







OAuth-Based Access Control Framework for IoT Systems

Min-Zheng Shieh¹ , Jui-Chun Liu¹ , Yi-Chih Kao¹ , Shi-Chun Tsai² ,
and Yi-Bing Lin² 

¹ Information Technology Service Center, National Chiao Tung University,
Hsinchu, Taiwan

{mzshieh,g0737,ykao}@nctu.edu.tw

² Department of Computer Science, National Chiao Tung University,
Hsinchu, Taiwan

{sctsai,liny}@cs.nctu.edu.tw

Abstract. With the emergence of the Internet of Things (IoT) technology, the number of related devices has been increasing at a very rapid speed. The security of IoT systems has become a crucial issue. Due to the complex IoT environment and users' unawareness, such issues are usually hard to resolve. Many IoT systems lack proper access control mechanisms and suffer from various large scale attacks. We need a robust and effective secure access control to build IoT systems that retain user privacy and data integrity with high availability.

In this paper, we propose an access control framework based on OAuth 2.0, with which we constructed a remote control system for various devices. The secured authentication schemes prevent possible private data leaks. The proposed framework provides flexibility for further functional extensions with new IoT devices.

Keywords: Access control · Authentication · Internet of Things · OAuth

1 Introduction

An integrated system with IoT devices often consists of embedded sensors, actuators, communication hardware, and software. The devices collect various data from the built-in sensors and generally communicate over a heterogeneous network. IoT systems add much more value to their hardware by providing applications in automation, artificial intelligence, etc. However, IoT systems have many inherent security threats and challenges due to their system scale [1], the heterogeneity of networks [2], various communication protocols [3], and the lack of proper access control [4]. Open Web Application Security Project (OWASP)

This work was financially supported by the Center for Open Intelligent Connectivity from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

[5] lists the “top 10 things to avoid when building, deploying, or managing IoT systems” to show adversaries in diverging aspects. It is a crucial issue to create IoT systems against various attacks.

Khan et al. [2] pointed out one essential characteristic of IoT devices is the limit of resources, such as electricity, computing power, and memory. When the adversary attacks an IoT system, the devices consume much more and may deplete certain kinds of resources. Without proper security control, there are risks of system meltdown. Chiang et al. [6] and Cirani et al. [7] showed that it is unrealistic to deploy complex access control protocol or to perform encryption algorithms on IoT devices since the resources are too limited. There are practical needs for simple access control protocols and lightweight authentication mechanisms.

Chiang et al. [6] suggested that many IoT devices may have a long life cycle, but the threats will become more advanced with the development of new technology. Therefore, we should keep enhancing IoT systems against new threats. In Sicari et al. [8] and Anggorojati et al. [9], they both addressed that IoT applications must satisfy security and privacy requirements. After all, IoT devices pervasively involve human life. To protect privacy, we need proper access control and authentication mechanism to prevent unauthorized accesses of sensitive personal information.

In a pilot project at National Chiao Tung University, we have a dormitory [10] installed with several IoT systems, including laundry machines, drying machines, air conditioners. These systems are provided by different suppliers and developed with various protocols, including WiFi, TCP/IP, HTTP, TaiSEIA [11], etc. Moreover, none of the system providers integrates their system with an authentication system. To ensure proper access to the tenants, we have to incorporate these IoT systems into the smart campus system. In this paper, we focus on secure access control and privacy protection issues. We create a framework to integrate IoT systems with the campus authentication system via the OAuth 2.0 protocol.

The OAuth 2.0 protocol allows information systems to access the private data authenticated by the users. For example, the user may grant access to their room number to an agent software. The agent will provide access to the air conditioner controller system in the private networks and configure the air conditioner at the users’ demand. With OAuth 2.0, we can even implement application without revealing information to client software, such as smartphone apps and web browsers. OAuth server synchronizes the user data with the other databases in the data center. Thus, there is no need to directly modify the architectures of IoT systems provided by the vendors.

Based on the proposed framework, we develop a universal remote control platform for smart home devices. Users may access the IoT devices properly via smartphone apps or web-based graphical user interface (GUI). We create an agent to deliver users’ commands to the home appliances, such as air conditioners in a room. Developers can compose their client software over the HTTPS protocol and OAuth 2.0. There is no need to involve any underlying heterogeneous communications to IoT devices. The framework also creates a barrier between

the users and the IoT systems. Therefore, we allocate private IP addresses to the IoT devices or their gateway. With such configuration, we prevent all direct communication and attacks from the public networks and among different IoT subsystems. Thus, the adversaries can only launch cyberattacks to the IoT devices in the same private networks, which are much harder to compromise.

Our approach is flexible to integrate new IoT subsystems into the existing systems. Adding an IoT subsystem means creating an agent to access it via its application programming interface (API) and the authentication system via the OAuth 2.0 protocol. Without modifying the IoT subsystem, it is much easier to maintain the software. We can also apply the framework for the smart appliance system in a residential facility. The facility will grant access to the tenants upon their check-in, and the facility will revoke their access rights when they check-out.

This paper is organized as follows. Section 2 reviews related background and works. Section 3 illustrates the proposed access control mechanism for IoT systems with a working example. Section 4 shows the security of the implementation. At last, we discuss future works and briefly conclude this paper in Sect. 5.

2 Background and Related Works

OAuth 2.0, often called OAuth in the following, is an open standard for open authorization. It enables third-party applications to obtain limited access to private data, without having to provide the access permission to third-party applications directly. RFC6749 [13] defines the roles of the OAuth authorization flow as follows.

- The resources owner authorizes the client to access the private data in the resource server.
- The authorization server is responsible for issuing an access token after obtaining authorization from the resource owner.
- The resource server is responsible for storing private data, allowing the clients to access private data according to the access token.
- The client accesses private data on behalf of the resource owner.

Many websites have adopted authorization and single sign-on (SSO) in recent years, with OAuth 2.0 being one of the most popular frameworks. Fett et al. [12] summarized that the prime identity providers, including Amazon, Facebook, Google, and GitHub, use OAuth 2.0. Being one of the most popular SSO systems on the web, OAuth enables billions of users to log in at millions of services and to authorize selected private data to applications.

There are many existing research results about IoT applications based on OAuth. Emerson et al. [14] utilized OAuth to provide a secure authentication mechanism for the IoT network. The security manager efficiently manages the database with a list of authorized users who can access the IoT network. In this approach, only authenticated users are allowed to access the IoT network,

protecting the IoT network from unauthenticated users. However, this research does not present any testing and verification results. Performance issues may exist when applied to complex network environments. Cirani et al. [7] proposed an approach targeting HTTP/CoAP services to provide an authorization framework by invoking an external OAuth-based authorization service. The proposed framework can be integrated with IoT scenarios to provide security assurance for IoT. The research by Fremantle et al. [15] combines OAuth with MQTT to let OAuth as a part of the MQTT protocol flow and within an MQTT broker, making federated and user-directed control decisions. Siris et al. [16] presented models for utilizing blockchain and smart contract technology with the widely used OAuth open authorization framework to provide delegated authorization for constrained IoT devices.

3 Proposed Framework

In this paper, we propose an access control mechanism for the IoT remote control system in a dormitory based on the integration of OAuth 2.0 and user databases. With proper access control, we ensure that the connection between user devices and IoT devices is secure. OAuth authorizes legitimate third-party applications to access private data to ensure each tenant can only control corresponding IoT devices in their room after authentication. We had constructed the IoT remote control system for the air conditioner in a dormitory allowing users to configure air-conditioning functions and check IoT device status.

3.1 Architecture

Figure 1 depicts the architecture of the proposed access control framework for IoT systems. The main components include User Device, IoT Agent, IoT Subsystem, OAuth Server, and Database. In particular, IoT Agent, OAuth Server, and Database are located in the data center, whereas IoT Subsystem is in the private network.

