

Blockchain-based secure firmware management system in IoT environment

Minsung Son*, Heeyoul Kim**

*Department of computer Science and engineering, Kyonggi University, Korea

sns2831@gmail.com, heeyoul.kim@kyonggi.ac.kr

Abstract— Following the recent increase in smart factories and smart cities and the fourth industrial revolution, the number of IoT devices has been increasing and accordingly, the security of IoT devices has also become an important issue. However, due to reasons such as the low performance of IoT devices, the currently existing security solutions can be hardly applied and even if safe firmware is supplied, security problems may arise due to attacks such as man-in-the-middle attacks and roll-back attacks. To resolve such problems, we propose a new firmware management architecture using blockchains and IPFS (The InterPlanetary File System). Using IPFS can ensure the integrity of the firmware. However, IoT device version management and IPFS URL integrity cannot be guaranteed. Therefore, IoT devices are enabled to be provided with firmware information and updated through blockchain networks to ensure the integrity of the IPFS URL, and an IoT devices management function is provided. The proposed system enables safe firmware distribution and update, and IoT devices' security level is expected to be higher. Through the proposed system, safe firmware distribution and update will become possible and IoT device security levels are expected to be enhanced.

Keywords— Blockchain, Firmware, Security, IoT, Hyperledger

I. INTRODUCTION

According to the 'smart factory industry and market trend' [1] report published by the Korea Technology Transfer Centre for National R&D Programs in 2018, the global smart factory demand market is expected to grow at an average annual rate of 8% until 2020, and the device and ICT markets that constitute the smart factory supply market are expected to grow at average annual rates of 7.8% and 8.1%, respectively.

In addition, the importance of IoT device security has been increasing to the extent that the theme of the CODEGATE, which is global information security contest in 2018, was 'a smart city completed with smart security' and most of the contents of related sessions were related to the hacking and security of IoT devices [2], [3]. In the CODEGATE sessions, most IoT device vulnerabilities were said to be caused by vulnerable firmware and the importance of firmware integrity and version management was emphasized.

However, most IoT devices do not have high performance because they use small devices. Therefore, they have many restrictions on the application of existing security solutions that require massive amounts of computation.

In order to solve this problem, we propose a new architecture that receives safety information from firmware providers using reliability and transparency which are

characteristics of blockchains, and can be provided with faultless firmware using IPFS

II. BACKGROUND

A. Blockchain

The blockchain [4] technology is one of Distribute Ledger Technologies (DLTs), which guarantees the integrity of stored data and supports data distribution [5]. The blockchain technology makes transactions that have occurred over a certain period of time into a block and connects the block to blocks that were made before so that transactions cannot be falsified. Blocks are connected by adding the hash value of the previous block to the current block, and the hash value plays the role of CHECKSUM in this case. Therefore, falsifying the connected blocks means that all blocks before and after the relevant blocks should be modified and this can guarantee irreversibility because it requires very difficult calculations.

Attempts have been made to combine blockchains in the public blockchain network environment to ensure firmware integrity in the IoT environment [6]. However, IoT device disk storage problems may occur because all transactions must be stored in the IoT device and the risk of 51% attacks may also exist because the PoW consensus method is used using IoT devices with relatively poor performance.

B. Hyperledger Fabric

The Hyperledger Fabric [7] is one of the Linux Foundation's hyperledger projects and is an open source based private blockchain platform being developed under the leadership of IBM. In the Hyperledger Fabric, one organization that constitutes a blockchain network is called an organization. Each organization has its own set of peers and uses these peers to form a blockchain network. In addition, the blocks created in this blockchain network are sent to peers in each organization and stored. This structure brings about the advantage of being capable of verifying one's data through the verification of each other's block and enables distributed storage of data

The Hyperledger Fabric stores data in transactions in the form of key-values and the transactions are stored in the Ledger. Each peer has Ledger and the Ledger can be divided into two states, called world state and blockchain, respectively. The world state is a database that stores only the latest data of the blockchain. The Ledger has an architecture where when

the blockchain data has been updated, the world state data are also updated into the latest data accordingly.

