An Over-the-Blockchain Firmware Update Framework for IoT Devices

Alexander Yohan[1] and Nai-Wei Lo[2]
Department of Information Management
National Taiwan University of Science and Technology
Taipei, Taiwan
d10309802@mail.ntust.edu.tw[1] and nwlo@cs.ntust.edu.tw[2]

***Abstract*— Nowadays, a lot of Internet-of-Things devices and its applications are developed to improve the people's quality of life. With the growth of circulating IoT devices in human's community, improper device management and firmware distribution mechanism could lead to many issues that harm the security and privacy of the device owner. In this paper, a firmware update framework for IoT devices based on blockchain technology is proposed. The proposed framework aims to provide secure verification on the firmware released by the device manufacturer. In addition, the integrity of distributed firmware to the end-device could be maintained. The proposed firmware update framework consists of four processes: the creation of firmware update contract, the creation of firmware replication contract, the direct firmware update mechanism and the indirect firmware update mechanism.**

Keywords—Internet-of-Things, firmware update, blockchain, smart contract

I. INTRODUCTION

In the recent years, the applications of Internet-of-Things (IoT) can be found in various fields such as transportation and logistics, automotive, healthcare, smart environments, industrial (manufacturing), and especially in consumer electronics. Based on survey performed by Gartner Inc., almost 70% (3.9 billion units out of 6.4 billion units) of the manufactured IoT devices are belonged to the category of consumer application-based devices [1]. Furthermore, business analysts predict the number of IoT devices circulating in the market will be more than 20 billion units by 2020.

As the technology of IoT advances and the IoT devices are massively adopted, there are many emerging vulnerabilities and security challenges in IoT environments [2]–[4]. Several vulnerabilities in IoT environment are listed by Miessler and Smith [3] such as manipulation in code execution flow of an IoT device, intervention during the firmware update process of a device, and anonymous attacker snatching control of a device's console access. As each IoT device is capable to connect with other devices and could connect with Internet, there exist the possibilities for malicious attacker to exploit any vulnerabilities in IoT environment and affecting billion of connected devices.

As the amount of IoT devices and its applications in the human community grow, the IoT device manufacturers are faced with various issues related with the management of the manufactured IoT devices. One of the issues is related with the firmware update process of IoT device. Traditionally, the firmware update process is performed in centralized network model. Although the IoT device manufacturers could have more control in the centralized network model, a centralized network model in IoT environment is prone to single-point of failure. As the blockchain technology is built upon decentralization concept, the workload of network traffic can be distributed to several blockchain nodes during the firmware update process [2], [5], [6]. In addition, blockchain technology offers transparency for all transaction data in the network. Thus, any party within the blockchain network is able to observe and verify the authenticity of all the transaction data.

In this paper, a scheme for firmware update process of IoT device based on blockchain technology is proposed. In order to verify the authenticity and integrity of a firmware during the firmware update process, smart contract [7] and consensus protocol from blockchain technology is used in the proposed scheme. In addition, the proposed scheme adopts push-based firmware update mechanism in the blockchain-based environment. Two mechanisms of firmware update process are designed namely direct and indirect firmware update process. In the proposed direct firmware update scheme, a new firmware update contract is created for every time a device vendor (manufacturer) releases a new version of firmware for a device. After the newly created firmware update contract is created, it will be distributed to all nodes in the blockchain network to be verified by peer nodes. The peer nodes verify the firmware update contract through consensus protocol. After the firmware update contract has passed the verification process, any devices that meet the requirements in the contract can download the associated firmware binary file.

In the proposed indirect firmware update scheme, firmware broker is introduced in the scheme to help the device vendor distributing the firmware update binary to the targeted IoT device. In the proposed indirect firmware update scheme, the firmware broker is required to send a contract proposal to the device manufacturer for providing the indirect firmware update service. Once the device manufacturer accepts the proposal, the device manufacturer creates a firmware replication contract

978-1-5386-5790-4/18/$31.00 ©2018 IEEE

and deploys the contract to all nodes in the blockchain network for verification process. After the firmware replication contract is verified, then the broker node could provide the indirect firmware update process for the corresponding firmware version.

