Research Article

Secure and Efficient Control Transfer for IoT Devices

Zhi Wang, Han Ding, Jinsong Han, and Jizhong Zhao

School of Electronic and Information Engineering, Xian Jiaotong University, Xian 710049, China

Correspondence should be addressed to Zhi Wang; zhiwang@mail.xjtu.edu.cn

Received 3 July 2013; Accepted 2 October 2013

Academic Editor: Yunhuai Liu

Copyright © 2013 Zhi Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The prevalence of Internet of Things (IoT) requires flexible and fine-grained controls over the IoT devices. Existing works rely on specific controllers or programs to remotely control IoT devices, which is inefficient to support intelligent control in IoT environments. In contrast, utilizing a common portal device, for example, smartphone, to control variant IoT devices is a promising solution. However, it is challenging to guarantee the security when transferring the control of IoT devices. In this paper, we design a lightweight protocol to enable secure control transfer among the IoT devices, portal devices, and backend server. We demonstrate the effectiveness of our protocol in defending against mainstream attacks. Experimental results show the efficiency of our protocol in the authentication and key-updating during the control transfer.

1. Introduction

With the rapid development of electronics, existing electronic devices, such as the TV, refrigerator, electromagnetic oven, and even electric lamp, become intelligent such that they can enable adaptive control to meet time-varying needs of users. With the emergence of the Internet of Things (IoT), many IoT devices have been equipped with IoT technology [1–5], which provides more intelligent services to users.

Remotely controlling these devices is essential for supporting the intelligent IoT service. Most of existing devices adopt the former patterns, which is cost-inefficient. Some recent solutions are designed to adopt portal devices, for example, smart phones or portal tablets, to control IoT devices. Those works still depend on the specific program designed for targeted devices; however, they are suffering from inefficiency and poor scalability.

To enable more flexible and fine-grained controls over the IoT devices, we are motivated to design a common control management system, which allows users to leverage their portal devices, for example, smart phones, to control variant devices in IoT environments. Imagine a scenario of smart space in hotels. When passengers check in, they can use their own smart phones or iPad-like devices to control the IoT devices in the room. In this way, the service will be more convenient to the customer and the cost of maintenance for the hotel will be significantly reduced. Our control system consists of IoT devices, portal devices, and backend server. The control over IoT devices should be managed by the backend server. For each user, his portal device plays the role as a single access entry to control different IoT devices. After validating the user, the backend server will transfer the control on the targeted devices to the user's portal device. The user then employs his portal device to control different targeted devices, including operating on a TV, controlling the air condition, or selectively playing CDs on a HiFi.

However, using a portal device to safely control IoT device is nontrivial in IoT applications. In particular, it is challenging to guarantee that the control can be securely transferred from the backend server to a validate user's portal device, considering the insecure IoT environments. For example, a host would like to allow his close friends to control his main door while an intruder should not be allowed to open it. Obviously, effective authentication or verification is of importance to transfer controls among IoT devices, portal devices, and backend server. Nevertheless, IoT devices are usually source limited. Thus, it is difficult and cost-inefficient to directly adopt complex asymmetric key-based algorithms for authentication. It is also challenging to securely control IoT devices with low overhead.

In this paper, we propose a secure and efficient control transfer protocol to allow users to control IoT devices using...
their portal devices, for example smart phones. We use a lightweight cryptographic hash function as the crypto-building block to support authentication in source-limited IoT devices. Each IoT device will share a transfer key \( K_T \) with the backend server. The IoT device also has a control key \( K_C \). When a user wants to control an IoT device, he sends a request to the IoT device via his smart phone. The IoT device replies with a cipher encrypted with \( K_C \). The smart phone relays this response to the backend server. The backend server locates the \( K_C \) in the database and hence identifies the IoT device. Then, the backend server can verify the validation of the user. If the user is valid, the backend server grants the control over the requested device and an updated \( K'_C \) to the user’s phone. The control is indeed a chipper encrypted with the key \( K_T \) and \( K'_C \). The user relays the control over the IoT device. The IoT device checks the validation of the control and gets \( K'_C \) by using \( K_T \). Finally, the user’s phone obtains the control, \( K'_C \), from the backend server. The main contributions of our protocols are as follows.

(i) We propose a common control transfer scheme, in which the control over IoT devices does not depend on specific controllers. The user can leverage his portal device as a common controller to control different IoT devices. To our knowledge, our work is among the first steps towards secure control transfer over IoT devices.

(ii) We perform a lightweight hash function as the cryptographic component, which significantly improves the authentication efficiency with sufficient security guarantee.

(iii) We conduct comprehensive analysis on the security of proposed protocol and demonstrate its effectiveness in defending against existing attacks. We also perform simulations to evaluate the performance of our protocols. The results show that our protocols can achieve secure control transfer with low overhead and latency.

The rest of the paper is organized as follows. In Section 2, we discuss related works. We outline the requirements of secure control transfer in Section 3. In Section 4, we present our protocol design. We analyze the privacy and security of protocol in Section 5. Section 6 reports the simulation result and Section 7 concludes the paper.

2. Related Work

In this section, we discuss the related work in authentication and ownership transfer in IoT environments.

Authentication. In RFID systems, the identity of tag (TID) is easily traced and cloned. Many solutions, such as [6–10], have been proposed to employ traditional cryptographic methods to enhance the IoT systems. However, these protocols are not often compatible with commercial industrial standards, for example, EPC C1G2 specifications [2]. Due to the extremely limited resource, RFID tags require lightweight solutions, such as hash functions [3] or PRNGs [11], in the authentication. Chabanne and Fumaroli [12] propose a noise-based protocol, in which they adopt a Bit Pair Iteration scheme to correct transmission errors and prevent passive eavesdropping. Juels protocol [13] generates a PINSet to provide reader-to-tag authentication. After hiding the correct access password (the correct PIN value) in a set containing \( k \) values, the probability of a correct guess is \( 1/k \), and the security level approaches \( O(k) \).

M^2 AP [14] is an ultralightweight mutual authentication protocol for low-cost RFID tags by only using simple operations such as XOR, OR, AND, and SUM of modulo. Later, researchers present some attacks to \( M^2 \) AP [15]. SASI [16] is another ultralightweight protocol using the same basic operations as \( M^2 \) AP, but it suffers from the desynchronization attack. Without changing the protocol flow of existing air interface protocol, the authors in [1] propose a new authentication protocol-Gen2+, which is a multiple round protocol using shared pseudonyms and cyclic redundancy check (CRC). It achieves the reader-to-tag authentication and provides sufficient security level for real-world settings. Similarly, the works proposed in [2, 17] also aim to be fully compatible with EPC C1G2 specifications.

Ownership transfer has two fundamental requirements. First, the former owner should not obtain the secret of new owner after the transfer. Second, the new owner should not deduce past transactions as well as the secret related to former owner. There are also a few solutions to address the ownership transfer problem. Some of them rely on hash functions or symmetric encryption functions [15, 18–25]. The authors in [26] propose a two-party ownership transfer protocol, which is based on the security of the backwards channel. The authors in [27] design a mutual authentication protocol for secure ownership transfer. Their work is based on minimal hardware Physical Unclonable Functions.

The work proposed in [28] presents a detail survey on the security requirements, such as indistinguishability, forward security, resistance against replay attack, resistance against tag killing, and ownership transferability. In order to securely deliver the services to the user devices, the authors in [29] propose a technique “secure three way authentication (STWA),” intending to protect the user privacy and accomplish ownership authentication.

3. Control Transfer Protocol Requirements

In this section, we first present our system model and assumptions. We then introduce the requirements of privacy, security, and performance relevant to control transfer protocol design.

3.1. System Model and Assumption. The control transfer protocol considered in this paper operates with the following model and assumptions.

(i) The communicating parties include a server, a portal device, for instance a phone, and an IoT device. The server also plays the role as a trust center (TC). The portal device acts as a common controller, denoted as P. We denote the IoT device as IoD.
(ii) P and IoD communicate via an unsecure wireless channel. In this paper, the wireless protocol is Zigbee.

(iii) TC and P communicate via a WiFi connection protected by WPA/WPA2.

(iv) TC maintains a database containing the secure information for IoD and P.

(v) IoD is source limited and hence can only afford lightweight cryptographic operations, for example, hash functions. We assume that such a device usually has a rewritable memory that is tamper resistant.

3.2. Privacy Requirements. There are two major threats to the user privacy in remote control systems [2, 3, 18, 19, 27].

Control Leakage. In a typical remotely controlling system, when TC queries a device D, D responds with its profile. If unauthorized entities obtain its profile, they may be able to obtain the control on P from the TC’s database. The essential requirement of the control privacy is that only authorized entities are able to access the information and control associated with IoDs.