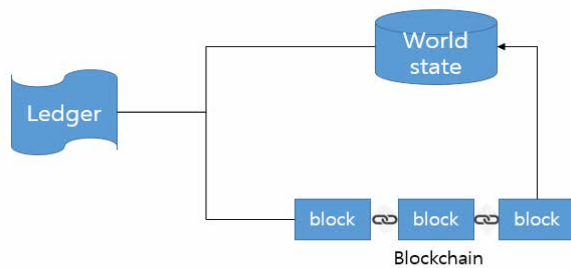


Figure 1. Structure of Ledger

The Hyperledger fabric collects transactions using the Ordering System and creates blocks to distribute the transactions. Here, collecting transactions literally means storing transactions sent by peers in order of arrival at the orderer. When certain conditions have been satisfied, the stored transactions are made into blocks and propagated to peers belonging to the network. Therefore, unlike the existing blockchain system, the Hyperledger Fabric does not have any consensus such as PoW.

Unlike public blockchain networks, private blockchain networks do not allow unspecified many individuals to access the network. Therefore, private blockchain networks use certificates and PKI to verify their participants and the certificates and PKI are managed using the Membership Service Provider (MSP) of the Hyperledger Fabric.

The Hyperledger Fabric provides Chaincode under a concept similar to the smart contract of the Ethereum. Through the foregoing, the state of the Ledger can be changed, Ledger information can be brought in, and different roles can be assigned to peers by installing different Chaincodes in individual peers. Using Chaincodes enables access to data in the Ledger and reading and writing. In this case, Hyperledger Fabric provides Query and Invoke as methods to conduct the above acts.

C. IPFS (The InterPlanetary File System)

IPFS [8] is a distributed file distribution system that plays a role similar to the role of the web. It is based on a peer-to-peer system and is similar to the Bit Torrent. Its entire structure can be regarded to be a single swarm unlike a Bit torrent, in which each file has a swarm. IPFS supports version management through Merkel DAG and supports self-certified names through the SFS naming scheme. That is, the name of the file per se becomes the path, and the file per se becomes a CHECKSUM for itself so that the integrity of the file can be ensured.

III. PROPOSED SYSTEM

The architecture proposed in this paper constructs a blockchain network using the Hyperledger Fabric and verifies file integrity using IPFS. If we were to solve our problem using only block chains, we would have to add the binary of the firmware to the transaction. This means that all nodes must

have all versions of the firmware binary, and the size of the transaction will also increase, which will cause the network traffic to increase.

When IPFS is used, the integrity of files can be ensured. However, the integrity of the URL that will be used when IoT devices download files cannot be ensured and whether the latest firmware is installed in the current device and whether the file being downloaded is the latest firmware version cannot be identified. To solve this problem, we will use blockchain to manage device state information and download URLs thereby ensuring the integrity of the information.

The architecture environment is as follows.

- Firmware provider: Those that provide firmware. This means firmware manufactures or certain organizations that supply firmware.
- Firmware requestor: Those that request firmware. This means places where IoT devices exist such as Smart Factories and Smart Cities.

Each of firmware providers and firmware requestors is a blockchain network, which is composed of many peers in each organization. In this case, the peers comprise separate devices other than IoT devices because they require performance at least at a certain level. In addition, peer for IPFS are also necessary and these peers can be separate ones or can be composed of several peers selected from among those peers that constitute the blockchain network. The IPFS peers become members that constitute the IPFS and also play the role of the router used by IoT devices receive files from the IPFS network.

The locations and organization of the peers can be determined by the blockchain network designer. For instance, in the case of a Smart City, each building in a complex can be a peer and in the case of a Smart Factory, each zone can be regarded to be a peer, and multiple organizations can be created by grouping several peers together. In this paper, we assume that two organizations exist and IoT devices can access the peers in the organizations to which they belong to participate the blockchain network and receive information.