The contributions of our work are as follows:

- A novel design of firmware update framework based on blockchain technology is proposed. There are six entities involved in the proposed firmware update framework, namely: vendor repository, broker repository, full node (for vendor and broker nodes), lightweight node, IoT gateway, and IoT device.
- Two firmware update mechanisms are designed namely direct and indirect firmware update process.
- Four process workflow diagrams are designed to support the functionalities of the proposed firmware update framework. These four processes are: the creation of firmware update contract, the direct firmware update process, the creation of firmware replication contract, and the indirect firmware update process.

This paper is organized as follows. Section II explains the existing works on firmware update mechanism for IoT and existing blockchain-based firmware update mechanism. Section III introduces and explains the design for the proposed blockchain-based firmware update framework. Finally, concluding remarks are presented in Section IV.

II. Literature Review

A. Firmware Update Mechanism for Internet-of-Things

As the adoption of IoT devices in human society increase, numerous vulnerabilities and threats directed to IoT devices and its applications are identified. Cui et al. [4] and Prada-Delgado et al. [8] listed several attacks on the firmware of IoT devices. An attacker could inject the targeted IoT device with malicious firmware in order to sabotage the functionalities of the corresponding target device. Malicious attacker exploits the vulnerabilities to control the infected target device remotely. The infected target device then is used to launch denial-of-service attack or even to violate the privacy of the target device owner.

Generally the process of firmware update mechanism can be differentiated into: manual and automatic updates. As the name implies, the two mechanisms are differ based on the participation of device owner during the firmware update process. In the manual firmware update process, the device owner must initiate the process manually. The manual firmware update process is used in the traditional update mechanism and preferred for device with limited network connectivity. However, the manual firmware mechanism generally has higher time consumption compared with the automatic firmware update mechanism. In addition, it is possible for human error to happen during the update process in the manual update mechanism. Compared with the manual update process, the automatic update mechanism is more attractive to be used nowadays. As the IoT device manufacturer could initiate the firmware update process without waiting for the device owner's participation.

In general, the automatic firmware update process follows the client-server architecture where as the device manufacturer repository acts as the server-side and the IoT devices are the client-side. Based on the procedure to deliver the firmware binary from the server side to the client side, the firmware update method is divided into two categories: PUSH method and PULL method [9]. The main difference between PUSH and PULL methods lies in the initiator of firmware update process. The device manufacturer takes the initiative to distribute the firmware binary to the IoT device management (gateway) in the PUSH method. On the contrary, the IoT device initiates the firmware update process by sending a request to download a specific version of firmware binary to the server of device manufacturer in the PULL method.

B. Blockchain-based Firmware Update Mechanism

In [10], Lee and Lee proposed a blockchain-based firmware verification and firmware update schemes for embedded devices in IoT environment. In their scheme, blockchain technology is used for three purposes: to verify the version of firmware, to verify the authenticity of a specific firmware version, and to distribute a specific version of firmware binary to the connected nodes in the blockchain network. In their proposed scheme, each IoT device represents a node in the blockchain network. Therefore, the IoT devices are required to store a copy of blockchain's ledger in the devices' local storage. As the majority of IoT devices have limited resources (e.g. energy, computation, and storage capacity), the proposed scheme of Lee and Lee might be difficult to be implemented in the real world IoT environment. In addition, the scheme of Lee and Lee only considers the mechanism of firmware update from one device manufacturer where as there are various kinds of on-the-shelf IoT devices from different manufacturers. Therefore, the firmware update framework proposed by Lee and Lee does not suitable for heterogeneous IoT ecosystem.

Another application of blockchain technology to update the firmware of IoT device was proposed by Boudguiga et al. in [11]. In this scheme, each IoT device is required to periodically poll random node to check the available firmware version. When a device manufacturer releases a new version of firmware, the newly released firmware must be verified first by peer nodes in the blockchain network through consensus protocol. If an IoT device from a specific manufacturer wants to perform the firmware update process, the corresponding device must create a transaction of firmware update request. In the scheme proposed by Boudguiga et al., the IoT device could not download the firmware binary from its associated manufacturer unless the associated firmware has been verified by the peer nodes in the network. In addition, all nodes in the blockchain network are required to store all firmware binaries that have been published in the blockchain network.