Device Tracking. If the responses of an IoD are distinguishable from those of other devices, the device can be tracked. Even worse, the social interactions of the individual user carrying the IoD may be disclosed while the user is unaware of the risk of being traced. Remotely controlling systems should be able to resist the device tracking attack by making the messages from devices indistinguishable from others. In detail, the track resistance should meet two guarantees. (a) New controller privacy: once the control on an IoD has been transferred to a new P, only the new P can identify and control the device. The previous controller of the device should no longer identify or control the device. (b) Old controller privacy: when the control on an IoD has been transferred to a new P, the new P should not trace past interactions between the IoD and its previous controller P.

3.3. Security Requirements. Communications between a controller and a device via an insecure wireless channel are susceptible to attacks. We outline the major attacks threatening remotely controlling systems. Due to the cost constraint, IoT devices usually cannot afford complex cryptographic algorithms to provide privacy and security. Therefore, remotely controlling schemes should meet the following requirements [2, 12–14, 18, 19].

Resistance to Device Impersonation. The attacker can impersonate a targeted IoD to a P, without knowing the device’s internal secrets. If the attacker succeeds, it will be authenticated as the targeted device. In our work, an adversary should not be able to impersonate an IoD by compromising attacks.

Resistance to Controller Impersonation. The attacker can impersonate a legitimate P to a compromised IoD. In this case, the attacker may need the knowledge of the device’s internal state. If the attacker succeeds, the attacker may ask the IoD to update its internal state. As a result, any legal P will no longer be able to successfully communicate with IoDs [17]. We should resist such an impersonation, even if the adversary compromises the IoDs.

Resistance to Replay Attack. The attacker can replay messages previously exchanged between a legitimate P and IoD. If the attack succeeds, the attacker may conduct a successful authentication between a device and a controller. We should prevent the adversary from building a session with a legitimate P and IoD by replaying their previously exchanged messages.

Resistance to Man-in-the-Middle (MITM) Attack. The attacker can block messages transmitted between a legitimate P and IoD; they would no longer be able to authenticate each other. Thus, blocking the messages transmitted between a legitimate P and IoD should not lead to a desynchronization.

Backward Traceability. An adversary should not be able to trace the prior transactions between a legitimate P and IoD, even if it compromises the devices [6, 20, 21, 27].

Forward Traceability. An adversary should not be able to deduce future transactions between a legitimate P and IoD, even if it compromises the devices.

3.4. Performance Requirements

(i) Small Storage: the volume of data stored in an IoD should be minimized.

(ii) Low Computation Complexity: the complexity of computations, especially those for cryptooperations, should be minimized.

(iii) Efficient Communication: the number and size of messages exchanged between a legitimate P and IoD should be minimized.

4. Control Transfer Protocol

In this Section, we present our protocol, control transfer (CT). The basic idea of CT is that a P can obtain the control on an IoD from TC. CT is realized by authentication and key-updating among three major parties: TC, P, and IoD. CT is comprised of three phases: initialization, control transfer, and control confirmation. In the initialization phase, TC will issue two keys to IoDs. One is the control key $K_A$, another is the transfer key $K_T$. Note that TC knows every IoD’s keys. But P only knows $K_A$ of an IoD if P obtains the control of this device. In the control transfer phase, P and IoD perform mutual authentication with the help of TC. At the meantime,
P will be granted the control on IoD from TC, for example, obtaining a new control key from TC, if it is verified by the IoD and TC. In this phase, IoD will update its transfer key. In the control confirmation phase, P will confirm that its control key is the same as the one on the IoD side and IoD has successfully updated its transfer key. Then, P will inform TC to update the transfer key in the database for synchronization.

Note that the communication between TC and P is via a secure WiFi connection, for example, protected by WPA/WPA2. While the communication channel between P and IoDs is not secure, we employ Zigbee as the communication protocol between P and IoDs in this paper.

### 4.1. Initialization

First, we define the system parameters as follows:

- (i) \( l \) is the bit length of a key.
- (ii) \( l_r \) is the bit length of a random string.
- (iii) The concatenation operator is represented by \( \| \). \n- (iv) \( K_C \) is the control key, which is initially generated by TC and embedded in an IoD during the issuance of the IoD. Once the control on the IoD is transferred to a new P, \( K_C \) will be updated. We denote \( K'_C \) as the new control key updated from \( K_C \).
- (v) \( K_T \) is the transfer key shared by TC and IoD. \( K_T \) should never be disclosed to any third party, including P. \( K_T \) will be updated on the side of IoD and synchronized with TC along with the control transfer and confirmation phases. \( K_T \) is also issued by TC. For each IoD, there is an entry in the database of TC containing a tuple \((K_C, K_T)\) with \( K_C \). The IoD is valid.
- (vi) \( h \) is a lightweight hash function, \( h : \{0,1\}^* \rightarrow \{0,1\}^l \). We assume that TC and IoD have the same hash function \( h \). \( h \) is collision resistant and suitable for implementation in IoT devices.