A. Network Setting

The networks in this architecture are divided into two types.

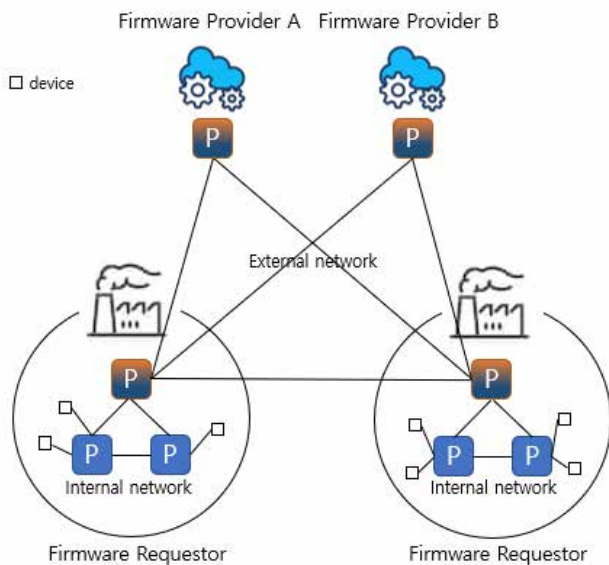


Figure 2. Network Overall

1) Internal network: This is a blockchain network constructed by the firmware requestor. It is composed of organizations and intended to provide firmware information to IoT devices belonging to each firmware requestor and manage environment information about IoT devices.

2) External network: This is a blockchain network composed of some of the peers of the firmware requestor and peers of the firmware provider. It exists to propagate information on firmware.

3) Broker: means a peer belonging to an external network while belonging to an internal network too. When the external network's Ledger has been updated, this plays the role of inputting the updated data to the internal network's Ledger.

B. Device registration

To add a new IoT device to the blockchain network of the firmware requestor, a process of registration by the manager is necessary.

The manager stores the state information of the current IoT device in the Ledger along with the unique identification number such as the serial number through the chaincode for registration of information on the IoT device. Thereafter, the manager enters information on the peers of the blockchain network and information on the IPFS peers into the IoT device, and creates a private key and a public key that enable participation in the blockchain network through FabricCA (Certificate Authority). When these keys have been provided to the IoT device, the IoT device becomes to have the right to access the blockchain network, and can be managed by the blockchain network when connected to the peers using the keys provided.

C. Firmware distribution

The process through which the firmware provider distributes new firmware is as follows.

First, the new firmware is uploaded to the IPFS network to obtain the URI information about the new firmware. Thereafter, the device to be updated, version information, and the URI information obtained as described above are registered in the Ledger through the chaincode for registration of new firmware information. Then, the above pieces of information are updated in the Ledger of the Broker of the Firmware requestor. Through the foregoing, the firmware information provided by the firmware provider is transmitted to the peers that constitute the internal network while maintaining the integrity.

When the Ledger of the external network has been updated, the Broker recognizes the relevant fact, fetches information from the Ledger of the external network through invoke, and registers it in the Ledger of the internal network.

Then all the peers in the internal network become to be able to have the latest firmware information, and the IoT devices connected to the peers can get the latest firmware through the peers.

D. Firmware check & update

IoT devices can communicate with the peers using an application created via fabric-sdk. In this case, there are two methods of retrieving data by communicating with the peers: query and invoke.

When Queries are used, data are retrieved from the world state and faster searches are possible because the entire data in the blockchain are not explored. However, since the relevant data are not verified either, the data obtained through queries become unreliable if the data is falsified after the world state (stored in couchdb, etc.) is hacked even if the peer is not hacked.

On the other hand, when Invoke is used, the verification of the chaincode being executed can be requested to appropriate peers. