Received: 18 March 2021 | Revised: 4 June 2021 | Accepted: 31 July 2021

IET Information Security

DOI: 10.1049/ise2.12043

The Institution of Engineering and Technology WILEY

CASE STUDY

Authorisation inconsistency in IoT third-party integration

Jiongyi Chen[1] | Fenghao Xu[2] | Shuaike Dong[3] | Wei Sun[4] | Kehuan Zhang[1]

[1]School of Electronic Science and Engineering, National University of Defense Technology, Changsha, China

[2]Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China

[3]TianQian Security Lab, Ant Group, Hangzhou, China

[4]Department of Electric Engineering, Columbia University, New York, USA

Correspondence

Kehuan Zhang, Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong 999077, China.
Email: khzhang@ie.cuhk.edu.hk

Funding information

Hong Kong S.A.R. Research Grants Council Early Career Scheme/General Research Fund, Grant/Award Numbers: 14208019, 14208818

Abstract

Today's IoT platforms provide rich functionalities by integrating with popular third-party services. Due to the complexity, it is critical to understand whether the IoT platforms have properly managed the authorisation in the cross-cloud IoT environments. In this study, the authors report the first systematic study on authorisation management of IoT third-party integration by: (1) presenting two attacks that leak control permissions of the IoT device in the integration of third-party services; (2) conducting a measurement study over 19 real-world IoT platforms and three major third-party services. Results show that eight of the platforms are vulnerable to the threat. To educate IoT developers, the authors provide in-depth discussion about existing design principles and propose secure design principles for IoT cross-cloud control frameworks.

KEYWORDS

authorisation, computer network security, internet of things

1 | INTRODUCTION

The drastic evolution in the Internet of Things (IoT) has come to the stage where a growing number of IoT platforms can connect with third-party services, such as trigger-action platforms (e.g. IF This Then That [IFTTT]) and voice assistants (e.g. Amazon Alexa and Google Assistant). Such an integration has brought great convenience to IoT computer network securityusers by adding plentiful usage scenarios to existing IoT ecosystems, including data sharing, voice control, automatic actions etc, which has largely enriched the basic functionality of IoT systems. For example, users can talk to third-party voice assistants to control IoT devices or setup automatic rules like 'turn on the light after sunset' and 'close the window when it rains'.

Unlike the authorisation in traditional online services (e.g. Single-Sign-On) whose interaction is limited within three parties, the authorisation of third-party services in IoT involves not only the third-party cloud but also the participators within IoT systems including the device, the user, and the IoT cloud. As an example, a device owner can share her/his device to a guest user through the IoT cloud, and the guest who accepts the device sharing can also authorise third-party services to control this device. The interdependence between authorisation within IoT systems (e.g. device binding and device sharing) and authorisation with third-party services (e.g. third-party; control) has inevitably increased the authorisation complication.

Due to the complexity in IoT authorisation, concern may arise if the authorisation frameworks of IoT clouds have been well-protected. Traditional attack scenarios of insecure authorisation result from poor access prevention where the access is not sufficiently checked [1]. Thus, attackers can easily exploit insecure direct object references or hidden endpoints by manipulating API/URL parameters [2]. On the other hand, a line of research focusses on coarse-grained authorisation within IoT platforms [3–7]. They proposed various solutions to address improper permission designs. However, to the best of our knowledge, less attention has been paid to authorisation inconsistency in the cross-cloud IoT environments.

In this study, we report the first systematic study on authorisation inconsistency between the IoT cloud and the

third-party cloud, in an attempt to understand the cross-cloud authorisation mechanism in IoT environments. More specifically, we demystify the design architecture with third-party services, formalise the change of the authorisation state, and build constraints of the state transitions that involve multiple parties. As the consequence of such demystification, we present two real-world attacks that introduce authorisation inconsistency and allow attackers to gain unauthorized, stealthy, and persistent control of IoT devices.

Given the serious consequence of the threat, it is necessary to evaluate its impact on today's IoT ecosystem. To this end, we collected 19 IoT platforms whose backend clouds integrate with third-party services such as Google Assistant, IFTTT [8, 9] (IF This Then That) and Amazon Alexa. To evaluate authorisation inconsistency in those IoT platforms, we purchased one device instance of each platform and manually examined whether it violates the permission constraints. Unfortunately, we found that the authorisation inconsistency turns out to be prevalent: 8 out of 19 popular IoT platforms are vulnerable to at least one type of vulnerability. Given that the vendors either provide business-to-business services to device manufacturers or offer a series of devices to the market by themselves, one single vulnerable instance means that a large number of IoT devices are affected.

Our discovery brings to light new understanding of this new threat. Without clear understanding of the threat, case-by-case fixation does not help to avoid similar issues in the future. To this end, highlights of our research also include in-depth discussion about existing design principles and a proposal of two secure design principles to educate IoT developers. We believe that our discovery, discussion, and proposal would benefit the community and IoT developers on the new understanding about the emerging threat in the IoT domain.

1.1 | Contributions

The contributions of this study are outlined as follows:

- **New understanding of IoT authorisation inconsistency.** We report authorisation inconsistency in the integration of IoT third-party services, a new type of vulnerability that is specific to the IoT and never investigated before. Such a problem allows attackers to take unauthorized, stealthy and persistent control of IoT devices. Our study brings the new security threat to the spotlight and contributes to a better understanding in the attack vector.
- **Measurement and discussion.** We evaluate 19 popular IoT platforms in the market. The results indicate that eight of them suffer from authorisation inconsistency, with one single vulnerable instance affecting considerable users and devices. This reveals a worrisome situation in today's IoT development. As such, new standards and regulations are urgently needed in the development of an IoT control framework. To this end, we make the first step to provide in-depth discussion about design issues and propose two secure design principles to educate IoT developers.

1.1.1 | Roadmap

The rest of the study is organised as follows: Section 2 describes the background and the adversary model of our study. Section 3 elaborates on the authorisation inconsistency, including the preliminaries and the details of the vulnerability. Section 4 gives the empirical measurement on real-world IoT platforms. Section 5 discusses the fundamental causes of attacks, describes existing design issues, and presents two secure design principles. Section 6 surveys the prior related research and Section 7 concludes this study.

2 | BACKGROUND

In this section, we first introduce the authorisation procedures within IoT systems. Then we describe how third-party services are authorised to access resources in the IoT clouds and present a complete picture of multi-party authorisation in IoT scenarios. We describe the threat model at the end of this section.

2.1 | Authorisation within IoT systems

Authorisation operations of IoT systems include device binding and device sharing. They often involve the IoT device, the user and the IoT cloud. With regard to device binding, a user needs to be bound with a specific device through the IoT cloud before the user can remotely access that device. From the user's perspective, as the cloud is in charge of the user's access rights (i.e. absolute control) to the devices, the physical IoT device is regarded as his/her resource in the IoT cloud. In that sense, device binding is essentially an authorisation step. On the other hand, to support common application scenarios like smart home, smart hotel, smart office etc., the IoT vendors implement device sharing so that the device owners can temporarily grant access rights of their devices to guests. Particularly, to send invitation requests, the owner needs to input the account information of the guest. After the guest acknowledges the invitation, he/she can successfully use the device with the permissions that the owner grants.

2.2 | Authorisation of third-party clouds

In addition to the above authorisation operations, IoT systems also allow users to connect with external third-party services such as Amazon Alexa [10], Google Assistant [11], and IFTTT [12], which enable users to set automation rules or control the devices with voice commands. To this end, users operate in third-party mobile apps to link the corresponding IoT cloud (as shown in Figure 1a) and authorise the third-party services, so that they can provide secure delegated access to the resources in the IoT cloud (as shown in Figure 1b). In this step, the vendors adopt the de facto Single-Sign-On (SSO) standard protocol OAuth2.0 for authorisation. Specifically, as shown in

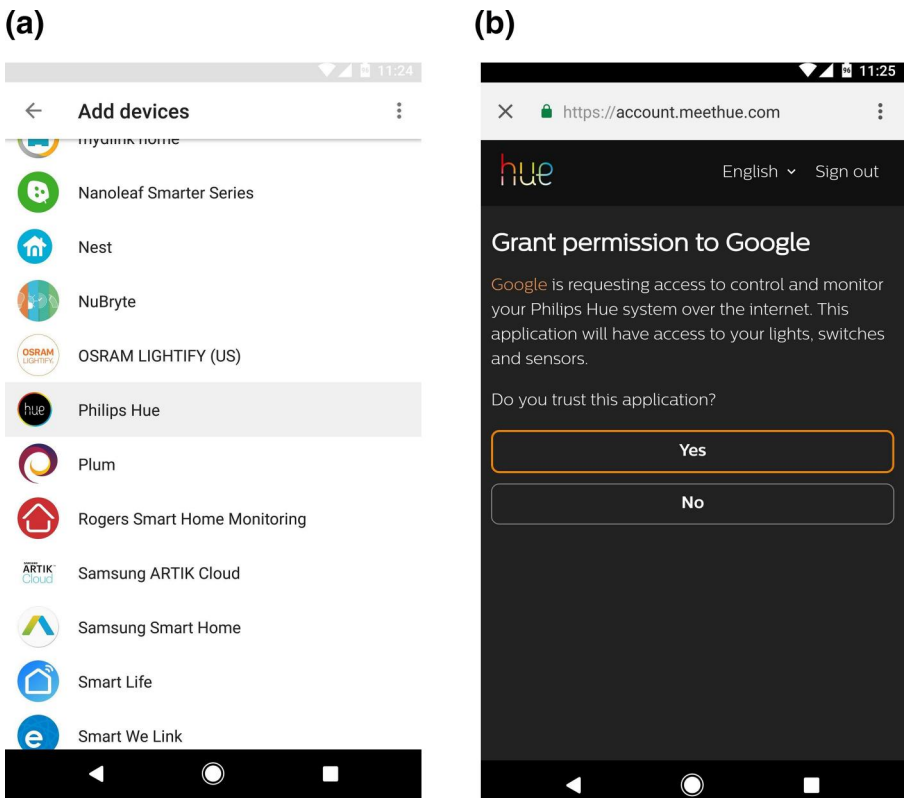

FIGURE 1 Third-party authorisation operations in IoT mobile apps

Figure 2, the third-party authorisation involves the following steps:

- **User-to-service authorisation.** The user initiates the SSO process with the third-party cloud and gives the IoT cloud his/her approval regarding the permissions requested by the third-party cloud.
- **Service-to-service authorisation.** With the user's approval, the IoT cloud distributes a *long-lived access token*[1] to the third-party cloud. When the user requests services from the third-party cloud, the third-party cloud uses the access token to acquire resources (e.g. the control of the user's device) in the IoT cloud.

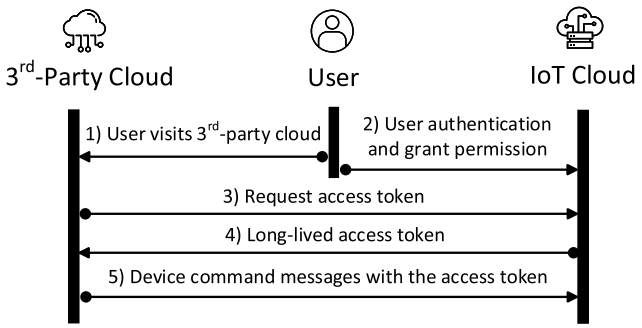

FIGURE 2 OAuth authorisation for IoT third-party control

2.3 | Multi-Party Control of IoT devices

As Figure 3 shows, once the authorisation within IoT systems and the authorisation of third-party services are completed, an IoT device can be controlled by four entities: the owner's IoT mobile app, the owner's third-party mobile app, the guest's IoT mobile app, and the guest's third-party mobile app. When the device control commands are initiated by the IoT mobile apps, requests are directly sent to the IoT cloud. In the cloud, the authorisation is often implemented using access control lists. When the control requests are initiated by the third-party mobile apps, they first send requests to the third-party cloud, and the third-party cloud communicates to the IoT cloud with the access token of the user (obtained from OAuth authorisation). As we can see, the IoT cloud plays an important role in managing the access control of the entire IoT ecosystem.

2.4 | Assumptions

In this study, the adversary's goal is to gain unauthorized permissions of the IoT devices. To this end, we consider the transfer of device control in authorisation scenarios. Specifically, we assume the adversary's capabilities based on the following scenarios:

[1]Third-party service providers recommend such a token never expire when implicit flow is used. When authorisation code flow is deployed, token exchange will be used to guarantee a long-lived refresh token. Such a refresh token can be used to exchange for a valid access tokens when the access token expires.

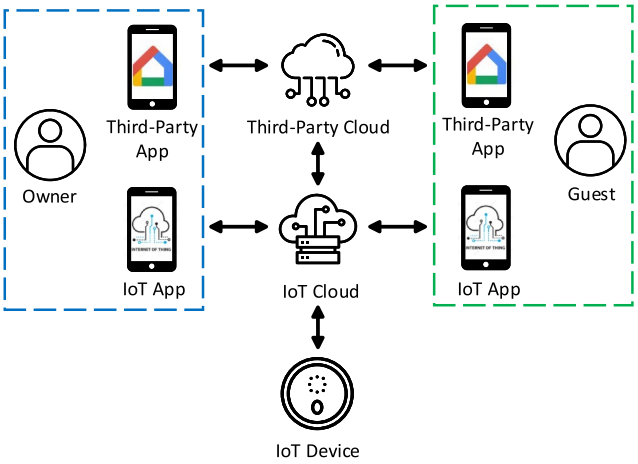

FIGURE 3 Multi-party control of the internet of things device

- **Device sharing.** The control permission of an IoT device could be temporarily authorised to another user in daily use. A prominent example is the use of the smart lock. In hotels, offices, and homes, the owner can authorise the open/close permission of a smart lock to a guest (i.e. a casual user). The guest who receives such a permission can temporarily use the lock during the period that is specified by the owner. On the basis of such scenarios, we assume that the adversary acts as the guest and had once received the control permission of an IoT device from the owner. More importantly, the device control permission is assumed to be revoked when the adversary launches the attack to gain unauthorized permissions.
- **Ownership transfer.** The IoT device is a 'thing' in nature and the device ownership transfer could take place in real life, such as device reuse, reselling, stealing and so forth. For example, the adversary can purchase an IoT device from Amazon, use it (gain control permission) and return it. Therefore, we assume that the adversary once had owned a device. After the device is transferred from the adversary to another user, we assume that the device has already been reset before it is put into use.

3 | ATTACKS

The authorisation of third-party services opens a new channel for controlling IoT devices. This allows users to control the IoT devices through IoT systems and third-party services. In this case, once the user authorises a third-party service, the third-party service acts as a delegation of authority for controlling the device. In this section, we build and illustrate end-to-end attacks with a concrete example of the IoT platform—*SmartLife* [13] and show that the attacker can leverage the third-party services to gain unauthorized control of the IoT device. Specifically, we first introduce the notations and the definitions of authorisation in cross-cloud IoT scenarios. Then we elaborate on the attacks with the help of formal notations.

3.1 | Preliminaries

3.1.1 | Authorisation states

The authorisation state is a set of capabilities represented by $\psi : \{Sub \xrightarrow{R} Obj\}$. Each capability specifies the fact that subject Sub is authorised to access object Obj with the permission set R. In addition, whenever the permission to the IoT device is changed, a state is transited to another state. Such a transition is accomplished by the authorisation operation Op, which is formally defined as the change of the authorisation state: $Op : \psi \Rightarrow \varphi$.

3.1.2 | Authorisation operations

In the adversary model, we consider that the attacker may gain unauthorized permission of the IoT device. Therefore, the authorisation operation is completed when a user's permission to the device is granted or revoked. In IoT scenarios, there are three types of authorisation operations: device binding, device sharing and third-party authorisation. Each of them also includes an inverse authorisation operation (i.e. unbinding, sharing deletion, third-party revocation) that restores the authorisation state to the original state:

- **Device binding & unbinding.** In the initial state, the user u has no access to the device d. With the completion of device binding, the user u can access the device d with designated permissions R (denoted in Equation (1)). Inversely, when the user u unbinds herself/himself with the device d, the permissions are revoked (i.e. $\varnothing$, denoted in Equation (2)).

$$u \xrightarrow{\varnothing} d \Rightarrow u \xrightarrow{R} d \tag{1}$$

$$u \xrightarrow{R} d \Rightarrow u \xrightarrow{\varnothing} d \tag{2}$$

- **Device sharing & sharing deletion.** The owner of the device u_1 can share the device to a guest user u_2. Originally, the owner u_1 can access the device with permissions R_1. After the sharing operation, the owner's permissions remain the same, while the guest u_2 receives the permissions R_2 that are granted by the owner (denoted in Equation (3)). Note that the guest u_2's permissions R_2 can be revoked by u_1 (denoted by $(\mapsto u_1)$). On the other hand, the owner can also delete such sharing, and the guest's permissions to that device are revoked accordingly (denoted in Equation (4)).

