DESIGN AND IMPLEMENTATION OF RASPBERRY HOUSE: AN IoT SECURITY FRAMEWORK

Wen Fei
Faculty of Computer Science
Dalhousie University
Halifax, Canada
wn335813@dal.ca

Hiroyuki Ohno
Information Media Center
Kanazawa University
Kanazawa, Japan
hohno@staff.kanazawa-u.ac.jp

Srinivas Sampalli
Faculty of Computer Science
Dalhousie University
Halifax, Canada
srini@cs.dal.ca

Abstract— **The rising popularity of the Internet of Things (IoT) on a global scale has led to an increase in cyber threats, and researchers are paying more attention to its security issues. So far, research on IoT security has focused on large-scale devices, but there is relatively less research on the security of small IoT devices. Therefore, our objective is to mainly study how to make the operation of small IoT devices safer. Raspberry House is a TCP/IP Layer 3 gateway built with Raspberry Pi, which can connect IoT devices to the private network generated by it, thereby preventing IoT devices from being exposed to outside networks. In addition, through a private network, IoT devices can also update their firmware wirelessly. This paper also studies the communication between IoT devices through different secure connections such as Secure Shell (SSH), and evaluates their results in different environments. Experimental evaluation of TCP/IP Layer 3 Gateway indicates that the proposed framework can provide security for small IoT devices.**

Keywords— Internet of Things (IoT), IoT security, Raspberry Pi, Raspberry House Gateway, Over-the-air (OTA), small IoT device

I. INTRODUCTION

IoT is a network that connects various physical objects to the Internet for information exchange and communication to realize intelligent identification, positioning, tracking, monitoring and management of items [1], [2]. As a large number of devices are connected to the IoT to provide innovative and interconnected services, network security has gradually developed into one of the prominent problems in the Internet field, and leakage incidents caused by network security are on the rise globally [3], [4]. At present, researchers mainly study the security of large IoT devices which includes connected vehicles, smart home, wearable technology, connected health devices, devices with remote monitoring capabilities and so on [5]. The security of small IoT devices has not received extensive attention from researchers. For some commercial small IoT devices, manufacturers can provide users with updates from the official repository. However, some non-professionals who are interested in IoT area may use original development devices to develop certain systems or projects (for example, in this research, we have used the ESP8266 development board as microcontroller). In such case, they just ensure that the small IoT devices in the system can work as they expect, without considering the security of the devices. Therefore, the purpose of this research is to establish a secure environment for small IoT devices by designing a TCP/IP Layer 3 gateway.

We have used Raspberry Pi to build this gateway and have named it Raspberry House. It will generate a private network and assign the private IP address to the small IoT devices so that the devices will not be exposed to outside networks. We have also implemented the wireless firmware update of the IoT devices inside the gateway. Therefore, all the IoT devices inside the gateway can be updated synchronously without connecting them to the USB interface, which save time. Furthermore, we have considered that in practical scenarios, each organization may invest in a large number of IoT devices to use, and these IoT devices may be distributed all over the world. Therefore, we need to consider the communication between IoT devices. In order to ensure that the communication is secure, we have used several popular methods to analysis the security of IoT device communication and given suggestions for different environmental conditions.

The rest of the paper is organized as follows. Section II and Section III introduces background and literature survey involved in the research. Section IV discusses design of the framework, TCP/IP Layer 3 gateway, inside and outside design of the Raspberry House. Section V describes the settings and analysis of the research, including the security settings on Raspberry Pi, Interior Connection of the Raspberry House and Exterior Connection of the Raspberry House. Section VI evaluates the performance of the research and summarizes the evaluation. Section VII concludes the paper and discusses the future work.

II. BACKGROUND

A. Raspberry Pi

Raspberry Pi is a microcomputer based on Linux. The components are all integrated on a motherboard that is only slightly larger than a credit card [6]. It has all the basic functions of a personal computer (PC) and only needs to be connected to a monitor and a keyboard to execute functions such as text processing, gaming, video and many others [6], [7]. Just like any other desktop computer or portable computer

running Linux, Raspberry Pi can be used to deploy many tasks. However, ordinary computer motherboards rely on hard drives to store data, while Raspberry Pi uses SD cards as "hard drives", or it can be connected to an external USB hard drive [6]. Raspberry Pi is relatively cheap, which means that its use is more extensive. This research has used the Raspberry Pi 3 B plus to build the security gateway.

Raspberry Pi used for this research is an open-source system based on Linux. We chose it because it is easy to operate and it is secure. Compared with Windows, which users cannot access its source code, Linux is based on a multi-user architecture, that is, it is more stable than single-user operating systems such as Windows. Since Linux is community-driven and is regularly monitored by developers from all over the world, any new issues raised can be resolved in a short period of time, and necessary patches can be provided at any time.

