Secure and Lightweight Firmware Update Framework for IoT Environment

Alexander Yohan*, Nai-Wei Lo†, Liem Peter Santoso‡
Department of Information Management
National Taiwan University of Science and Technology
Taipei 106, Taiwan
{*d10309802,‡m10515809}@mail.ntust.edu.tw; †nwlo@cs.ntust.edu.tw

***Abstract*—High market demands on IoT devices give rise to numerous malware attacks in IoT ecosystem. In order to protect IoT devices, device owners are required to keep their device's firmware up-to-date. However, sometimes the firmware update process could be very challenging for technology illiterate device owners. As there are risks of online adversaries to launch attacks to the IoT device during the firmware update process. In this paper, a robust and lightweight firmware update framework utilizing skipchain technology is proposed to ensure the integrity of distributed firmware and secure firmware distribution process. A prototype based on the proposed firmware update framework is developed to show the feasibility of the proposed framework design.**

***Index Terms*—Internet of Things, firmware update, skipchain, blockchain, p2p verification**

I. Introduction

As IoT gains a lot of popularity and massively adopted in all aspect of human life, IoT is put under the spotlight of online adversaries [1, 2]. In order to protect those IoT devices from online adversaries and malware attacks, it is necessary to keep the device's firmware up to date. Traditionally, the new version of firmware is distributed in a centralized approach from device manufacturer's firmware management center (FMC) to each corresponding IoT device through the Internet, known as Firmware Over-the-Air (FOTA) [3]. Several issues found in traditional FOTA model are: new version of firmware cannot be distributed to all deployed IoT devices in timely manner, FMC suffers excessive network traffic when a lot of IoT devices simultaneously download the new version of firmware, and each IoT device is required to always connected with Internet to receive the new version of firmware from FMC.

In this paper, we propose a decentralized firmware update framework based on skipchain technology. Skipchain is a new blockchain technology that combines the concept of blockchain and skiplist [4]. In our proposed firmware update framework, the skipchain network is constructed from a collection of device manufacturer nodes and gateway nodes in order to efficiently verify the correctness of each transaction data or the distributed firmware in the skipchain network. Two protocols are designed to support the proposed firmware update framework: secure firmware release protocol and secure firmware distribution protocol. In addition, a prototype is implemented to assess the feasibility of the proposed firmware update framework.

II. The Proposed Firmware Update Framework

There are two protocols in the proposed skipchain-based firmware update framework, namely: (1) secure firmware release protocol and (2) secure firmware distribution protocol. Protocol (1) is executed when a device manufacturer releases a new version of firmware and publishes the corresponding firmware information to the skipchain network. Protocol (2) is executed when a new version of firmware is distributed to the targeted IoT device either from a gateway or from another IoT device. There are five entities in the proposed skipchain-based firmware update framework: device manufacturer's firmware repository V_r, device manufacturer node V_n, gateway G_d, gateway node G_n, and IoT device D. The term *firmware repository* refers to the device manufacturer's firmware repository for the rest of this paper.

The secure firmware release protocol is explained as follows:

Step 1. $V_r \leftrightarrow V_n$
V_r and V_n performs Diffie-Hellman key exchange protocol and generates a unique session key k.

Step 2. $V_r \rightarrow V_n$ $\{M = Enc_k\left(M_d^v \parallel LFV_d^v \parallel URL_d^v\right)\}$
Each firmware FW has four attributes: the targeted IoT device model M_d^v, the firmware version LFV_d^v, the firmware binary FB_d^v, and the url to download the corresponding firmware binary URL_d^v.
V_r creates an encrypted message M using session key k and sends the encrypted message M to V_n.

Step 3. The corresponding V_n decrypts the encrypted message M using session key k to obtain the information of new firmware as follow $Fm = Dec_k\left(M\right)$.
The V_n broadcasts the decrypted Fm to cothority members of skipchain network to be verified. Once the verification process of new firmware is finished, the device manufacturer node V_n collects the signature from the participated cothority members. The collected signatures are used to prove that the corresponding Fm has been verified and ready to be put in the new skipchain block $BLOCK_t$.

The secure firmware distribution protocol is explained as follows:

Step 1. $D \leftrightarrow G_d$
D and G_d performs Diffie-Hellman key exchange

978-1-7281-3575-5/19/$31.00 ©2019 IEEE

protocol and generates a unique sesion key k.

Step 2. $D \rightarrow G_d$ $\{M_1 = Enc_k(m_1)\}$
The D creates a message $m_1 = \{sid, ID_d, ADDR_{BLOCK}\}$; in which sid is the current session ID, ID_d is the current IoT device ID, and $ADDR_{BLOCK}$ is the skipchain block address of the currently installed firmware.
D creates an encrypted message M_1 using session key k and sends the encrypted message M_1 to G_d.