III. Proposed Framework Design

A firmware update framework that utilizes blockchain network is proposed in this paper. In the proposed firmware

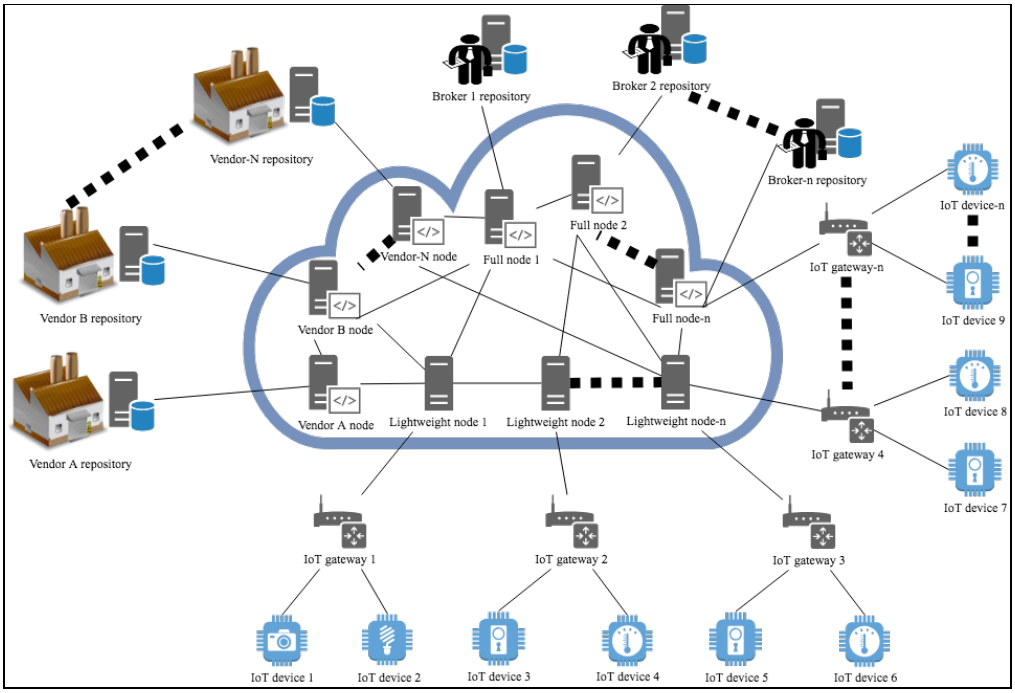

Fig. 1. The proposed system architecture for blockchain-based firmware update framework.

update framework, the device manufacturer distributes a new version of firmware by publishing information of the corresponding firmware to the blockchain network. Fig. 1 shows the proposed system architecture of blockchain-based firmware update framework.

The proposed firmware update framework consists of six entities as follows:

- Vendor repository: a firmware repository owned and managed by device manufacturer in order to store the firmware binaries and to provide information related for the corresponding firmware.
- Broker repository: a firmware repository owned and managed by firmware broker in order to provide the indirect firmware update service to IoT devices. Broker repository also stores firmware binaries and provide information related to the corresponding firmware.
- Full node: a node in blockchain network owned by either device manufacturer or firmware broker. If the full node is owned by device manufacturer (vendor node), it could create smart contract for a new firmware update and smart contract for firmware replication. Full node also acts as miner of the blockchain network and actively verifies all transaction in the blockchain network.
- Lightweight node: a node that connects IoT gateways to the blockchain network. Lightweight nodes receive and execute the firmware update contract from full nodes.
- IoT gateway: a gateway for IoT devices such a Wi-Fi router in smart home. Information of registered IoT devices is stored in the IoT gateway (e.g. device manufacturer, device model, and the installed firmware version). Each gateway is identified by a public address (or wallet address) in a node.
- IoT device: sensors or embedded devices.

A. *Assumptions*

Assumptions for the proposed protocol are as follows:

1. Each device manufacturer has at least one vendor node in the blockchain network and one vendor repository to store the firmware binaries and the corresponding information.
2. Each firmware broker has at least one broker node in the blockchain network and one broker repository to store the firmware binaries and the corresponding information.
3. Vendor repository is connected with vendor node through a secure channel.
4. Broker repository is connected with broker node through a secure channel.
5. Lightweight node does not participate in the mining process and only needs to synchronize the data stored in the node's local ledger with the data from public ledger data.
6. Each IoT device is connected to IoT gateway.