B. Security Issues of IoT Devices

Some of the security issues commonly encountered in IoT devices are follows:

a) Authenticity: Some websites or markets sell pirated components [4].

b) Weak password: The attacker uses some tools to crack the system password, such as collect a list of commonly used weak passwords, and obtain authentication passwords for system-related services through machine attempts [8].

c) Information leakage: The leaked information [9] greatly facilitates the attacker's attack on the target, and the attacker can obtain the plaintext password by force cracking.

d) Unauthorized access: The attacker can access and control the target system without the authorization of the administrator [4].

e) Remote code execution: Developers lack secure coding abilities, fail to strictly filter and verify input parameters, resulting in remote code execution or command intrusion when calling dangerous functions [9].

f) Man-in-the-middle (MITM) attack: The attacker is in the middle of the link at both ends of the communication and acts as a data exchange role [4], [9].

g) Cloud attack: As IoT devices are gradually being managed through the cloud [9], attackers can exploit cloud provider vulnerabilities to analyze communication between the device and the cloud, and forge data to perform replay attacks to gain device control.

This research protects small IoT devices from encountering the security issues in above. First of all, the components of this research are purchased from official websites, which guarantees the authenticity. We have set strong password and authentication keys for Raspberry Pi to avoid weak password issue. We have saved the information in a safe way to avoid information leakage, and set security settings for both Raspberry Pi and IoT firmware to avoid unauthorized access. We have used SSH port forwarding to avoid remote code execution, and TLS encryption to avoid MITM attack. Finally, for cloud attack, we have used Sakura Virtual Private Service (VPS) as our cloud service which is safe.

C. Sakura VPS

Sakura VPS [10] is an IoT platform which can provide a communication module for sending and receiving data by repeating input and output of information between the IoT. Furthermore, it is a system for data storage and collaborate. Because it uses a closed network security environment, we do not need to build or run another secure communication environment or cloud environment [11], [12] for storing or classifying data.

III. Literature Survey

A. Performance Measurement of Raspberry Gate

Hu et al. [13] use Raspberry Pi to build a small security gateway named Raspberry Gate, and use Raspberry Guardian (human community) to update information, manage and control IoT devices to establish a secure network environment. Raspberry Gate has three modes, namely router mode, bridge mode and maintenance mode. The router mode allows the source IP or destination IP in the IP data packet converted between the private and public network. In this case, Raspberry Guardian can use Raspberry Gate to protect the internal network and the devices which connected to the internal network. The bridge mode can provide packet bridging between inbound and outbound traffic with or without packet filters. Maintenance mode is the mode used by Raspberry Gate for automatic updates. The latest update package of Raspberry Gate can automatically download from the Raspberry Guardian software repository on GitHub.

B. Security Measurement of IoT

Assiri and Almagwashi [9] introduced the IoT and its three-tier architecture, reviewed its main security and privacy issues and challenges, and focused on some proposed solutions for IoT security. They pointed out that security and privacy are the main challenges facing the future adaptation and development of IoT. Minoli et al. [14] also proposed that it is necessary to develop a comprehensive security and privacy framework to meet the challenges and related influencing factors in the IoT environment.

Sezer [15] introduced the vulnerability of embedded systems, various IoT security and privacy threats and challenges caused by basic communication and chip technologies of IoT devices. He also summarized the emerging IoT security technologies and the trend of future IoT security research. Meneghello et al. [4] outline the security risks in the IoT field and discuss the specific security mechanisms used by the most popular IoT communication protocols.

Chouhan et al. [16] introduced the basic elements of the IoT model and evaluated current state of the IoT applications. The authors proposed that in order to ensure security, IoT devices, data, sensors, interfaces, and connected physical need to be protected. Neshenko et al. [17] focused on IoT

vulnerabilities. The authors introduced many IoT vulnerabilities and their attack vectors through classification, and discussed corresponding remedial methods to monitor the attacks of such vulnerabilities.

Chandra et al. [18] used small IoT devices such as Raspberry Pi to build an intruder image capture system to automate home appliances. When an intruder enters the house, the system can capture the image of the intruder and send it to authorized mail via the Simple Mail Transfer Protocol (SMTP). Although this system can provide safety protection for the family, when developing the system, the authors did not consider the security of the small IoT devices they used.

C. *Remote Management of IoT*

Shen et al. [19] proposed a series of solutions to improve standard over-the-air (OTA) measurement to meet IoT OTA requirements. For single-input single-output (SISO) terminals, three testing techniques have been introduced to speed up the overall isotropic sensitivity test and improve the test accuracy. For IoT multiple input multiple output (MIMO) devices, the Radiation Two-stage (RTS) method is introduced, which can adapt to OTA evaluation and provide assistance to the IoT industry.

Yoon and Kim [20] proposed the functional architecture of a remote security management server to improve the security of IoT devices in the IoT environment, which can prevent various infringement incidents in the IoT environment in advance. Even if a serious attack occurs, the damage can be minimized by taking quick and effective countermeasures.