User Device (Fig. 1a) could be any device that has access to the IoT remote control mobile apps or web browsers, such as smartphones, laptops, or any other mobile devices. The IoT remote control apps or web browsers provide web-based GUI for users to interact with the devices in the IoT Subsystem (Fig. 1c) via IoT Agent (Fig. 1b). IoT Agent resides in a virtual machine installed in the data center that systematically manages the features of the IoT Subsystem and allows users to access the IoT Subsystem properly. IoT Agent and IoT Subsystem interact via Application Programming Interface (API). Authenticated users can read the sensors and configure the IoT devices through the IoT Agent, which redirects the client software to the OAuth Server (Fig. 1d) to initiate the authorization code grant flow for the first access. After that, the User Device receives a token for accessing the IoT Subsystem.

User Device may communicate with the IoT agent and OAuth Server over public networks. The IoT subsystem can only interact with the IoT Agent by

using its API. The separation guarantees that IoT Subsystem is immune from cyberattacks outside of its private network unless IoT Agent is compromised. For user privacy protection, the OAuth protocol allows users to decide which data in the database (Fig. 1e) is accessible by the IoT Agent. With the proper design of IoT Agent, we can control what data can be revealed to the IoT subsystem and the client software.

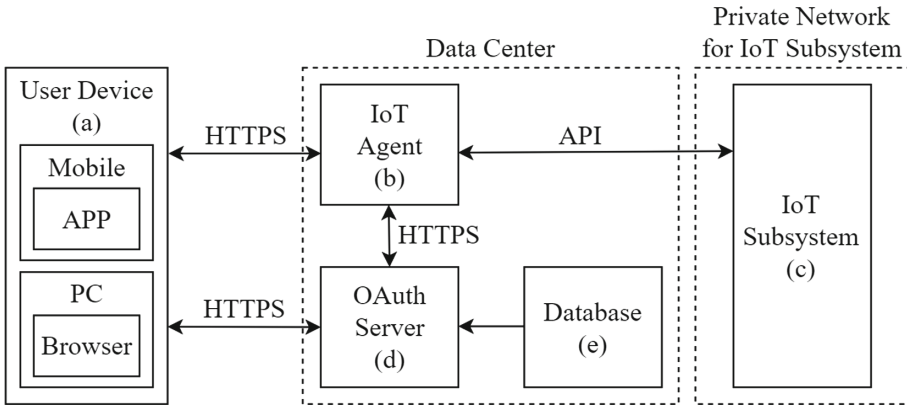


Fig. 1. Access control framework for IoT systems

3.2 Use Case

In this subsection, we show a use case of the proposed framework. We construct a remote control system for the air conditioners (Fig. 2) in the dormitory. Here, we use NCTU OAuth service (Fig. 2g) and Resident Database (Fig. 2h) to provide the user information. The air conditioner IoT subsystem consists of an IoT Gateway (Fig. 2c), one hundred Access Points (Fig. 2d), and two hundred air conditioners (Fig. 2f). We install an IoT Dongle (Fig. 2e) to each air conditioner.

The API used for the interactions between the IoT Agent and the IoT Gateway is based on WebSocket. The IoT Agent retrieves the information of the air conditioner system from the IoT Gateway, where the information contains underlying devices' name and type, managed device list, connected device list, device configurations, and sensor values. We can configure the air conditioners through the WebSocket API. Since IoT Agent itself maintains the information of the IoT subsystem and user binding tables, the status of each IoT Dongle will be updated simultaneously when the IoT Dongle connects to the IoT Gateway. The customized IoT Agent provides HTTP-based RESTful APIs for User Device to display or operate the air conditioner on the IoT remote control apps or web browsers.

