applications Conference, pp. 97–106. ACM (2010)
24. Caro, A.D., Iovino, V.: Java pairing based cryptography library (2011). http://libeccio.dia.unisa.it/projects/jpbc
25. Lynn, B.: Pairing-based cryptography library (2007). http://crypto.stanford.edu/pbc
26. Nir, Y., Langley, A.: ChaCha20 and Poly1305 for IETF Protocols (2015). https://tools.ietf.org/html/rfc7539