IV. Raspberry House Design

This section describes the design framework, gateway design and the Raspberry House design.

A. *Design Framework*

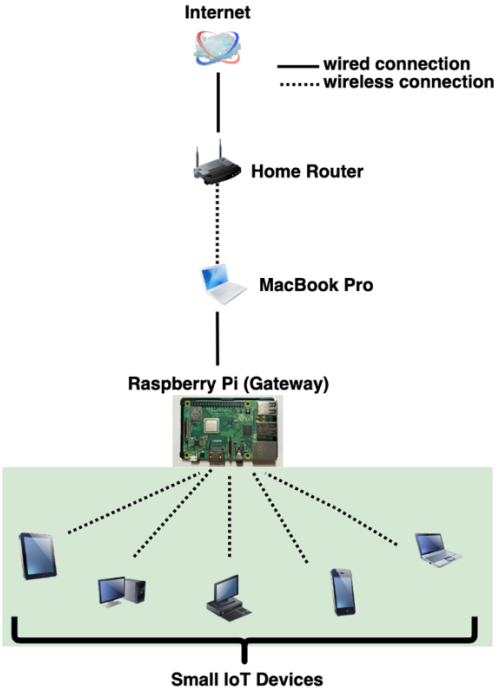

Fig. 1. Design framework

Figure 1 shows the design framework. It shows that small IoT devices and the Internet are connected and communicated through the gateway which designed by Raspberry Pi. By defining an IP address for the gateway, it can generate the private network IP address and assign it to small IoT devices. In this way, all small IoT devices are in a private network, thereby improving security. In order to achieve communication between small IoT devices, the gateway is connected to the Internet.

A Raspberry Pi within an Ethernet network can be used as a wireless access point by creating a private network. The resulting new wireless network is entirely managed by the Raspberry Pi. As shown in Figure 1, we used MacBook Pro as a server and used the Ethernet interface of the Raspberry Pi to connect to the MacBook Pro with an Ethernet cable. Then used the Wi-Fi interface of the Raspberry Pi to connect to small IoT devices. In addition, MacBook Pro uses a wireless connection to the home router. The home router is then connected to the Internet. In this way, we have made a gateway in the gateway, which enhances security of small IoT devices connected to our gateway.

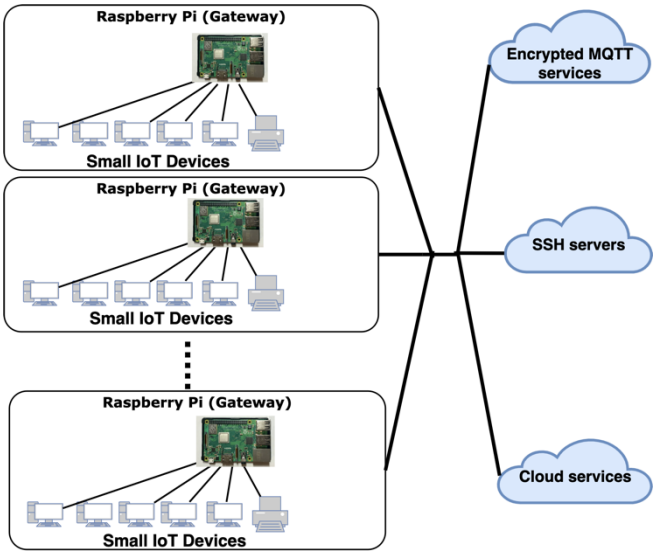

Fig. 2. Implementation framework

As shown in Figure 2, to achieve secure communication between small IoT devices, we have tested TLS encryption technology, SSH port forwarding and cloud services.

B. *TCP/IP Layer 3 Gateway Design*

The latest version of Raspberry Pi already has on-board wireless capabilities, which can act as many different devices. In this research, Raspberry Pi acts as a gateway. We name the gateway Raspberry House, which is a TCP/IP Layer 3 gateway, and make Raspberry House as a wireless access point which can connect small IoT devices to a proprietary network to achieve security protection for small IoT devices.

Raspberry House provides Dynamic Host Configuration Protocol (DHCP) server, Timedatectl server, Hostapd server and Domain Name System (DNS) server. The DHCP server is responsible for providing IP addresses for DHCP clients and managing the assigned IP addresses. Timedatectl can provide time services for computers on the network, function of hostapd is to act as an access point authentication server,

responsible for controlling and managing the access and authentication of stations, and DNS server can uniquely identify the Internet host [13].

C. *Raspberry House Design: Internal Environment*

In order to enable the firmware of multiple IoT devices to be updated at the same time, this research considered how to enable small IoT devices to update their firmware securely over a wireless network. Therefore, the user of the devices no longer needs to use a USB cable to connect each small IoT device to update the firmware, which can save the user a lot of time.