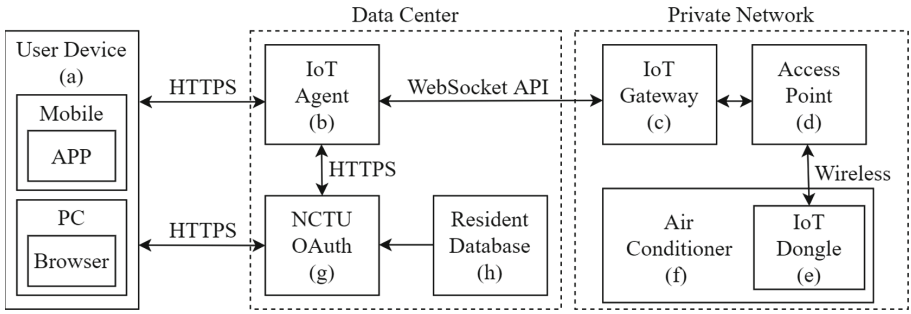


Fig. 2. Remote control system for air conditioners

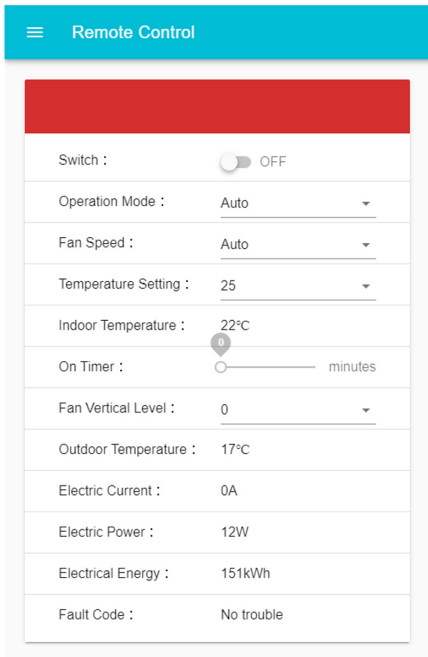


Fig. 3. IoT remote control web GUI



Fig. 4. IoT remote control app GUI

IoT Dongle is embedded in the air conditioner to control air-conditioning functions and possesses a control module that supports the TaiSEIA [11] protocol enabling smart appliances to be interconnected. This protocol features cross-vendor management that solves the vendor-dependent issue, providing optimized flexibility for IoT device management. IoT Dongle establishes Wi-Fi Protected Access (WPA/WPA2) connection to wireless Access Point (AP) (Fig. 2d) in each room via Wi-Fi, and the IoT Gateway gathers the traffic from each room’s AP. After the connection is completed, IoT Dongle then establishes Transport

Layer Security (TLS) connection to IoT Gateway. By using secure Transmission Control Protocol (TCP), IoT Gateway sends control commands to IoT Dongle to perform air-conditioning functions. The IoT remote control web and app GUIs are illustrated in Fig. 3 and Fig. 4, users can adjust the ON/OFF switch, operation mode, fan speed, temperature, and can also monitor the indoor and outdoor temperature. With the GUIs, users can conveniently monitor the air conditioner's real-time status and configure control functions through the smart-phone app or web browser. Since we deploy the air conditioner subsystem in the designated private network, all direct traffic from public networks are prevented. Thus, we keep the IoT system away from cyberattacks initiated in public networks.

NCTU OAuth (Fig. 2g) is an open authorization system developed based on the OAuth 2.0 protocol that integrates user databases. It is up to the users to determine whether their data is authorized for use by third-party applications, providing third-party applications the data required to verify user identity. NCTU OAuth is responsible for bridging the User Device, IoT Agent, and Resident Database (Fig. 2h), which stores resident information and room information, including a binding table of room number to resident information. User Device authorizes IoT Agent to access user's private data in Resident Database through NCTU OAuth without having to provide the database access permission to IoT Agent. The whole OAuth authorization flow uses HTTPS RESTful API for communication and data exchange, and we will describe the detailed authorization flow in the following.