### 4.2. Control Transfer

The phase of control transfer is comprised of two steps: authentication and key-updating. Figure 1 shows the entire authentication process.

#### 4.2.1. Authentication

(a) When a user wants to use his portal device P to control an IoD, P first generates a \( l_r \)-bits random number \( r_1 \) and sends \( r_1 \) to the IoD. Then, the IoD computes \( N_1 = h(K_C, r_1) \) and sends \( N_1 \) back to P.

(b) P concatenates \( N_1 \) with \( r_1 \) and sends \( r_1 \| N_1 \) to TC. TC searches in its database to find \( K_C \). For each control key \( K_C \) stored in the database, TC computes \( M_1 = h(K_C, r_1) \) and compares it with \( N_1 \). If there exists a \( K_C \) such that \( M_1 = N_1 \), then \( K_C \) is found and the tuple of \((K_C, K_T)\) can be identified immediately.

(c) TC generates a \( l_r \)-bits random number \( r_2 \). It computes \( M_2 = h(K_C, r_1, r_2) \) and \( K'_C = h(K_C, r_2, K_T) \). TC then sends \( r_1 \| r_2 \| M_2 \| K'_C \) as well as a message of SUCCESS to P. This message is to inform P that the IoD is valid.

#### 4.2.2. Key Update

P forwards the \( r_1 \| r_2 \| M_2 \) to the IoD. The IoD calculates \( N_2 = h(K_C, r_1, r_2) \) using its own control key, \( K_C \), and compares \( N_2 \) with \( M_2 \). If there is a match, P is successfully verified by the IoD. Otherwise, the protocol terminates. If P is valid, IoD then updates \( K_C \) by calculating \( K'_C = h(K_C, r_2, K_T) \) and \( K_T \) by computing \( K'_T = h(K_T, r_1, r_2) \). The IoD sends a message \( N_3 = h(K'_C, r_1, r_2) \) to P.

#### 4.3. Control Confirmation

When P receives \( N_3 \), it computes \( M_3 = h(K'_C, r_1, r_2) \) using the \( K'_C \) received from TC. P compares \( N_3 \) with \( M_3 \). If they match each other, P is aware of the fact that IoD has successfully updated its \( K_C \) and \( K_T \).

One challenge is that the control confirmation might be interrupted if attackers block the delivery of \( N_3 \), resulting in a potential flaw that the \( K_T \) of IoT will be desynchronized with the one in the database of TC. To address the problem, we introduce a probe mechanism to P. The probe mechanism is indeed an iterative process. When P sends out the message of \( r_1 \| r_2 \| M_2 \) in the control transfer phase, it will set a timer \( T \). The length of the timer, denoted as \( |T| \), depends on the duration of key-updating step on the IoD side plus the delivery of \( N_3 \). If the timer is triggered, P will send a request \( R = \text{comm} \| r_1 \| h(K'_C, r_2, \text{comm}) \) to IoD, where the comm is a command to ask for resending \( N_3 \). When

| Trust center (TC) | Phone (P) | IoT device (IoD) |
|-------------------|-----------|------------------|
| \( K_C, K_T \)    | \( r_1 \) | \( N_1 \) |
| \( N_1 \)         | \( r_1 \| N_1 \) | \( M_2 \) |
| \( r_1 \| r_2 \| M_2 \| K'_C \) | \( N_2 \) | \( K'_C \| K'_T \) |
| \( r_1 \| r_2 \| M_2 \| K'_C \) | \( N_3 \) | \( N_3 \) |
| \( r_1 \| r_2 \| M_2 \| K'_C \) | \( N_3 \) | \( N_3 \) |

**Figure 1:** Control transfer protocol.
sending the request, P sets the timer again. When the timer is triggered, later P sends \( r_1 \parallel r_2 \parallel M_2 \) to IoD and sets the timer as \( |T| \). This process will be repeated for \( c \) times; \( c \) is a system parameter based on real applications. Note that P alternatively sends \( R \) and \( r_1 \parallel r_2 \parallel M_2 \). The purpose of this treatment is to guarantee that the protocol can be correctly executed no matter whether IoD updates its \( K_T \) or not. In our experiments, we set the value of \( c \) as 4. If P cannot receive any expected \( N_3 \) from IoD within \( c \times T \), P terminates the protocol and alerts TC that IoD is out of control. Along with the correct execution of the previously mentioned control transfer protocol, P and IoD achieve mutual authentication.