In order to realize this design, we have used OTA. OTA is a technology that realizes remote management of mobile terminal equipment through the air interface of mobile communication [19]. Generally, OTA is divided into two categories, one is Firmware Over The Air (FOTA) [21], [22], which refers to downloading a complete firmware image to a device or Electronic Control Unit (ECU) flash memory, or repairing existing firmware and updating Flash memory. The other is Software-over-the-air (SOTA), which refers to those applications and map OTAs that seem closer to the user. This research has used FOTA to complete the firmware update of small IoT devices.

D. *Raspberry House Design: External Environment*

In this research, in order to ensure the security outside the gateway, we have used some common methods. For example, this research has used information encryption technology to encrypt the transmitted messages, SSH port forwarding to set the port, and the cloud service to control the information transmission of small IoT devices.

V. Experimental Settings and Analysis

A. *Security Settings on the TCP/IP Layer 3 Gateway for Raspberry House – General Settings*

We used Raspberry Pi 3B plus, Ethernet cable, USB to ethernet adapter, Micro SD card, SD card reader and MacBook Pro (server) as the hardware to build Raspberry Pi as a wireless access point and make it act as a gateway by creating a private network. Regarding the software part, we need to install Raspberry Pi Operating System (OS) with desktop and recommended software on the Raspberry Pi, connect the Raspberry Pi to the Ethernet, and then start the Raspberry Pi OS.

In order to be used as an access point, Raspberry Pi needs to install the hostapd access point software package and configure hotspot parameters. In this research, the SSID in the hostapd configuration file is set to Raspberry House, which is the name of the WiFi signal broadcast from the Raspberry Pi. We also set the password, wpa_passphrase, which needs to be entered when connecting devices to Raspberry House for the first time. After completing the above steps, we enabled and started the hostapd, and as shown in Figure 3, hostapd is enabled at wlan0 interface.

```
pi@wenpipi:~ $ sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
Configuration file: /etc/hostapd/hostapd.conf
wlan0: Could not connect to kernel driver
Using interface wlan0 with hwaddr b8:27:eb:79:20:82 and ssid "RASPBERRYHOUSE"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
```

Fig. 3. Result of hostapd status

The Raspberry Pi runs a DHCP server for the wireless network, which requires static IP configuration for the wireless interface (wlan0) in the Raspberry Pi, and we have set it to 192.168.4.1. dnsmasq is a small tool for configuring DNS and DHCP. It is suitable for small networks and provides DNS and DHCP functions. As shown in Figure 4, for wlan0, it's going to provide IP addresses between 192.168.4.2 and 192.168.4.20, with a lease time of 24 hours, to wireless DHCP clients. In addition, it is able to reach the Raspberry Pi under the name gw.wlan from wireless clients.

```
interface=wlan0
dhcp-range=192.168.4.2,192.168.4.20,255.255.255.0,24h
address=/gw.wlan/192.168.4.1
```

Fig. 4. dnsmasq configuration

After adding routing and masquerade, rebooted Raspberry Pi and tested the new wireless access point with the wireless device. The network SSID specified in the hostapd configuration should be present, and it should be accessible with the specified password.

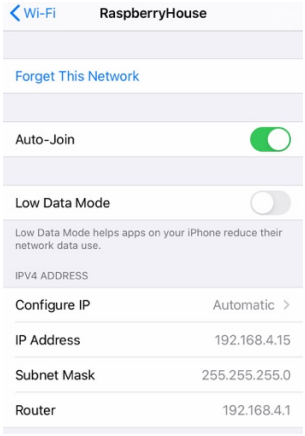

Fig. 5. Use smartphone to connect to Raspberry House

As shown in Figure 5, this is the test result of trying to connect to the wireless access point with a smart phone, and the IP address assigned to the phone is 192.168.4.15, indicating that the Raspberry Pi can be used as a wireless access point: the server is MacBook Pro, and the client can be any wireless device. In addition, because Raspberry Pi can convert the IP address from 192.168.3.2 to 192.168.4.1. Therefore, Raspberry Pi can be used as a TCP/IP layer 3 gateway.

B. *Security Settings on the TCP/IP Layer 3 Gateway for Raspberry House – Special Settings*

In order to make the gateway more secure, we have carried out the following security settings.

a) Change the default username and password: It can be changed by using the raspi-config application.

b) Make 'sudo' require a password: When add 'sudo' in front of the command line, it means that it will run as a superuser which by default does not need a password.

c) Ensure Raspberry Pi has the latest security fixes: In this research, since we have used SSH to connect to Raspberry Pi, we installed "openssh-server" on Raspberry Pi to update ssh-server specifically.

d) Improving SSH security: Besides change the default username and password, we also used key based authentication to improve the security.

e) Install a firewall: We have used Uncomplicated Fire Wall (ufw), which can provide a simpler interface than iptables and it can allow/deny a specific port/service.

f) Install fail2ban: Fail2ban captures the violence of multiple login attempts and can notify any installed firewall to stop further login attempts from suspicious IP addresses.

g) Synchronized time on Raspberry Pi: For ease of use, we have used timedatectl to synchronize time.