Step 3. $G_n \rightarrow G_d$ $\{BLOCK_{curr}\}$
G_d decrypts the received M_1 using session key k and uses the obtained $ADDR_{BLOCK}$ to obtain the skipchain block $BLOCK_{curr}$ from G_n

Step 4. $G_n \rightarrow G_d$ $\{BLOCK_t\}$
G_d obtains the requesting IoT device model M_d^v and the version of firmware CFV_d^v installed in device D from the contract stored in $BLOCK_{curr}$.
Next, the G_n retrieves and sends the latest skipchain block $BLOCK_t$ to G_d. The retrieved $BLOCK_t$ is linked with $BLOCK_{curr}$ and for the specific device model M_d^v.

Step 5. $G_d \rightarrow D$ $\{M_2 = Enc_k(LFB_d^v)\}$
G_d obtains the latest version of firmware LFV_d^v and the url to download the firmware binary URL_d^v of target device model M_d^v from $BLOCK_t$.
G_d checks wheter the requesting device D requires to update its firmware using the following operation $(LFV_d^v > CFV_d^v)$
If the firmware update is required by the requesting device D, then the corresponding G_d will download LFB_d^v from the URL_d^v. Otherwise, terminate the session.
G_d sends and encrypted message M_2 to the requesting device D.

Step 6. The requesting IoT device D decrypts the encrypted message M_2 using the following operation $m_2 = Dec_k(M_2)$ and obtains latest firmware version LFB_d^v. Afterward, the IoT device D verifies the obtained LFB_d^v using its pre-installed device manufacturer's public key. Once the firmware binary LFB_d^v is verified, the corresponding IoT D will update its firmware.

III. Prototype Implementation

Three devices are used to simulate the proposed firmware update framework as follows: MacBook Pro (2.7GHz Intel Core i5 with 8GB RAM), ASUS Desktop PC (3.1GHz Intel Core i5 with 8GB RAM), and Raspberry Pi 3B (1.2GHz QuadCore Broadcom with 1GB RAM). Firmware repository server is implemented in the MacBook Pro. The ASUS Desktop PC runs six VMs as follows: four VMs for four device manufacturer nodes, one VM for gateway node, and one VM for gateway. Lastly, the Raspberry Pi 3B acts as the IoT device. In our experiment, four device manufacturer nodes and one gateway node construct the skipchain network. The four device manufacturer nodes also act as cothority members of the skipchain network. The skipchain network is implemented in Go programming language [4]. In our prototype, Python programming language is used to implement the firmware repository, the interface for device manufacturer node, the interface for gateway node, the gateway, and the IoT device.

During our experiment, we simulate one testing scenario. We assume that one specific device manufacturer releases a new version of firmware through his firmware repository. The information of new firmware is sent from the firmware repository to the device manufacturer node through Internet connection (either cellular or WiFi network). In addition, the gateway sends the new version of firmware to the requesting IoT device through Bluetooth connection.

IV. Conclusion

This paper presents a new skipchain-based firmware update framework for IoT environment in order to securely distribute the new version of firmware to IoT devices. The proposed firmware update framework utilizes the decentralization concept of blockchain in order to avoid the single point of failure issue and to enable quick distribution of a new version of firmware to the IoT devices. In additon, skipchain technology is utilized to efficiently perform the peer-to-peer firmware verification process. Therefore, the proposed skipchain-based firmware update framework could enhance the end-to-end security of the distributed firmware during the firmware update process.

Two protocols is designed to support the proposed firmware update framework, namely: secure firmware release protocol and secure firmware distribution protocol. In addition, a working prototype is implemented based on the proposed protocol design to demonstrate the feasibility of skipchain-based firmware update framework. Furthermore, the proposed skipchain-based firmware update framework could be improved in the future because there are some limitations on the development of the current skipchain technology.

Acknowledgment

The authors gratefully acknowledge the support from TWISC and Ministry of Science and Technology, Taiwan, under the Grant Numbers MOST 108-2218-E-011-021, MOST 108-2221-E-011-063, and MOST 108-2221-E-011-065.

References

[1] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "Ddos in the iot: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[2] N. Vlajic and D. Zhou, "Iot as a land of opportunity for ddos hackers," *Computer*, vol. 51, no. 7, pp. 26–34, jul 2018.

[3] K. Doddapaneni, R. Lakkundi, S. Rao, S. Kulkarni, and B. Bhat, "Secure fota object for iot," in *2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops)*. IEEE, oct 2017, pp. 154–159.

[4] K. Nikitin, E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, I. Khoffi, J. Cappos, and B. Ford, "Chainiac: Proactive software-update transparency via collectively signed skipchains and verified builds," in *26th USENIX Security Symposium (USENIX Security 17)*, 2017, pp. 1271–1287.