3.3 Authorization Code Grant Flow

RFC6749 [13] defines four grant types, including authorization code, implicit, resource owner password credentials, and client credentials. The NCTU OAuth adopts the authorization code grant, and the grant flow works as Fig. 5 shows:

- (a) The user initially logs in NCTU OAuth by entering the username and password on User Device.
- (b) NCTU OAuth returns an authorization code and a redirected URL according to the required scope.
- (c) The client software on User Device sends the authorization code to the IoT Agent after redirecting.
- (d) IoT Agent uses the authorization code provided by the User Device to exchange an access token from NCTU OAuth via a POST request.
- (e) NCTU OAuth issues an access token and returns to the IoT Agent.
- (f) IoT Agent uses the access token to access device information and user information (e.g., room number and resident information) from NCTU OAuth via a GET request.
- (g) NCTU OAuth returns device information and user information.
- (h) IoT Agent sends a SESSION_TOKEN to User Device for further authentication.

- (i) User Device gets the managed device list and device configurations via the SESSION_TOKEN.
- (j) IoT Agent responses data.

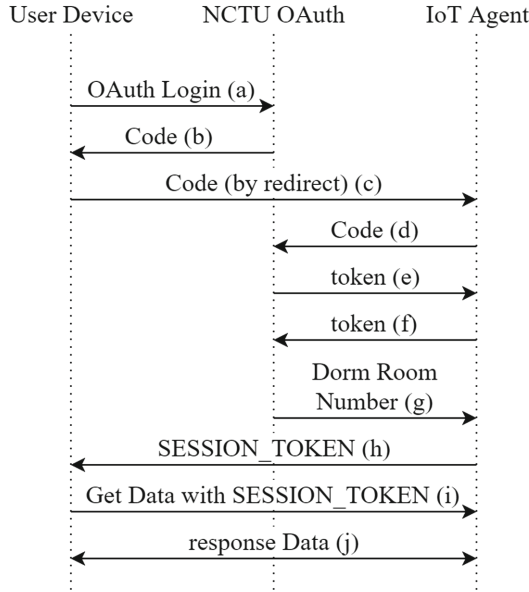


Fig. 5. Authorization code grant flow

Since NCTU OAuth only authenticates the identity of User Device, any other clients within the proposed architecture cannot access the information of User Device, ensuring confidentiality for the network environment and preventing private data leakage.

3.4 Summary

IoT Agent maintains a binding table associating each room to the devices under management. Therefore, when IoT Agent retrieves the binding table and resident information from NCTU OAuth, the devices in the room match with the room’s tenant. In this way, the User Device can effectively and safely communicate with the IoT Dongle. It is easy to reconfigure the devices with the binding table and readjust the binding when the tenants move.

This paper proposes an access control framework for the IoT system based on OAuth. This framework only allows valid users to access resources and eliminates malicious adversaries and attackers. Authenticated users are limited in operating the devices they have the permission to; they can not tamper with other IoT devices deliberately, thereby ensuring the integrity of the data and enhancing the security of the network.

4 System Security and Discussion

4.1 Vulnerability Scanning

Even though we allocate private networks for the IoT system, which prevents cyberattacks from public networks, potential internal threats within private networks still exist. We simulate internal attacks by exploiting port scanning to find possible weaknesses and vulnerabilities of IoT devices.