C. *Security Settings on the TCP/IP Layer 3 Gateway for Raspberry House – Performance Analysis*

Compared with the Raspberry Gate in the literature survey [13], the router mode of Raspberry House works similarly to Network Address Translation (NAT) routers [13], and mainly uses NAT packet masquerading and port forwarding. Data packet pseudo can change the address information in the data packet to a unified external address information, so that the internal network host will not be directly exposed to the Internet, thereby ensuring the security of the internal network. Port forwarding is required on the gateway to forward data packets of a specific service to the intranet host. In this research, the external network cannot bypass Raspberry House to control the internal small IoT devices, and the small IoT devices cannot access the external network without Raspberry House. Raspberry House does not have maintenance mode, which means it cannot update automatically. Although it has to update manually each time, Raspberry House can change the configuration files more flexible than Raspberry Gate.

D. *Interior Connection of the Raspberry House – Prototyping of IoT Devices using Small Microcontrollers*

In this research, due to resource constraints, there is only one NodeMCU device (micro-controller unit for small IoT device) inside the gateway. We have used Micro USB cable, 3 LEDs, NodeMCU board, breadboard jumpers and breadboard as IoT device, and have used Arduino IDE to write execution code for NodeMCU board so that the LEDs can be turned on and off. NodeMCU board includes ESP8266 with a WiFi-enabled chip. ESP8266 is a low-cost WiFi chip developed using the Espressif system's TCP/IP protocol [23]. Figure 6 shows the schematic diagram used in this research.

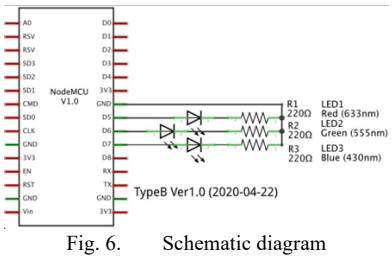

Fig. 6. Schematic diagram

E. *Interior Connection of the Raspberry House – IoT Device to Private Network Connection*

This research has used MQ Telemetry Transport (MQTT) protocol, which is a lightweight messaging protocol and it uses a publish/subscribe (pub/sub) model [24], [25]. The pub/sub model is based on a MQTT broker. Clients can publish a message on a topic or subscribe to a particular topic through the broker. Furthermore, topics basically represents the destination address for the message which sent by MQTT broker. Client who subscribe to a topic will receive all the message published to that topic [24]. We have used public MQTT broker to connect the small IoT device to Raspberry House. This research used MQTT protocol not only because it is easy to send messages, but also because that small IoT devices inside the gateway cannot reach the outside of the gateway, and devices outside the gateway cannot check the internal state of the gateway, so we have used MQTT protocol to exchange messages from inside the gateway to outside the gateway.

F. *Interior Connection of the Raspberry House – Secure Firmware Update of IoT Device*

This research has used OTA to update the firmware of small IoT devices, which means the research needs to make an "OTA over the TCP/IP Layer 3 Gateway". Therefore, we have prepared a binary file of the sketch and have used the scp command on the laptop to remotely copy the target file (*.ino.bin file) and espota.py file to Raspberry Pi. After logging in to the Raspberry Pi, changed the directory to the location of the binary file and espota.py file, and uploaded the firmware.

G. *Exterior Connection of the Raspberry House – TLS/SSL Encryption*

The certificate generator has 3 scripts namely the root certification authority (CA) generator, the server certificate generator and the client certificate generator [26], [27]. We have logged in to Raspberry Pi via SSH and taken a note of the hostname of the Raspberry Pi using the hostname command, which would passed as an argument during the server certificate generation. Then, we have established the trust network by generating a CA. Furthermore, we have generated the certificate and key for both server and client.

To enable security in the MQTT broker, we have changed the mosquitto configuration so that the broker understands to use the secure channel. In the configuration file, change the secured default port to 8883 and set the certificate authentication flag to true. Figure 7 shows that when MQTT broker is running, it is listening on the secure port 8883 to prove that the TLS encryption is successful.

```
pi@wenpipi:~/Desktop/certs $ sudo mosquitto -v -c /etc/mosquitto/mosquitto.conf
1596776622: mosquitto version 1.5.7 starting
1596776622: Config loaded from /etc/mosquitto/mosquitto.conf.
1596776622: Opening ipv4 listen socket on port 8883.
1596776622: Opening ipv6 listen socket on port 8883.
```

Fig. 7. Result of TLS encryption

H. *Exterior Connection of the Raspberry House – SSH Port Forwarding*

In this research, we have used remote port forwarding, which is extremely similar to local port forwarding [28]. Remote SSH port forwarding is typically used if the local

computer doesn't have a public IP address, and researchers need to access it from the outside. Figure 8 shows that we have used SSH to access local PC via port 22222 of remote server.