5. Privacy and Security Analysis

In this section, we analyze the privacy and security of our protocol based on the requirement raised in Section 3.

5.1. Privacy

Control Privacy. In our protocol, the control indeed is represented as the control key and transfer key shared among TC, P, and IoD. All these keys are not delivered in plaintext. We employ cryptographic hash function to generate ciphers for secure transmission. As a result, only TC knows the secret keys of P and IoD, and only legal P and IoD can successfully conduct the control transfer protocol. Therefore, our protocol is resilient to control leakage.

Tracking Resistance. As we analyzed in control privacy, the messages delivered in our protocol are encrypted using cryptographic hash function. The usage of random numbers further enhances the security. Due to the properties of hash function, the hash value of inputs will be evenly mapped to the output space. Thus, it is negligible to distinguish two devices from each other based on their messages, that is, the hash values computed from the involved keys and random numbers. In particular, P will never know the transfer key shared between TC and IoDs. As a result, P cannot reveal the new transfer keys of IoDs only based on its control key after a control transfer. On the other hand, the P will only get its control key from TC but have no idea of the control key used by the previous P. In short, our protocol can achieve privacy for both the old and new P.

5.2. Security

Resistance to Device Impersonation. For a legitimate P, a malicious IoD can launch the impersonation attack by manipulating \( N_1 \). However, it cannot succeed because of the lack of control key \( K_C \). Thus, the device impersonation is infeasible.

Resistance to Controller Impersonation. A P can only control a legitimate IoD after it obtains the control key from TC. Since the communication between TC and P is protected by WPA/WPA2, which supports TC to verify P a malicious party thereby cannot impersonate a legitimate P without the permission from TC.

Resistance to Replay Attack. Our protocol encrypts the messages using the time-varying random number as inputs. As a result, an adversary cannot relay messages previously exchanged between a legitimate P and IoD to successfully build a session between them.

Resistance to MITM Attack. Again due to the usage of hash functions on the components contained in the message, an adversary should not be able to manipulate messages exchanged between a legitimate P and IoD to perform cheating.

Resistance to DoS Attack. In our protocol, blocking most likely happens in the control confirmation phase, since the attacker can block the transmission of \( N_3 \) to yield a desynchronization. We address this problem twofold. First, we introduce an interactive probe mechanism for IoD to resend \( N_3 \) or \( r_1 \parallel r_2 \parallel M_2 \). Second, we set timers to avoid infinite loop of running the probe process. As a consequence, our protocol can effectively mitigate the impact of blocking attacks.

Backward Untraceability. For an IoD, its new control key \( K_C' \) is obtained by hashing the old control key \( K_C \), transfer key \( K_T \), and a random number generated by TC. Due to the one-way feature of hash functions, it is infeasible to recover old \( K_C \) and \( K_T \) based on the new control key. Then, the attacker cannot track the transactions of this IoD in previous sessions. The attacker can intercept the unsecure channel between the P and IoD to get the information used to compute the control key. However, the attacker still has no knowledge of \( K_T \) to compute the legal control key to trace past transactions.

Forward Untraceability. Because of the use of cryptographic hash function and key-updating during each control transfer, it is difficult for attackers to deduce future transaction messages of a given IoD. The most severe case is that an old P is malicious. Such a P can get \( r_1 \) and \( r_2 \) by overhearing the unsecure channel between the victim IoD and a new P. It then calculates the \( M_2 \) with \( K_C, r_1, \) and \( r_2 \). This may lead to a flaw that the attacker can track transactions in future sessions. In our protocol, the old P has no knowledge of \( K_T \), and the new control key \( K_C' \) is generated by TC using the transfer key \( K_T \). Enhanced by this treatment, even if the attacker intercepts the message exchanged between the legal P and IoD, it has no chance to reveal the new control key, let alone computing the future transactions. The privacy and security features of our protocol are summarized in Table 1.

6. Performance Evaluation

6.1. Experiment Setup and Metrics. Zigbee is a mainstream short-distance wireless communication technology with attractive features such as near distance, low complexity, low power consumption, low data rate, low cost, and flexible communicating mode. Those features make Zigbee suitable for the intelligent control IoT devices, especially for those
nonintelligent devices, such as lights, air conditioners, and refrigerators. On the other hand, we can also adopt other communication protocols or schemes, such as NFC or Bluetooth, in our solution. In this way, a concern about the usability or scalability may be raised considering the implementation of variant IoT device.