Port scanning is an approach to obtain the device operating system (OS) version, software version, service ports, and other information. Through this information, we can collect possible vulnerabilities of the target device. In this test, we connected a Raspberry Pi with Kali Linux to the IoT Gateway. We used Nmap [17] tool to perform port scanning on the IoT dongle to see if the IoT dongle opens up some service ports that are vulnerable to security attacks. As shown in Fig. 6, we used the `nmap -p 1-65535 192.168.100.4` command to perform port scanning to diagnose the operating system and various services of the target IoT dongle, only to find that all 65,535 scanned ports were filtered. Again, we used the `nmap -A 192.168.100.4` command to probe the OS information of the hosts that are mapped, as illustrated in Fig. 7. However, Nmap cannot interpret the OS information of the IoT dongle. As a result, the possibility of malicious attacks by learning the hosts' weaknesses through vulnerability scanning is reduced.

```

root@kali:~# nmap -p1-65535 192.168.100.4
Starting Nmap 7.80 ( https://nmap.org ) at 2019-12-24 06:19 UTC
Nmap scan report for 192.168.100.4
Host is up (0.0030s latency).
All 65535 scanned ports on 192.168.100.4 are filtered
MAC Address: 9E: :3B (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1328.33 seconds

```

Fig. 6. Scanning all 65,535 ports using nmap -p 1-65535

4.2 Security Testing

Unlike many previous works, we send the implemented remote control system and its mobile app to a third party for verifying their security. Chunghwa Telecom Laboratories Testing Center tested the system with schemes based on “Basic Security Testing Baseline for Mobile Applications V3.0” [18] defined by the Industrial Development Bureau, Ministry of Economic Affairs, Taiwan. The security of the IoT remote control app and its back-end server is tested and certified by Chunghwa Telecom Laboratories Testing Center. The tests include but not limited to the following items.

```

root@kali:~# nmap -A 192.168.100.4
Starting Nmap 7.80 ( https://nmap.org ) at 2019-10-24 15:37 UTC
Nmap scan report for 192.168.100.4
Host is up (0.0039s latency).
All 1000 scanned ports on 192.168.100.4 are filtered
MAC Address: 9E: [REDACTED]:3B (Unknown)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   3.91 ms  192.168.100.4

Nmap done: 1 IP address (1 host up) scanned in 43.77 seconds

```

Fig. 7. Scanning the OS information using namp -A

1. Authentication and authorization

- The app offers an appropriate authentication mechanism to identify the user.
- The app authorizes users by user’s identity.
- The app prevents users from entering common injection strings.

2. Transmission security

- The app securely encrypted transmitted private data by using an appropriate and effective key length of the encryption algorithms.
- The app avoids connecting and transferring private data to servers without a valid certificate.
- The app avoids using regular SESSION_TOKEN, and the SESSION_TOKEN does not relate to time, regular numbers or strings, or anything related to the user’s submission.

The implemented system fully complies both the security baselines for mobile applications and server-side defined in “Basic Security Testing Baseline for Mobile Applications V3.0”. It means that the systems implemented under the same framework are likely to provide the same security and pass the same test.

4.3 Discussion

In previous subsections, we showed that the implementation of the remote control system for the dormitory air conditioner system, including the back-end and the client software, is resilient to major cyberattacks. The framework can isolate every single integrated IoT subsystem in a private network so that we can provide the same security to any new integrated IoT subsystem.

The proposed framework is not only suitable for applications on campus IoT systems. Some facilities, like hotels and fitness centers, have similar needs. Hotels may install several smart home appliance systems in the guest rooms and provide a mobile app as a remote controller for the guests. Nowadays, hotel chains like

Marriott International and Hilton Worldwide have their account system for their members. Using our proposed framework, one can build a remote control system with the authentication system without much effort. The guests may control the IoT devices in the room with their mobile phones upon check-in, and the system automatically revokes their access right after they check out.

5 Conclusion

In this paper, we proposed an OAuth-base framework for integrating IoT systems on campus or a similar facility. The framework enables us to use the campus authentication system to construct an access control system for the IoT systems; even the vendors do not provide such a function. The framework reduces the security risks by separating IoT subsystems into independent private networks. For privacy, the client software may only access the necessary information by using the OAuth authorization code grant flow.

We specifically built the remote control system for the air conditioners with the proposed framework. With the performance test results, we have shown that the remote control system is efficient. For security proof, it passed the security tests and obtained the certificate from Chunghwa Telecom Laboratories Testing Center. We are seeking new applications on more smart IoT devices and opportunities on similar facilities, such as chained hotels and fitness centers.