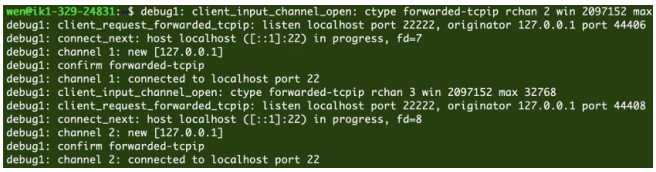

Fig. 8. Result of SSH remote port forwarding

I. Exterior Connection of the Raspberry House – Cloud Services

This research has used Sakura VPS as cloud service. As shown in Figure 9, inside the Raspberry Pi, Sakura VPS can be connected the same way as SSH remote port forwarding.

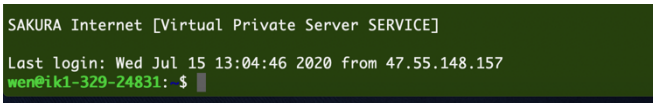

Fig. 9. Result of using Sakura VPS

VI. Evaluation and Summary

A. Evaluation of IoT device to private network connection

This research successfully designed and implemented Raspberry House, which can separate the internal network from the external network. Without Raspberry House, small IoT devices cannot be able to communicate with external networks. Furthermore, because small IoT devices use dedicated IP addresses for internal networks, external networks cannot bypass Raspberry House and connect to the devices. For OTA security, this research ensures the security of cloud servers, small IoT devices, and communication between small IoT devices and the cloud.

B. Evaluation of TLS/SSL Encryption

For TLS protocol, it is important to consider whether the connection is safe and reliable. Using the TLS protocol, each message sent will pass a message authentication code (MAC) to check the integrity of the message, which means that the connection is reliable. In addition, using this protocol, the public key can be used to verify the identity of the communicating party, but for some very confidential applications, it is still necessary to verify the identity of both parties, which means that TLS encryption is safe.

C. Evaluation of SSH Port Forwarding

Basically, all processes of SSH data transmission are encrypted using symmetric keys. However, asymmetric encryption is used in the initial connection creation phase and the authentication handshake phase. The difference between asymmetric encryption and symmetric encryption is that in order to send data in a single direction, a related set of keys are required (public key and private key). In this research, we only share the public key, and keep the private key strictly confidential and not disclosed to anyone, and we have set a password for the private key used for system authentication in SSH to prevent it from leaking. Therefore, SSH port forwarding is secure.

D. Evaluation of Cloud Services

Cloud services are relatively safe. The cloud service provider guarantees that data will not be tampered with by unauthorized users, or can be quickly discovered by the system after tampering, to ensure data integrity during data transmission and storage. Moreover, cloud service providers attach importance to every aspect of massive data transmission, storage and processing to protect the privacy of the data.

However, the disadvantage of cloud services is that cloud servers rely on the network, which means that if researchers lose Internet access, they will lose access to data. The data is still safe, but temporarily unavailable. This will not affect local networks with independent servers. Due to the small scale of this research and independent server, this drawback can be temporarily ignored.

E. Summary

Based on the above, we can conclude that Raspberry House is relatively safe. It can protect small IoT devices and keep them in a safe environment. The connection inside the gateway is secure, and wireless firmware updates can be performed safely. In order to protect the external security of Raspberry House, if the focus is on encrypting the transmitted content during data transmission and avoiding MITM attacks, the TLS protocol is relatively better; if the local computer does not have a public IP address and researchers need to access it from the outside, then SSH remote port forwarding is relatively better than others; if researchers have a large number of IoT devices to be managed, then cloud services will be a good choice. As far as this research is concerned, due to the small scale of the research, TLS protocol and SSH port forwarding performs better than cloud services.

VII. Conclusion and Future Work

A. Conclusion

Some researchers use small IoT devices to build systems to achieve their goals without considering the security of small IoT devices in the system, which will bring security risks. For example, if they use a small IoT device as the microcontroller of their system, once the device is attacked, the entire system will collapse. Therefore, our research uses Raspberry Pi to build a TCP/IP Layer 3 gateway, which can connect small IoT devices to the private network generated by it, thereby preventing small IoT devices from being exposed to public networks. In addition, through a dedicated network, small IoT devices can also update their firmware wirelessly and securely. This research also studies the communication between IoT devices through different secure connections and evaluates their results in different situations. If researchers want to encrypt the transmitted data to avoid MITM attacks, then TLS protocol is better than other protocols. If the local computer does not have a public IP address, and researchers need to access it from the outside, which means that the port being mapped is on the remote server, and the request being sent is sent from the local computer, then they should use SSH port forwarding. If researchers need to manage a large number of IoT devices, then cloud services will be a good choice. Therefore, it can be qualitatively judged that this research can provide a secure environment for small IoT devices.