We set up a testbed to examine the performance of our control transfer protocol. The testbed simulates the real IoT environment. We employ a notebook to simulate TC. The portal device is by a combination of a cellphone (HTC Diamond) and a TelosB Node. TC and portal device communicate through IEEE 802.11.b/g in a WPA mode. We simulate IoDs using 10 TelosB nodes. The purpose of using TelosB nodes is twofold. First, the TelosB node is suitable for reflecting the limited resource of IoT devices. Second, TelosB nodes communicate with each other via Zigbee, which is a mainstream communication protocol in remotely controlling systems. The detail information of experiment setup is summarized in Table 2. We choose BKDR as the hash function and program it over the TelosB node. We conduct 1000 round tests over 10 simulated IoDs, say 100 times per IoD. In each test we perform a complete control transfer procedure.

Performance Metrics. We evaluate the performance of our protocol via three critical metrics, storage, computation overhead, and communication latency. The storage reflects how many bits one needed for storing $K_C$ and $K_T$ on the IoT device side. Considering the source constrain, this parameter should be minimized. We also have a concern on the computation efficiency of our protocol, especially for the IoD to conduct cryptographic hash functions. In addition, the communication latency of our protocol includes the message exchanged between IoD and P and between P and TC.

Performing authentication is important to ensure the security among control transfer processes. The authentication in our protocol is mainly based on the cryptographic hash functions. On the other hand, the efficiency of authentication is also important because most IoT devices work with weak capacity of computation and storage. Considering these, in order to analyze the efficiency of our protocol, we make a comparison with the schemes proposed in [30, 31]. The front protocol works in WSN and provides a two-factor user authentication before the legitimate users access data in the sensor/gateway nodes. The latter one is a bidirectional efficiency-privacy transferable (BEST) authentication protocol which can balance the privacy and communication efficiency dynamically.

### 6.2. Experiment Result

In our control transfer protocol, the storage mainly consists of the space for storing the keys on the sides of IoDs and TC. The length of hash value is 128 bits if using BKDR hash function, which determines the key size of our protocol. Since each IoD will have two keys, $K_C$ and $K_T$, the total storage is 256 bits in each IoD. During the authentication and key updating, each IoD also needs to temporarily store two random numbers, which are also in the same size as keys. Thus, the total storage will be 512 bits for each IoD. On the TC side, TC takes only $O(N)$ for the storage, where $N$ is the number of IoDs in the system.

A complete control transfer involves 10 hash computations among TC, P, and IoD. In particular, the IoD undertakes 5 hash computations. It is necessary to investigate the overhead of hash computation of each party, which indicates the computation complexity of our protocol. In Figure 2, we plot the average time for conducting one hash function in TC, P, and IoD, respectively. From the result, we can find that the IoD has lowest computation speed, 0.78 in average. This value is acceptable because the system can still afford more than 100,000 control transfers per second in this case. Considering the rapid configuration update on the hardware on the portal and IoT devices, the computation overhead of our protocol will be trivial in future.

We also check the communication latency when performing our protocol. The communication latency is mainly caused by the message exchange. We examine the time consumed to send $N_1, r_1 \parallel r_2 \parallel M_2,$ and $N_3$, because those messages are delivered between P and IoD, which may potentially become the system bottleneck. Figure 3 shows the average time used for transmitting above messages via the Zigbee channel. The result shows that the communication latency is sufficiently small to enable an efficient control transfer.

### Computation Cost

From Table 3, it is easy to find that in our protocol IoD requires 2 hash operations for authentication,
Table 2: Experiment setup.

| Role                  | Device type | Device model                     |
|-----------------------|-------------|----------------------------------|
| Trust center          | Notebook    | Thinkpad T400                    |
| Portal device         | Smartphone  | HTC Diamond 2; Qualcomm MSM7200A 528 MHz; |
| IoT device            | TelosB Node | Telosb Rev.B                      |
| Communication equipment | Wireless router and Zigbee component | TP-Link WR740N 300 M |

Table 3: Computation cost of the protocols.

| Scheme          | User/phone | Authentication | Sensor node/IoD |
|-----------------|------------|---------------|----------------|
| Das [30]        | 4H         | 4H            | 1H             |
| Qi et al. [31]  | —          | 1H            | 2H             |
| Proposed        | —          | 2H            | 2H             |

Figure 3: Transmission time.

which is the same as Qi's work, whereas the sensor node needs only 1 hash operation in M. L. Das' protocol. But from Figure 2, in our protocol, 2 hash operations are completed in 2 ms, which is acceptable for IoT devices and meets the requirements of controlling the device in our applications. In addition, M. L. Das' protocol does not provide mutual authentication between the sensor and gateway node, also Qi's protocol suffers from the DoS attack, while our protocol achieves higher security.