B. *Proposed Framework*

The proposed firmware update framework is designed based on the architecture of Ethereum. There are four processes in the proposed framework: the creation of contract

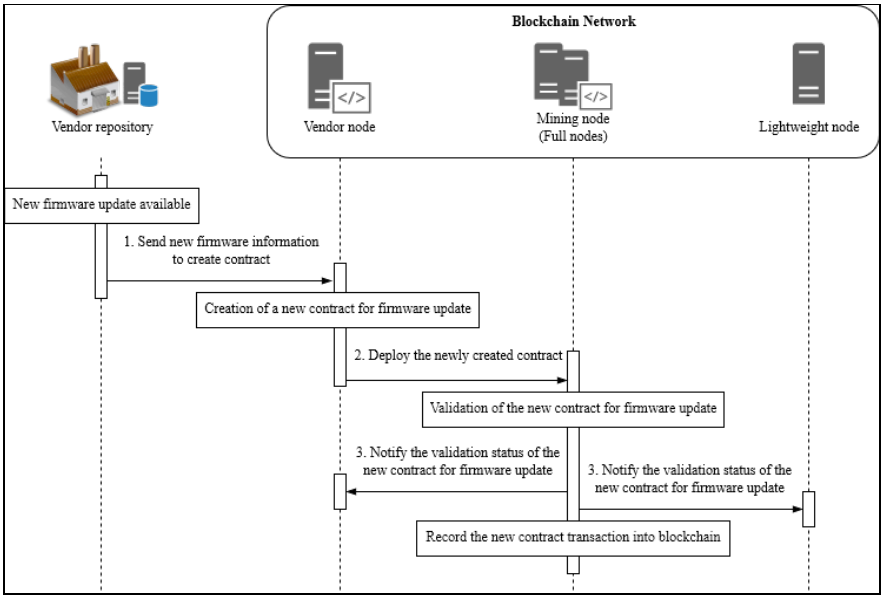

Fig. 2. The process flow for creating a new firmware update contract.

for firmware update, the process of direct firmware update, the creation of contract for firmware replication, and the process of indirect firmware update. In order to verify all firmware and created smart contract involved in the network, the verification mechanism of blockchain network through the use of consensus protocol is used.

Fig. 2 shows the process flow during the creation of smart contract for a new version of firmware. The processes are explained as follows:

1. Device manufacturer develops and releases a new version of firmware for a specific IoT device. The binary of the newly developed firmware and the information related with the firmware are stored in the vendor repository. Then, the information regarding the associated firmware will be sent to the vendor node.
2. The vendor node creates a new smart contract of firmware update based on the received firmware information. After the contract is created, the corresponding vendor node deploys the newly created smart contract to peer nodes (especially the full nodes) for verification process.
3. All the other full nodes, which act as miner nodes, will perform the mining operation to verify and validate the recently deployed smart contract in the blockchain network. Once the smart contract has been verified and validated, the status of the associated smart contract will be changed and the corresponding verifier node will notify all other nodes regarding the verification result of the corresponding smart contract.
4. The miner nodes collect all firmware update contracts that have been verified and store them into the public ledger of blockchain network.

After a firmware update contract is verified and stored in the public ledger of blockchain network, the firmware binary correspond with the contract could be distributed to the suitable IoT devices. The process flow for the direct firmware distribution is shown in Fig. 3 and explained as follows:

1. After a new contract for firmware update has been verified and stored in the blockchain public ledger, gateway node (either lightweight node or full node) could send a notification to the associated IoT gateway regarding the newly deployed firmware version.
2. The IoT gateway checks the information of the deployed firmware and compares it with the specification of all IoT devices associated with the gateway. If the gateway finds any IoT device that matched with the requirement of the firmware update from the received firmware update contract, then the gateway will send a request to the associated gateway node for detailed information to download the firmware's binary file.
3. The gateway node receives the firmware update request from the associated gateway. Afterward, the gateway node checks whether the requirements on the firmware contract are fulfilled with the received request. If the requirements are fulfilled, then the gateway node sends the URI of the requested firmware binary to the requesting IoT gateway.

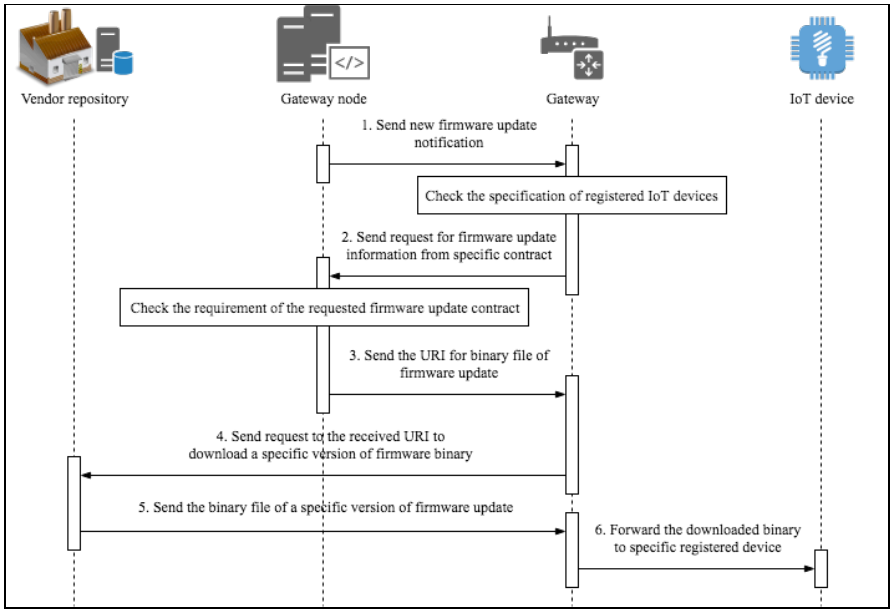

Fig. 3. The process flow for direct firmware update from the repository of device manufacturer.

4. After the IoT gateway receives the URI of the requested firmware binary, the gateway sends a request to the vendor repository to download the requested firmware binary.
5. The vendor repository receives the request sent by the corresponding IoT gateway. Afterwards, the vendor repository sends the requested binary file to the requesting IoT gateway.
6. After the IoT gateway has finished downloading the binary file of a firmware, the gateway forwards the firmware binary file to the associated IoT device.

In the proposed framework, the IoT devices could also download a specific version of firmware binary from the repository of firmware broker. In order for firmware broker to be able to provide firmware binary from the corresponding broker repository, a firmware replication contract needs to be made beforehand. The process flow for the creation of firmware replication contract is shown in Fig. 4 and explained as follows:

1. The firmware broker repository sends a request to its corresponding broker node to check the available version of firmware.
2. The broker node responses by sending list of available firmware version to the broker repository.
3. The broker then sends the information regarding a specific version of firmware to be replicated to the associated broker node.
4. After receiving the information of specific version of firmware to be replicated, the broker node creates a contract proposal for firmware replication to the specific vendor node.
5. After the vendor node receives the contract proposal from the broker node, a notification regarding the contract proposal will be sent from the vendor node to the associated vendor repository.
6. After receiving the notification of contract proposal, the device manufacturer decides whether to accept the firmware replication proposal from the corresponding firmware broker. If the device manufacturer accepts the proposal, then the vendor repository sends the acknowledgment to the vendor node.
7. Once the vendor node receives the acknowledgment from the vendor repository, the vendor node creates a firmware replication contract. Afterward, the newly created firmware replication contract is deployed to the peer nodes for verification.
8. All the other full nodes perform the verification and validation on the corresponding firmware replication contract. After the contract is verified and validated, the verified firmware replication contract will be recorded in the public ledger of blockchain. Then, verifier miner node sends notification regarding the validation status of the newly verified firmware replication contract to all nodes in the blockchain network.
9. When the broker node receives the notification status of the validated firmware replication contract, it will forward the notification to the broker repository. Then the broker repository will wait to receive the associated firmware binary from the vendor repository.