$$u_1 \xrightarrow{R_1} d \Rightarrow \{u_1 \xrightarrow{R_1} d, u_2 (\mapsto u_1) \xrightarrow{R_2} d\} \tag{3}$$

$$\{u_1 \xrightarrow{R_1} d, u_2 (\mapsto u_1) \xrightarrow{R_2} d\} \Rightarrow \{u_1 \xrightarrow{R_1} d, u_2 (\mapsto u_1) \xrightarrow{\varnothing} d\} \tag{4}$$

- **Third-party authorisation & revocation.** The third-party service is delegated to access a user's device after the user's

authorisation. In this study, we denote the third-party service under the name of the user u as: $\hat{u}$ and use $\hat{R}$ to represent the permissions of $\hat{u}$.

$$u \xrightarrow{R} d \Rightarrow \{u \xrightarrow{R} d, \hat{u}(\mapsto u) \xrightarrow{\hat{R}} d\} \tag{5}$$

$$\{u \xrightarrow{R} d, \hat{u}(\mapsto u) \xrightarrow{\hat{R}} d\} \Rightarrow \{u \xrightarrow{R} d, \hat{u}(\mapsto u) \xrightarrow{\varnothing} d\} \tag{6}$$

3.2 | Device sharing attack

As illustrated in Figure 4a, the device owner u_1 takes control of the device with permission R_1 and adds a sharing with the guest u_2 (device control ❶ and permission propagation ②). After the guest confirms such a sharing, the guest can take control of the device with permission R_2 (device control ❸). The authorisation state change is as follows:

$$u_1 \xrightarrow{R_1} d \Rightarrow \{u_1 \xrightarrow{R_1} d, u_2(\mapsto u_1) \xrightarrow{R_2} d\} \tag{7}$$

Also, the guest can authorise a third-party service ($\hat{u}_2\ (\mapsto u_2 \mapsto u_1) \xrightarrow{\hat{R}_2} d$) to obtain the permissions and uses the third-party app to control the device (permission propagation ④ and device control ❺). As can be seen from the representation, the third-party service is essentially a delegator's delegator. At this stage, there are three entities that can control the device: the IoT mobile app of the device owner, the IoT mobile app of the guest, and the third-party mobile app of the guest. The authorisation state becomes

$$\{u_1 \xrightarrow{R_1} d, u_2(\mapsto u_1) \xrightarrow{R_2} d, \hat{u}_2(\mapsto u_2 \mapsto u_1) \xrightarrow{\hat{R}_2} d\} \tag{8}$$

As shown in Figure 4b, if the owner revokes the sharing with the guest, the guest will lose his/her device control in the IoT cloud. However, because the internal control framework of the IoT cloud does not restrict the double-delegated permissions $\hat{R}_2$ within the authorising user u_1's control, the third-party service still contains the permission to control the IoT device, even if the guest's permission was already revoked by the owner (device control ❺). As a consequence, the guest could still use the third-party app to take unauthorized control of the device. Even worse, new users are unable to remove the invisible binding between the adversary and the device and would be unaware of the adversary's stealthy control. At this time, the authorisation state becomes

$$\{u_1 \xrightarrow{R_1} d, u_2(\mapsto u_1) \xrightarrow{\varnothing} d, \hat{u}_2(\mapsto u_2 \mapsto u_1) \xrightarrow{\hat{R}_2} d\} \tag{9}$$

whereas the following authorisation state is expected as

$$\{u_1 \xrightarrow{R_1} d, u_2(\mapsto u_1) \xrightarrow{\varnothing} d, \hat{u}_2(\mapsto u_2 \mapsto u_1) \xrightarrow{\varnothing} d\} \tag{10}$$

3.3 | Ownership transfer attack

The other attacking scenario is in ownership transfer where device control permissions can be changed. In this case, device ownership transfer could also leak control permissions to the adversary. As shown in Figure 5a, we consider that the adversary u_1 first registers his/her device in the IoT cloud so that he/she can take control of the device with permission R_1 (device control ❶). The adversary then authorises a third-party service and controls the device through the third-party service $\hat{u}_1(\mapsto u_1)$ (permission propagation ② and device control ❸). The authorisation state change is shown below:

$$u_1 \xrightarrow{R_1} d \Rightarrow \{u_1 \xrightarrow{R_1} d, \hat{u}_1(\mapsto u_1) \xrightarrow{\hat{R}_1} d\} \tag{11}$$

After that, the adversary can control the IoT device through both the IoT cloud and the third-party cloud. Next, the adversary re-distributes it to another user u_2 (ownership transfer ④). As shown in Figure 5b, the user who receives the physical device needs to perform the factory reset and confirm that the device is clear and not bound with previous users.

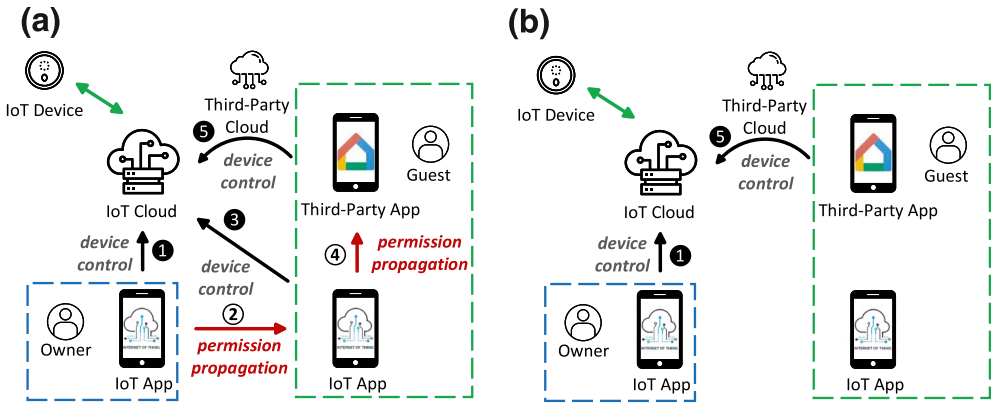

FIGURE 4 Illustration of device sharing attack. (a) Before the owner's revocation (b) After the owner's revocation