Communication Cost. Due to the cost constraint and limited source of IoT devices, we need an efficient communication between IoD and the controller. We compare the communication time and the size of exchanged messages in an authentication process. Three messages are exchanged for a successful authentication in M. L. Das', Qi's and our protocols. However, we observe that the total data size of three exchanges is different. In M. L. Das' protocol, about 532 bits are required. In Qi's protocol, a successful single tag authentication needs about 266 bits. However, the existence of inevitable conflicts enlarges the required size of message to be transferred. The demand size of message in our protocol is 512 bits. In summary, without a reduction in the performance, our protocol achieves better security enhancement.

7. Conclusions

In this paper, we propose a control transfer protocol to enable common portal devices to control large-volume IoT devices. The protocol leverages lightweight hash functions to achieve secure and efficient control transfer among resource-limited IoT devices. We analyze the privacy and security guarantee of our protocol. We also conduct simulations over real IoT devices to evaluate the performance. The results demonstrate the effectiveness of our protocol. Our future work includes releasing the constraint of using secure channel and conducting our protocol in large scale IoT applications.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (NSFC) under Grants nos. 61033015 and 61170220, the Fundamental Research Funds for the Central Universities of China under Project no. 2012jdgz02 (Xi’an Jiaotong University), and the Research Cooperation Special Funds of Guangdong under Project no. 2011B090400563.

References

[1] H. Sun and W. Ting, "A Gen2-based RFID authentication protocol for security and privacy," IEEE Transactions on Mobile Computing, vol. 8, no. 8, pp. 1052–1062, 2009.
[2] M. Burmester and J. Munilla, "Lightweight RFID authentication with forward and backward security," ACM Transactions on Information and System Security, vol. 14, no. 1, p. 11, 2011.
[3] T. Dimitriou, "A secure and efficient RFID protocol that could make big brother (partially) obsolete," in Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '06), pp. 269–274, March 2006.
[4] D. Zanetti, B. Danev, and S. Căpătun, "Physical-layer identification of UHF RFID tags," in Proceedings of the 16th Annual Con-
ference on Mobile Computing and Networking (MobiCom ’10), pp. 353–364, September 2010.

[5] N. Saxena, M. B. Uddin, J. Voris, and N. Asokan, “Vibrate-to-unlock: mobile phone assisted user authentication to multiple personal RFID tags,” in Proceedings of the 9th IEEE International Conference on Pervasive Computing and Communications (PerCom ’11), pp. 181–188, March 2011.

[6] C. C. Tan, B. Sheng, and Q. Li, “Serverless search and authentication protocols for RFID,” in Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom ’07), pp. 3–12, March 2007.

[7] M. Conti, R. Di Pietro, L. V. Mancini, and A. Spognardi, “RIPPF: an RFID identification, privacy preserving protocol with forward secrecy,” in Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom ’07), pp. 229–234, March 2007.

[8] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, “Security and privacy aspects of low-cost radio frequency identification systems,” in Security in Pervasive Computing, vol. 2802 of Lecture Notes in Computer Science, pp. 201–212, 2003.

[9] G. Tsudik, “A family of dunces: Trivial RFID identification and authentication protocols,” in Proceeding of the 7th International Symposium on Privacy Enhancing Technologies (PET ’07), 2007.

[10] M. Burmester, T. Van Le, and B. De Medeiros, “Provably secure ubiquitous systems: universally composable RFID authentication protocols,” in Proceedings of the 2nd International Conference on Security and Privacy in Communication Networks (SecureComm ’06), September 2006.

[11] E. Y. Choi, D. H. Lee, and J. I. Lim, “Anti-cloning protocol suitable to EPCglobal Class-1 Generation-2 RFID systems,” Computer Standards and Interfaces, vol. 31, no. 6, pp. 1124–1130, 2009.

[12] H. Chabanne and G. Fumaroli, “Noisy cryptographic protocols for low-cost RFID tags,” IEEE Transactions on Information Theory, vol. 52, no. 8, pp. 3562–3566, 2006.