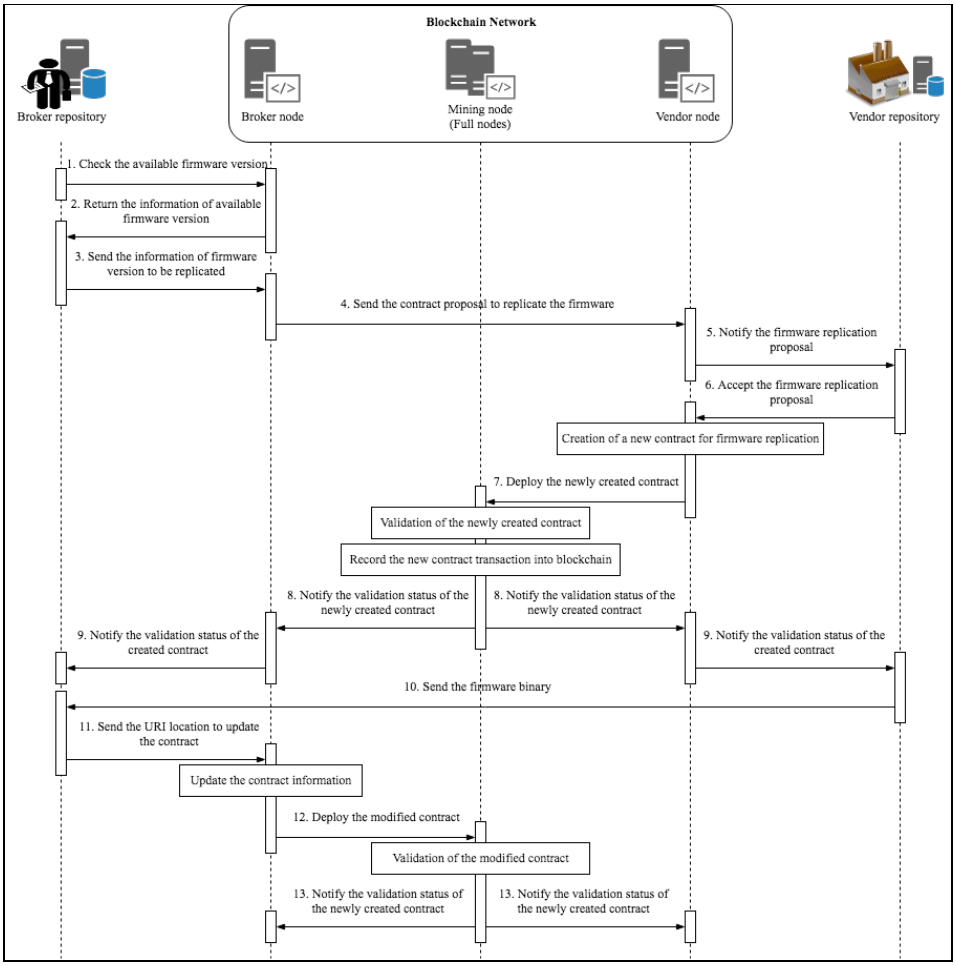

Fig. 4. The process flow for creating a new firmware replication contract.

10. Meanwhile, after the vendor node receives the notification status of the validated firmware replication contract, it will forward the notification to the vendor repository. After receiving the notification, the vendor repository will send the specified firmware binary to the associated broker repository.
11. After receiving the firmware binary, the broker repository will send the URI location of the associated firmware binary to the associated broker node.
12. The broker node updates the corresponding firmware replication contract and adds the URI location of the firmware binary into the contract. Afterward, the broker node deploys the modified contract to the peer nodes for verification process.
13. Once the other full nodes receive the modified firmware replication contract, the full nodes verify and validate the modified contract. After the modified firmware replication contract has been verified and validated, the verifier node sends notification to all other nodes in the network.

After a firmware replication contract is verified and stored in the public ledger of blockchain network, the firmware binary correspond with the contract could be downloaded by the suitable IoT devices. The process when an IoT device downloads a firmware binary from a firmware broker repository is called indirect firmware update process in the proposed framework. The process flow for the indirect firmware update is shown in Fig. 5 and explained as follows:

1. An IoT device sends a request for a new version of firmware to the associated gateway.
2. The gateway sends the specifications of the IoT device to the gateway node.
3. The gateway node searches for the latest firmware version based on the received specification of target IoT device. When the gateway node finds the latest version of firmware for the associated IoT device, the URI for downloading the associated firmware binary and the firmware information are sent to the corresponding gateway.
4. The gateway send request to the received URI to download the specific version of firmware binary from the broker node.
5. The broker node sends the requested firmware binary to the requesting gateway.