B. Future work

In future research, we will consider how to maintain Raspberry House more efficiently, and we will also try to test the security communication of Virtual Private Network (VPN) [29]. Up to now, the Raspberry House can detect some common attacks very well. In the future, we will also consider making Raspberry House detect more complex attacks. In addition, we will conduct experiments to test real security in practice. Since our research is still in the development, this research has used both Raspberry Pi and MacBook Pro because MacBook Pro is very convenient to use. Our final goal is to export all content from MacBook Pro to Raspberry Pi to achieve introducing a single unit instead of two units. Furthermore, our input/output number is limited, which is enough to meet the needs of our current research work. In the future, we will consider using field programmable gate array (FPGA).

ACKNOWLEDGMENT

The authors thank Mr. Hironobu Suzuki and Prof. Yoshiaki Kitaguchi for their creative suggestions, and Shuting Hu for helping with the experimentation.

REFERENCES

[1] K. Routh and T. Pal, "A survey on technological, business and societal aspects of Internet of Things by Q3, 2017," in *Proceedings - 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages, IoT-SIU 2018*, 2018, doi: 10.1109/IoT-SIU.2018.8519898.

[2] S. Narang, T. Nalwa, T. Choudhury, and N. Kashyap, "An efficient method for security measurement in internet of things," in *Proceedings of the 2018 International Conference On Communication, Computing and Internet of Things, IC3IoT 2018*, 2019, pp. 319–323, doi: 10.1109/IC3IoT.2018.8668159.

[3] A. J. Cui, C. Li, and X. M. Wang, "Real-time early warning of network security threats based on improved ant colony algorithm," in *Proceedings - 2019 12th International Conference on Intelligent Computation Technology and Automation, ICICTA 2019*, 2019, pp. 309–316, doi: 10.1109/ICICTA49267.2019.00072.

[4] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8182–8201, Oct. 2019, doi: 10.1109/JIOT.2019.2935189.

[5] S. Chaudhary, R. Johari, R. Bhatia, K. Gupta, and A. Bhatnagar, "CRAIoT: Concept, Review and Application(s) of IoT," in *Proceedings - 2019 4th International Conference on Internet of Things: Smart Innovation and Usages, IoT-SIU 2019*, 2019, doi: 10.1109/IoT-SIU.2019.8777467.

[6] N. S. Yamanoor and S. Yamanoor, "High quality, low cost education with the Raspberry Pi," in *GHTC 2017 - IEEE Global Humanitarian Technology Conference, Proceedings*, 2017, vol. 2017-January, pp. 1–5, doi: 10.1109/GHTC.2017.8239274.

[7] R. W. Sudibyo, N. Funabiki, M. Kuribayashi, K. I. Munene, M. M. Islam, and W. C. Kao, "A TCP Fairness Control Method for Two-Host Concurrent Communications in Elastic WLAN System Using Raspberry Pi Access-Point," in *2019 2nd International Conference on Communication Engineering and Technology, ICCET 2019*, 2019, pp. 107–111, doi: 10.1109/ICCET.2019.8726907.

[8] A. Hameed and A. Alomary, "Security issues in IoT: A survey," in *2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies, 3ICT 2019*, 2019, pp. 1–5, doi: 10.1109/3ICT.2019.8910320.

[9] A. Assiri and H. Almagwashi, "IoT Security and Privacy Issues," in *1st International Conference on Computer Applications and Information Security, ICCAIS 2018*, 2018, doi: 10.1109/CAIS.2018.8442002.

[10] S. Cherrared, S. Imadali, E. Fabre, and G. Gössler, "Sakura a model based root cause analysis framework for VIMS," in *MobiSys 2019 - Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, 2019, pp. 594–595, doi: 10.1145/3307334.3328642.

[11] J. Upadhyaya and N. J. Ahuja, "Quality of service in cloud computing in higher education: A critical survey and innovative model," in *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017*, 2017, pp. 137–140, doi: 10.1109/I-SMAC.2017.8058324.

[12] A. Sun, G. Gao, T. Ji, and X. Tu, "One Quantifiable Security Evaluation Model for Cloud Computing Platform," in *Proceedings - 2018 6th International Conference on Advanced Cloud and Big Data, CBD 2018*, 2018, pp. 197–201, doi: 10.1109/CBD.2018.00043.

[13] S. Hu, H. Suzuki, Y. Kitaguchi, H. Ohno, and S. Sampalli, "Design, implementation and performance measurement of raspberry gate in the IoT field," in *ACM International Conference Proceeding Series*, 2019, pp. 82–89, doi: 10.1145/3361821.3361827.