[13] A. Juels, “Strengthening EPC tags against cloning,” in Proceedings of the 4th ACM Workshop on Wireless Security (WiSe ’05), pp. 67–75, September 2005.

[14] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevuez-Tapiador, and A. Ribagorda, “M2AP: a minimalist mutual-authentication protocol for low-cost RFID tags,” in Proceedings of the 3rd International Conference on Ubiquitous Intelligence and Computing (UIC ’06), 2006.

[15] M. Bárász, B. Boros, P. Ligeti et al., “Passive attack against the M2AP mutual authentication protocol for RFID tags,” in Proceedings of the 1st International Workshop on RFID Technology (EURASIP ’07), September 2007.

[16] H. Chien, “SASI: a new ultralightweight RFID authentication protocol providing strong authentication and strong integrity,” IEEE Transactions on Dependable and Secure Computing, vol. 4, no. 4, pp. 337–340, 2007.

[17] A. Juels and S. A. Weis, “Defining strong privacy for RFID,” ACM Transactions on Information and System Security, vol. 13, no. 1, p. 7, 2009.

[18] S. Fouladgar and H. Afifi, “A simple delegation scheme for RFID systems (SiDeS),” in Proceedings of the IEEE International Conference on RFID, 2007.

[19] B. Song, “RFID tag ownership transfer,” in Proceedings of the Radio Frequency Identification System Security (RFIDsec ’08), 2008.

[20] Y. Seo, T. Asano, H. Lee, and K. Kim, “A lightweight protocol enabling ownership transfer and granular data access of RFID tags,” in Proceedings of the Symposium on Cryptography and Information Security (SCIS ‘07), 2007.

[21] K. H. S. S. Koralalage, S. M. Reza, J. Miura, Y. Goto, and J. Cheng, “POP method: an approach to enhance the security and privacy of RFID systems used in product lifecycle with an anonymous ownership transferring mechanism,” in Proceedings of the ACM Symposium on Applied Computing, pp. 270–275, March 2007.

[22] K. Osaka, T. Takagi, K. Yamazaki, and O. Takahashi, “An efficient and secure RFID security method with ownership transfer,” in Computational Intelligence and Security, vol. 4456 of Lecture Notes in Computer Science, pp. 778–787, 2007.

[23] T. Li, W. Luo, Z. Mo, and S. Chen, “Privacy-preserving RFID authentication based on cryptographical encoding,” in Proceedings of the 31st Annual IEEE International Conference on Computer Communications (IEEE INFOCOM ’12), 2012.

[24] M. Li, C. Fung, K. Sampigethaya et al., “Public key based authentication for secure integration of sensor data and RFID,” in Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc ’08), pp. 61–65, May 2008.

[25] M. E. Hoque, F. Rahman, and S. I. Ahamed, “AnonPri: an efficient anonymous private authentication protocol,” in Proceedings of the 9th IEEE International Conference on Pervasive Computing and Communications (PerCom ’11), pp. 102–110, March 2011.

[26] J. Saito, K. Imamoto, and K. Sakurai, “Reassignment scheme of an RFID tag key for owner transfer,” in Embedded and Ubiquitous Computing-EUC 2005 Workshops, vol. 3823 of Lecture Notes in Computer Science, pp. 1303–1312, 2005.

[27] L. Kulseung, Z. Yu, Y. Wei, and Y. Guan, “Lightweight mutual authentication and ownership transfer for RFID systems,” in Proceedings of IEEE INFOCOM 2010, March 2010.

[28] K. Osaka, T. Takagi, K. Yamazaki, and O. Takahashi, “An efficient and secure RFID security method with ownership transfer,” in Proceedings of the International Conference on Computational Intelligence and Security (ICCIAS ’06), pp. 1090–1095, October 2006.

[29] B. H. Pradeep and S. Singh, “Privacy preserving and ownership authentication in ubiquitous computing devices using secure three way authentication,” http://arxiv.org/abs/1206.0978.

[30] M. L. Das, “Two-factor user authentication in wireless sensor networks,” IEEE Transactions on Wireless Communications, vol. 8, no. 3, pp. 1086–1090, 2009.

[31] S. Qi, L. Lu, Z. Li, and M. Li, “BEST: a bidirectional efficiency-privacy transferable authentication protocol for RFID-enabled supply chain,” in Proceedings of the IEEE 18th International Conference on Parallel and Distributed Systems (ICPADS ’12), December 2012.