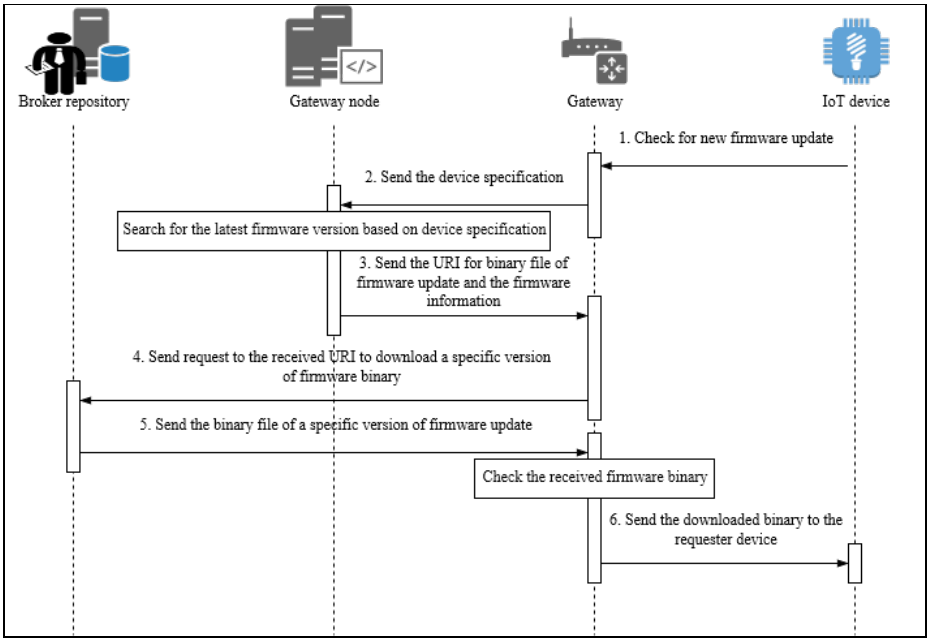

Fig. 5. The process flow for indirect firmware update mechanism.

6. After receiving the firmware binary, the gateway checks the received firmware binary with the firmware information. Afterward, the gateway forwards the firmware binary to the associated IoT device.

IV. Discussion

In this section, we provide features comparison between our proposed firmware update framework with existing blockchain-based firmware update frameworks. To the best knowledge of the authors, there are only two works on firmware update mechanism using blockchain technology by the time of the writing of this paper. Table I shows features comparison between our proposed blockchain-based firmware update framework with the blockchain-based firmware update frameworks proposed in [10], [11].

TABLE I. Feature comparison between the proposed firmware update framework and existing frameworks

Feature	Proposed Framework	Lee and Lee's Framework [10]	Boudguiga et al.'s Framework [11]
Update method	Hybrid	Pull method	Pull method
Peer-to-peer firmware sharing	Only allow the firmware to be distributed from the repository of vendor and/or broker	All nodes in the network could share the firmware binary file	All nodes in the network could share the firmware binary file
IoT device ecosystem	Multi-vendors and heterogeneous IoT device network	Single vendor and homogeneous IoT device network	Multi-vendors and heterogeneous IoT device network
Blockchain technology	Ethereum	Based on Bitcoin technology	Based on Bitcoin technology

Both firmware update framework proposed in [10], [11] use pull method, in which the IoT device needs to inquire the latest version of firmware to the network periodically. As the firmware update mechanism in [10], [11] uses blockchain technology, the drawback of pull method in which it would cause numerous network load on the firmware distribution center is mitigated to the other nodes in the blockchain network. In our proposed firmware update framework, hybrid method which combines both push and pull method is adopted. The push method is adopted in our proposed firmware update mechanism since it could reduce the attack window time from attackers by instantly distribute a newly released firmware to the targeted IoT devices.

In the firmware update framework proposed in [10], [11], all nodes in the blockchain network could share the firmware binary to the requesting IoT device. The framework designed by Lee and Lee in [10] is suitable to be implemented in a homogeneous IoT device network that comes from one similar vendor. However, heterogeneous IoT device network and multiple vendor is supported in the Boudguiga et al.'s framework. As all nodes in the blockchain network are required to store all the ledger data of the corresponding blockchain network, both frameworks in [10], [11] require the node to have more resource especially in storage. In order to leverage the storage resource constraint on the blockchain nodes, the firmware binary is only allowed to be distributed from the repository of vendor and/or broker in our proposed framework. In addition, our proposed framework is designed to support the firmware update ecosystem for heterogeneous IoT device network with multiple vendors.

Another difference between our proposed firmware update framework and the existing two other frameworks lies in the

blockchain technology used to build the framework. The two existing firmware update frameworks were designed based on the concept of Bitcoin technology. Every record of firmware update request sends by IoT device to the other nodes in the blockchain network is considered as a transaction in the blockchain network. In addition, each transmission of firmware binary file from one node to the requested node (IoT device) is considered as another transaction in the blockchain network too. Moreover, it takes around 10 minutes in order to verify a transaction and to create a block in the Bitcoin network. In contrast, our proposed protocol uses Ethereum to design the firmware update framework. Each version of firmware that is ever released by the device manufacturer is recorded in smart contract. Moreover, every distribution of firmware binary from the vendor and/or broker's repository in our proposed protocol will be verified with the corresponding smart contract. Ethereum technology is chosen because it has faster block creation speed compared to the Bitcoin technology, which is around 10 – 19 seconds in average. Considering that time is an important factor to distribute a newer version of firmware to the end devices, it is more preferable for every newly released firmware to be verified as soon as possible.