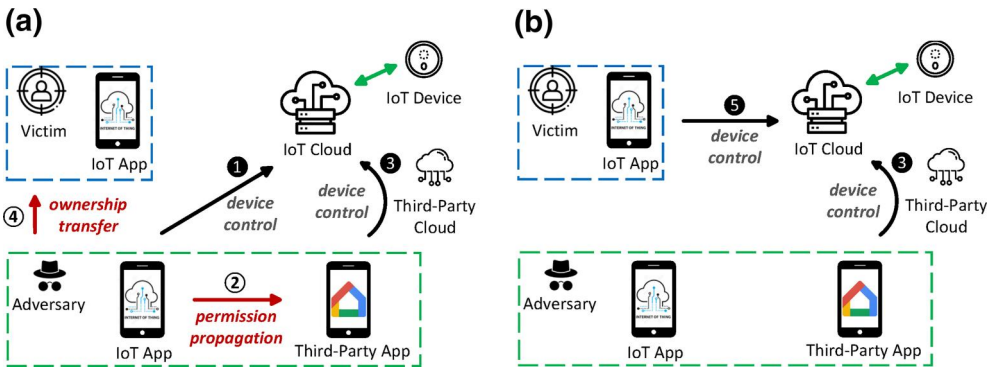

FIGURE 5 Illustration of ownership transfer attack. (a) Before the victim's reset and use (b) After the victim's reset and use

After resetting, the user u_2 can take control of the device (device control ❺). The state change of the reset and rebind operation is shown below:

$$\{u_2 \xrightarrow{\varnothing} d, u_1 \xrightarrow{R_1} d, \hat{u}_1(\mapsto u_1) \xrightarrow{R_1} d\} \Rightarrow \{u_2 \xrightarrow{R_2} d, u_1 \xrightarrow{\varnothing} d, \hat{u}_1(\mapsto u_1) \xrightarrow{R_1} d\} \quad (12)$$

However, although the binding relationship between the adversary and the device in the IoT cloud is cleaned by the factor reset, the adversary's binding with the device is permanently maintained through the third-party service (device control ❸). As a consequence, the adversary can take stealthy and permanent control of the device.

4 | EMPIRICAL MEASUREMENT

We demonstrated the feasibility and the details of the attacks in Section 3. In this section, we measure the prevalence of the attacks to understand their real-world impact. Specifically, we first provide the statistics about the evaluated IoT platforms and the experiment setup. Then we describe the evaluation results of the attacks on those platforms.

4.1 | Preparation

4.1.1 | IoT market

As IoT is on the stage of rapid development, considerable companies are extending their business in IoT, including small business and giant IT companies, especially those which provide cloud services like Google, AWS and Samsung. In such an emerging area, every player is a newcomer, which results in a *highly fragmented IoT market* [14]. Overall, there are three types of IoT providers: (1) cloud service providers, like Google, AWS, and Samsung,[2] who have powerful computation capability in the cloud and provide various cloud applications (e.g. voice assistant) to let IoT vendors and developers process a large volume of IoT data; (2) solution providers, who provide device management interfaces and release SDKs to integrate into traditional device manufacturers' solutions; (3) product providers, who have their own clouds and sell various IoT devices on the consumer market.

4.1.2 | Evaluated platforms and third-party services

To allow third-party control of IoT devices, more and more IoT solution providers and product providers have registered and integrated their platforms and clouds with third-party services/clouds. In this study, we evaluate those two types of providers' backend clouds that can manage IoT devices and integrate with third-party services. To evaluate the backend clouds, we purchased 19 IoT devices and tested their corresponding backend clouds. We listed all of the 430 IoT solution providers and product providers in Google Assistant under the category of 'Smart Home' [11] and downloaded the corresponding apps from Google Play App Store [15]. Those apps all support Google Assistant as a third-party integration. Based on that, we manually checked the product types of those platforms and selected 19 IoT devices that belong to each IoT platform and cost less money (e.g. smart bulbs and smart plug). Note that although the devices we choose are preferred in terms of the cost, a vendor's expensive devices (e.g. thermostats and dehumidifiers) and cheap devices share the same backend cloud.

In Table 1, we show the statistics of the 19 evaluated platforms. As can be seen in the third column, each IoT platform supports 12 different products on an average. The products include plugs, bulbs, cameras, heaters, ovens, detectors etc. From the fourth column and the fifth column, we can see that about 84% of the platforms allow users to share their devices within IoT systems, and all of the 19 platforms support third-party integration. In this study, we study three popular third-party services: Google Assistant, Amazon Alexa and IFTTT. Google and Amazon (AWS) are cloud service providers while IFTTT is a web service that can integrate various web applications.

[2]Samsung's third-party voice service Bixby [33] is still under development and has not been widely supported yet. Therefore we did not evaluate it.