[14] D. Minoli, K. Sohraby, and J. Kouns, "IoT security (IoTSec) considerations, requirements, and architectures," in *2017 14th IEEE Annual Consumer Communications and Networking Conference, CCNC 2017*, 2017, pp. 1006–1007, doi: 10.1109/CCNC.2017.7983271.

[15] S. Sezer, "T1C: IoT Security: - Threats, Security Challenges and IoT Security Research and Technology Trends," in *2018 31st IEEE International System-on-Chip Conference (SOCC)*, 2018, pp. 1–2, doi: 10.1109/SOCC.2018.8618571.

[16] P. K. Chouhan, S. McClean, and M. Shackleton, "Situation assessment to secure IoT applications," in *2018 5th International Conference on Internet of Things: Systems, Management and Security, IoTSMS 2018*, 2018, pp. 70–77, doi: 10.1109/IoTSMS.2018.8554802.

[17] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019, doi: 10.1109/COMST.2019.2910750.

[18] M. L. R. Chandra, B. V. Kumar, and B. Sureshbabu, "IoT enabled home with smart security," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing, ICECDS 2017*, 2018, pp. 1193–1197, doi: 10.1109/ICECDS.2017.8389630.

[19] P. Shen, Y. Qi, W. Yu, J. Fan, and F. Li, "OTA Measurement for IoT Wireless Device Performance Evaluation: Challenges and Solutions," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 1223–1237, Feb. 2019, doi: 10.1109/JIOT.2018.2868787.

[20] S. Yoon and J. Kim, "Remote security management server for IoT devices," in *International Conference on Information and Communication Technology Convergence: ICT Convergence Technologies Leading the Fourth Industrial Revolution, ICTC 2017*, 2017, vol. 2017-December, pp. 1162–1164, doi: 10.1109/ICTC.2017.8190885.

[21] C. C. Teng, J. W. Gong, Y. S. Wang, C. P. Chuang, and M. C. Chen, "Firmware over the air for home cybersecurity in the Internet of Things," in *19th Asia-Pacific Network Operations and Management Symposium: Managing a World of Things, APNOMS 2017*, 2017, pp. 123–128, doi: 10.1109/APNOMS.2017.8094190.

[22] H. A. Odat, A. Nsour, and S. Ganesan, "Firmware over the air ad-hoc network, FOTANET," in *IEEE International Conference on Electro Information Technology*, 2015, vol. 2015-June, pp. 101–106, doi: 10.1109/EIT.2015.7293326.

[23] K. A. Arunkumar, K. Kriti, A. Das, and B. Bhattacharyya, "Wireless Speakers using WiFi and IoT," in *Proceedings - International Conference on Vision Towards Emerging Trends in Communication and Networking, ViTECoN 2019*, 2019, doi: 10.1109/ViTECoN.2019.8899444.

[24] D. Soni and A. Makwana, "A SURVEY ON MQTT: A PROTOCOL OF INTERNET OF THINGS(IOT) MP-Index View project Analysis and Survey on String Matching Algorithms for Ontology Matching View project A SURVEY ON MQTT: A PROTOCOL OF INTERNET OF THINGS(IOT)."

[25] S. L. Jurj, R. Rotar, F. Opritoiu, and M. Vladutiu, "White-Box

Testing Strategy for a Solar Tracking Device Using NodeMCU Lua ESP8266 Wi-Fi Network Development Board Module," in *2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging, SIITME 2018 - Proceedings*, 2019, pp. 53–60, doi: 10.1109/SIITME.2018.8599250.

[26] P. Szalachowski, S. Matsumoto, and A. Perrig, "PoliCert: Secure and flexible TLS certificate management," in *Proceedings of the ACM Conference on Computer and Communications Security*, 2014, pp. 406–417, doi: 10.1145/2660267.2660355.

[27] P. Fu, Z. Li, G. Xiong, Z. Cao, and C. Kang, "SSL/TLS Security Exploration Through X.509 Certificate's Life Cycle Measurement," in *Proceedings - IEEE Symposium on Computers and Communications*, 2018, vol. 2018-June, pp. 652–655, doi: 10.1109/ISCC.2018.8538533.

[28] G. P. Sharma, W. Tavernier, D. Colle, and M. Pickavet, "Dynamic accelerator provisioning for SSH tunnels in NFV environments," in *Proceedings of the 2019 IEEE Conference on Network Softwarization: Unleashing the Power of Network Softwarization, NetSoft 2019*, 2019, pp. 242–244, doi: 10.1109/NETSOFT.2019.8806690.

[29] J. Zhang, "Research on Key Technology of VPN Protocol Recognition," in *Proceedings of 2018 IEEE International Conference of Safety Produce Informatization, IICSPI 2018*, 2019, pp. 161–164, doi: 10.1109/IICSPI.2018.8690472.