V. Conclusion

As the usage of IoT devices increase, various issues related with the management of IoT devices emerge. One of the problems is related with the distribution of authentic version of firmware from device manufacturer to the IoT devices. Centralized network architecture was used in the traditional firmware update mechanism to deliver the firmware to the target IoT device. However, the centralized network architecture might not suitable anymore considering the rapid growth of IoT devices application.

In this paper, a firmware update framework based on the blockchain technology is proposed. The blockchain technology is utilized to securely verify the version of firmware and to verify the integrity of the distributed firmware binary during the update process. Two firmware update mechanisms are designed for the proposed framework namely direct and indirect firmware update mechanism.

There are four processes in the proposed firmware update framework. The first process is the creation of firmware update contract. In the first process, each vendor is required to create a firmware update contract for each released version of firmware, and the contract must be deployed to the blockchain network for verification process. Then, the Peer nodes verify the newly deployed firmware update contract through consensus protocol. The second process is the direct firmware distribution process from the vendor repository to the target IoT device. The third process is the process to create firmware replication contract. The firmware broker is required to create firmware replication contract in order to provide the indirect firmware update service. The fourth process is the indirect firmware update process, in which the firmware binary is distributed from a broker repository to the target IoT device. In the direct and indirect firmware update process, secure verification mechanism is applied to securely distribute the firmware binary from the repository of either device manufacturer or broker to the target IoT device.

For future works, a robust and lightweight security protocol will be designed based on the proposed firmware update framework design. In addition, a prototype for the proposed framework is being developed to test and verify the proposed framework design. Based on the experiment result, we hope that the proposed framework design could accommodate secure IoT ecosystem (including the firmware distribution process) while preserving the privacy of device owner.

Acknowledgment

The authors gratefully acknowledge the support from TWISC and Ministry of Science and Technology, Taiwan, under the Grant Numbers MOST 105-2221-E-011-080-MY3, MOST 107-2218-E-011-012, and MOST 106-2218-E-011-003.

References

[1] R. van der Meulen, "Gartner Says 8.4 Billion Connected 'Things' Will Be in Use in 2017, Up 31 Percent From 2016," 07-Feb-2017.

[2] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Futur. Gener. Comput. Syst.*, vol. 82, pp. 395–411, May 2018.

[3] D. Miessler and C. Smith, "OWASP Internet of Things Project - OWASP." [Online]. Available: https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Vulnerabilities. [Accessed: 10-Apr-2018].

[4] A. Cui, M. Costello, and S. J. Stolfo, "When Firmware Modifications Attack : A Case Study of Embedded Exploitation," *20th Annu. Netw. Distrib. Syst. Secur. Symp.*, 2013.

[5] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[6] N. Kshetri, "Blockchain's roles in strengthening cybersecurity and protecting privacy," *Telecomm. Policy*, vol. 41, no. 10, pp. 1027–1038, Nov. 2017.

[7] Ethereum, "Introduction To Smart Contracts," 2016. [Online]. Available: http://solidity.readthedocs.io/en/latest/introduction-to-smart-contracts.html. [Accessed: 10-Apr-2018].

[8] M. A. Prada-Delgado, A. Vazquez-Reyes, and I. Baturone, "Trustworthy firmware update for Internet-of-Thing Devices using physical unclonable functions," in *2017 Global Internet of Things Summit (GIoTS)*, 2017, pp. 1–5.

[9] K. Doddapaneni, R. Lakkundi, S. Rao, S. G. Kulkarni, and B. Bhat, "Secure FoTA Object for IoT," in *2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops)*, 2017, pp. 154–159.

[10] B. Lee and J.-H. Lee, "Blockchain-based secure firmware update for embedded devices in an Internet of Things environment," *J. Supercomput.*, vol. 73, no. 3, pp. 1152–1167, Mar. 2017.

[11] A. Boudguiga *et al.*, "Towards Better Availability and Accountability for IoT Updates by Means of a Blockchain," in *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2017, pp. 50–58.