TABLE 1 Statistics of the evaluated platforms

IoT platforms	Evaluated device	# of types	Device sharing	Third-party control?
Arlo	Camera	16	✓	✓
Belkin	Plug	4	✓	✓
BroadLink	Remote controller	9	✓	✓
DoHome	Bulb	8	✓	✓
elinkSmart	Camera	26	✓	✓
ETEKCITY	Bulb	11	✓	✓
eWeLink	Plug	15	✓	✓
EZVIZ	Camera	15	✓	✓
Koogeek	Plug	64	✓	✓
LIFX	Bulb	2	×	✓
MagicHome	Bulb	4	✓	✓
Meross	Plug	10	×	✓
Philips Hue	Bulb	3	✓	✓
Sengled	Bulb	6	✓	✓
SmartLife	Plug	14	✓	✓
Smartbulb	Bulb	1	✓	✓
Topgreener	Plug	17	✓	✓
Kasa	Plug	7	×	✓
WeConn	Plug	5	✓	✓

4.1.3 | Experiment setup and procedure

We run the experiments with the IoT devices and two Android-based Nexus 6 smartphones. One smartphone acts as the victim's smartphone. The other acts as the attacker's. The companion IoT mobile apps and third-party mobile apps (i.e. Google Home, Amazon Alexa, and IFTTT) are installed on both smartphones. Each of the attacker and the victim has an account in the IoT app and the third-party app. For the network connection, at first, the smartphones are connected to the local network when we configure the IoT device. After device configuration, we connect the smartphones to the Internet using cellular data to ensure that they do not affect each other in the communication channel. For the configuration of third-party apps, setting up Google Assistant and Amazon Alexa in their apps is relatively straightforward and we do not elaborate on the details. For IFTTT, we search the name of the IoT vendor and choose a connection/an applet that can control the IoT device.

4.2 | Experimental results

4.2.1 | Overall results

We give the experimental results of the IoT platforms in Table 2. On the whole, there are 8 vulnerable platforms. Specifically, there are 6 platforms suffering from the vulnerability that occured in device sharing and 8 platforms suffering from the vulnerability in owner transfer. We found that neither vulnerabilities do not occur in the platform 'LIFX' and 'EZVIZ', as they do not support guest users' third-party authorisation. Preventing third-party authorisation after device sharing avoids the problem of authorisation inconsistency but limits the functionality of IoT applications. On the other hand, the platforms 'Arlo', 'Meross', and 'Kasa' do not suffer from the vulnerability in device sharing, as they do not support device sharing. Besides, the results are consistent in all three third-party services: Google Assistant, Amazon Alexa and IFTTT (as shown in Table 2, there are some exceptions in IFTTT because we could not find any applets of the vendor to trigger the device control, such as 'DoHome', 'eLinkSmart' and 'WeConn'). The consequence of the vulnerabilities is serious: attackers who once *temporarily* owned the control permission of an IoT device can permanently and stealthily take control of that device.

In the attacks, the leaked permissions are determined by the functionalities of the device. As we observed in the evaluation, the attacker can use major functionalities of the devices. We also measured the validity period of the attacks. The experiments show that, for all vulnerable platforms, the third-party services hold the control permission for at least 60 days during which we conducted the experiments. Such a period would last even longer, as third-party services

TABLE 2 Experimental results

IoT platforms	Device sharing attack w. Google/Alexa/IFTTT	Ownership transfer attack w. Google/Alexa/IFTTT	Leaked permissions	Validity period of attack
Arlo	O/O/O	✓/✓/✓	Camera view	>60 days
Belkin	×/×/×	×/×/×	O	O
BroadLink	×/×/×	×/×/×	O	O
DoHome	×/×/O	×/×/O	O	O
elinkSmart	×/×/O	×/×/O	O	O
ETEKCITY	✓/✓/✓	✓/✓/✓	Turn on/off	60 days
eWeLink	×/×/×	×/×/×	O	O
EZVIZ	O/O/O	×/×/×	O	O
Koogeek	✓/✓/O	✓/✓/O	Turn on/off	>60 days
LIFX	O/O/O	×/×/×	O	O
MagicHome	×/×/×	×/×/×	O	O
Meross	O/O/O	✓/✓/✓	Turn on/off	>60 days
Philips Hue	×/×/×	×/×/×	O	O
Sengled	×/×/×	×/×/×	O	O
SmartLife	✓/✓/✓	✓/✓/✓	Turn on/off	>60 days
Smartbulb	✓/✓/O	✓/✓/O	Turn on/off	>60 days
Topgreener	✓/✓/✓	✓/✓/✓	Turn on/off	>60 days
Kasa	O/O/O	×/×/×	O	O
WeConn	✓/✓/O	✓/✓/O	Turn on/off	60 days

Note: ✓: vulnerable to the attack; ×: fail to launch the attack; O: not applicable.

recommend IoT developers to guarantee a long-lived access token and recommend such a token never expires [16].

4.2.2 | Responsible disclosure

We had made responsible disclosure to each of the affected IoT providers. At the time of writing this version, as we confirmed, all of the IoT providers have fixed the reported vulnerabilities. Interestingly, there is a provider that claims it provides strong protection for the platform by following international security standards and industry requirements. Our study shows that although serious efforts had been taken to protect the platform, the risk of authorisation inconsistency still existed.

5 | DISCUSSION ON AUTHORIZATION INCONSISTENCY

In this section, we first discuss existing designs and fundamental causes of authorisation inconsistency. Then we propose principles that should be followed in the secure design of the IoT control framework.

5.1 | Existing designs

As multiples parties are involved, much complication is added to the authorisation process. In this study, the presence of the vulnerabilities implies the weakness in IoT's design philosophy and its ecosystem: the authorisation schemes in cross-cloud IoT environments are heterogeneous. Case-by-case fixation does not help to address the fundamental cause that there lacks secure design principles and standards for IoT third-party integration. As demonstrated in previous sections, serious consequences are caused if the expected constraints are violated. Besides, there are a few IoT platforms that do not support control permission transfer to a third-party service when devices are shared, as shown in the evaluation. Thus, we observe and discuss two types of design choices that result in authorisation inconsistency:

- **Non-transitive delegation control.** As we observed in the empirical measurement, this kind of delegation principle only allows one delegation operation at a time and does not permit any delegation chain (i.e. a delegatee can authorise a new delegatee). Although this avoids security risks of

authorisation complication, the functionalities of IoT applications are restricted. Such a non-transitive delegation control scarifies usability for security and therefore is not preferable in practice.