References

1. Andy, S., Rahardjo, B., Hanindhito, B.: Attack scenarios and security analysis of MQTT communication protocol in IoT system. 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), pp. 1–6. IEEE, Yogyakarta, Indonesia (2017)
2. Khan, M.A., Salah, K.: IoT security: review, blockchain solutions, and open challenges. *Future Gener. Comput. Syst.* **82**, 395–411 (2018)
3. Ojo, M., Adami, D., Giordano, S.: A SDN-IoT architecture with NFV implementation. In: 2016 IEEE Globecom Workshops (GC Wkshps), pp. 1–6. IEEE, Washington, DC, USA (2016)
4. Ouaddah, A., Mousannif, H., Abou Elkalam, A., Ouahman, A.A.: Access control in the Internet of Things: Big challenges and new opportunities. *Comput. Netw.* **112**, 237–262 (2017)
5. OWASP Internet of Thing Top 10. <https://owasp.org/www-project-internet-of-things/>. Accessed 10 Aug 2020
6. Chiang, M., Zhang, T.: Fog and IoT: an overview of research opportunities. *IEEE Internet Things J.* **3**(6), 854–864 (2016)
7. Cirani, S., Picone, M., Gonizzi, P., Veltri, L., Ferrari, G.: Iot-oas: an oauth-based authorization service architecture for secure services in iot scenarios. *IEEE Sens. J.* **15**(2), 1224–1234 (2014)
8. Sicari, S., Rizzardi, A., Grieco, L.A., Coen-Porisini, A.: Security, privacy and trust in Internet of Things: the road ahead. *Comput. Netw.* **76**, 146–164 (2015)

9. Anggorojati, B., Mahalle, P.N., Prasad, N.R., Prasad, R.: Capability-based access control delegation model on the federated IoT network. In: The 15th International Symposium on Wireless Personal Multimedia Communications, pp. 604–608. IEEE, Taipei, Taiwan (2012)
10. Lin, Y.-B., Shieh, M.-Z., Lin, Y.-W.: DormTalk: edge computing for the dormitory applications on campus. *IET Networks* **8**(3), 179–186 (2018)
11. TaiSEIA 101 Interconnection protocol for devices in smart home. <http://www.taiseia.org.tw/Affairs/>. Accessed 10 Aug 2020
12. Fett, D., Küsters, R., Schmitz, G.: A comprehensive formal security analysis of OAuth 2.0. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1204–1215. ACM, Vienna, Austria (2016)
13. The OAuth 2.0 Authorization Framework. <https://tools.ietf.org/html/rfc6749>. Accessed 10 Aug 2020
14. Emerson, S., Choi, Y.-K., Hwang, D.-Y., Kim, K.-S., Kim, K.-H.: An OAuth based authentication mechanism for IoT networks. In: 2015 International Conference on Information and Communication Technology Convergence (ICTC), pp. 1072–1074. IEEE, Jeju, South Korea (2015)
15. Fremantle, P., Aziz, B., Kopecký, J., Scott, P.: Federated identity and access management for the internet of things. In: 2014 International Workshop on Secure Internet of Things, pp. 10–17. IEEE, Wroclaw, Poland (2014)
16. Siris, V.A., Dimopoulos, D., Fotiou, N., Voulgaris, S., Polyzos, G.C.: OAuth 2.0 meets blockchain for authorization in constrained IoT environments. In: 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), pp. 364–367. IEEE, Limerick, Ireland (2019)
17. Nmap: the Network Mapper - Free Security Scanner. <https://nmap.org/>. Accessed 10 Aug 2020
18. Basic Security Testing Baseline for Mobile Applications v3.0. <https://www.mas.org.tw/spaw2/uploads/files/benchmark/Basic-Security-Testing-Baseline-for-Mobile-Applications-v3.0.pdf>. Accessed 10 Aug 2020