- **Inconsistent delegation control.** The fundamental cause of the presented attacks is the lack of a monolithic design/standard to regularise cross-cloud delegation in IoT environments. Today's IoT clouds adopt heterogeneous and customised management of authorisation in the internal control framework. The combination of the ad hoc authorisation protocol and the standardized authorisation protocol—OAuth—introduces in-compliance and results in error-prone policy enforcement.

5.2 | Secure design principles

For the purpose of mitigating the risk and safeguarding the control framework of the IoT cloud, we propose the following secure design principles to guide the design of authorisation in cross-cloud delegation of the IoT and educate developers:

- **Secure principle of device sharing.** When the control permission of the device is propagated from the authorising user to the delegatee, the delegatee's authorisation actions (e.g. authorising to another delegatee) should be absolutely contained and under the original authorising user's control.
- **Secure principle of ownership transfer.** Whenever the device's ownership is transferred, any new authorisation actions (including reset operations) of the device should be under the previous owner's permission. After the previous owner's grant of ownership transfer, all permissions of the transferred device (including third-party authorisation) should be completely transferred to the new owner.

We suggest to explicitly visualise all the granted permissions on the user's app, to facilitate authorising users to manage their delegatees.

5.3 | Future direction

Notably, we could have further extended this research, for example, by developing automated verification tools to systematically identify this type of attack. It requires us to automate the experimental procedures in order to reach vulnerable authorisation states. To this end, the first option is to directly operate physical devices and interact with the IoT cloud to achieve the vulnerable authorisation states. However, manual setup is unavoidable in the process. The second option is to directly manipulate the requests to the IoT cloud. The difficulty lies in how to test the revocation of third-party control through manipulation of unattainable cloud-to-cloud (i.e. the IoT cloud and the third-party cloud) communication traffic. We leave it as our future work.

6 | RELATED WORK

Considerable research has been devoted to authorisation and the IoT. In this section, we show that the unique insight in our work is different from existing studies.

6.1 | Coarse-grained authorisation of the IoT

Some research studies have explored the coarse-grained authorisation of IoT [3–7, 17–23], including the security analysis of the IoT platforms and the proposals of fine-grained authorisation mechanisms. In particular, by examining the source code of cloud-side applications (i.e. SmartApps), Earlence et al. [3] found that the coarse-grained capabilities of SmartApps could lead to several attacks. The root cause is that the Samsung IoT platform grants full access to the SmartApps even if they only require limited permissions. On the other hand, several new schemes were proposed to provide more fine-grained authorisation for IoT, such as SmartAuth [21], ContexIoT [5] etc [4, 6, 7]. These studies have emphasised the designs of authorisation schemes within the IoT platforms, instead of focussing on the authorisation inconsistency with the integration of third-party services.

6.2 | Security analysis of IoT third-party services

Another direction is to perform security analysis for IoT third-party services themselves [24], such as trigger-action platforms and voice assistants. Milijana Surbatovich [25] conducted the first in-depth analysis of security impacts on trigger-action platforms (e.g. IFTTT and Zapier) that can be brought to common users. Their work indicates that 50% of the wild recipes in the IFTTT platform can be unsafe as they either violate the integrity or secrecy. Earlence Fernandes [26] focussed on the security risk of trigger-action platforms and introduced a decentralised security principle to prevent the misuse of OAuth tokens once the trigger-action platforms get compromised. In the context of voice assistants, research has tended to focus on the misinterpretation of natural languages [27] and acoustic-based attacks [28]. Those works have explored privacy issues, insecure architectures, and security threats of the voice for third-party services. However, they do not address insecure authorisation.

6.3 | Insecure authorisation in the cloud

There were only a few past works that focussed on insecure authorisation in the cloud [29–31]. In most cases, online services are provided by back-end cloud servers. To promise a secure access, back-end servers must implement a considerable and robust access control mechanism. It then becomes a critical task to analyse whether there exist any security risks in the

authorisation parts of a system. For this purpose, Zuo et al. [1] proposed a mobile-based tool to automatically pinpoint vulnerable access control implementations through differential analysis. However, the tool only targets at shallow authorisation problems such as whether the authorisation token is bound with the user ID in the cloud and whether the user ID contains sufficient randomness. On the other hand, cloud-side logic flaws in multi-party interaction have drawn attention of the research community. For instance, Yang et al. [32] proposed a symbolic reasoning-based tool to automatically check the correctness and vulnerabilities of SSO SDKs in the cloud applications. Their experiment shows that popular cloud-side SDKs contain different logic flaws. However, those works do not address the IoT specific authorisation problems that exist in third-party integrated IoT systems.

7 | CONCLUSION

In this study, we report our research on a new type of attack —authorisation inconsistency in IoT third-party integration, which is specific to IoT systems and has never been studied before. The vulnerability allows the attacker to exploit third-party services of IoT and take unauthorized, permanent, and stealthy control of IoT devices. To investigate the pervasiveness of the problem, we evaluated 19 popular IoT platforms and discovered that 8 of them are vulnerable to the new attack vector. We then present in-depth discussion and propose two secure principles to educate developers: (1) When the control permission of the device is propagated from the authorising user to the delegatee, the delegatee's authorisation actions (e.g. authorising to another delegatee) should be absolutely contained and under the original authorising user's control; (2) Whenever the device's ownership is transferred, any new authorisation actions (including reset operations) of the device should be under the previous owner's permission. After the previous owner's grant of ownership transfer, all permissions of the transferred device (including third-party authorisation) should be completely transferred to the new owner. Our research takes the first step towards understanding insecure authorisation in the integration of IoT third-party services and brings to light the significance of this security risk.

ACKNOWLEDGMENTS

This work was partially supported by the Hong Kong S.A.R. Research Grants Council (RGC) Early Career Scheme/General Research Fund No. 14208019 and No. 14208818.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in [zenodo] at https://doi.org/10.5281/zenodo.5501202.

ORCID

Jiongyi Chen https://orcid.org/0000-0003-0776-4073

REFERENCES

1. Zuo, C., Zhao, Q., Lin, Z.: Authscope: towards automatic discovery of vulnerable authorizations in online services. In: ACM Conference on Computer and Communications Security, Dallas (2017)
2. Owasp: Mobile Top 10 2016-m6 Insecure Authorization. https://owasp.org/www-project-mobile-top-10/2016-risks/m6-insecure-authorization
3. Fernandes, E., Jung, J., Prakash, A.: Security analysis of emerging smart home applications. In: IEEE Symposium on Security and Privacy (SP), pp. 636–654. IEEE, San Jose (2016)
4. He, W., et al.: Rethinking access control and authentication for the home internet of things (iot). In: 27th USENIX Security Symposium (USENIX Security 18). pp. 255–272. USENIX Association, Baltimore (2018)
5. Jia, Y.J., et al.: Contexlot: towards providing contextual integrity to appified iot platforms. In: NDSS. San Diego (2017)
6. Lee, S., et al.: Fact: functionality-centric access control system for iot programming frameworks. In: Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies, pp. 43–54. Indianapolis (2017)
7. Rahmati, A., et al.: Tyche: Risk-Based Permissions for Smart Home Platforms. IEEE Secure Development Conference, Cambridge (2018). arXiv preprint arXiv:1801.04609
8. Ovadia, S.: Automate the internet with "If This Then That"(IFTTT). Behav. Soc. Sci. Libr. 33(4), 208–211 (2014)
9. Ur, B., et al.: Trigger-action programming in the wild: an analysis of 200,000 ifttt recipes. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp. 3227–3231. San Jose (2016)
10. Amazon alexa. https://developer.amazon.com/alexa
11. Google assistant. https://assistant.google.com/smart-home/
12. Ifttt helps your apps and devices work together. https://ifttt.com/
13. Tuya Platform for Smart Home. https://www.tuya.com/
14. Internet of Things (iot) Markets: A Global Outlook to 2023 - Emergence of ai and Smart Home Devices, and Innovations in Sensors Technologies Support Growth. https://www.businesswire.com/news/home/20190501005556/en/Internet-Things-IoT-Markets-Global-Outlook-2023
15. Google Play App Store. https://play.google.com
16. Account Linking with Oauth. https://developers.google.com/actions/identity/oauth2?oauth=code
17. Balliu, M., Bastys, I., Sabelfeld, A.: Securing iot apps. IEEE Security & Privacy. 17(5), 22–29 (2019)
18. Ouaddah, A., Abou Elkalam, A., Ouahman, A.A.: Towards a novel privacy-preserving access control model based on blockchain technology in iot. In: Europe and MENA Cooperation Advances in Information and Communication Technologies, pp. 523–533. Springer, Saidia (2017)
19. Ouaddah, A., et al.: Access control in the internet of things: big challenges and new opportunities. Comput. Network. 112, 237–262 (2017)
20. Rahman, A.F.A., Daud, M., Mohamad, M.Z.: Securing sensor to cloud ecosystem using internet of things (iot) security framework. In: Proceedings of the International Conference on Internet of things and Cloud Computing, vol. 1–5. Cambridge (2016)
21. Yuan, T., et al.: Smartauth: User-centered authorization for the internet of things. In: 26th USENIX Security Symposium (USENIX Security 17).
22. Vasilomanolakis, E., et al.: On the security and privacy of internet of things architectures and systems. In: 2015 International Workshop on Secure Internet of Things (SIoT), pp. 49–57. IEEE, Vienna (2015)
23. Yan, H., et al.: Iot-fbac: function-based access control scheme using identity-based encryption in iot. Future Generat. Comput. Syst. 95, 344–353 (2019)
24. Kim, J., et al.: Standard-based iot platforms interworking: implementation, experiences, and lessons learned. IEEE Commun. Mag. 54(7), 48–54 (2016)
25. Surbatovich, M., et al.: Some recipes can do more than spoil your appetite: analyzing the security and privacy risks of ifttt recipes. In: WWW. Perth (2017)
26. Fernandes, E., et al.: Decentralized action integrity for trigger-action iot platforms. In: NDSS. San Diego (2018)

27. Zhang, Y., et al.: Life after speech recognition: fuzzing semantic misinterpretation for voice assistant applications. In: NDSS. San Deigo (2019)
28. Zhang, G., et al.: Dolphinattack: inaudible voice commands. In: ACM Conference on Computer and Communications Security, Dallas (2017)
29. Barcena, M.B., Wueest, C.: Insecurity in the internet of things, Security Response, Symantec, 20 (2015)
30. Nelson, N.M., et al.: A framework for authentication and authorization credentials in cloud computing. In: 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, pp. 509–516. IEEE, Melbourne (2013)
31. Whaley, J., Purtell, T.J. II & Thomas, G.G. U.S. Patent: Protecting data in insecure cloud storage.
32. Yang, R., et al.: Vetting single sign-on implementations via symbolic reasoning. In: 27th USENIX Security Symposium (USENIX Security 18), pp. 1459–1474. Baltimore (2018)
33. Samsung bixby: Your personal voice assistant. https://www.samsung.com/us/explore/bixby/

How to cite this article: Chen, J., et al.: Authorisation inconsistency in IoT third-party integration. IET Inf. Secur. 16(2), 133–143 (2022). https://doi.org/10.1049/ise2.12043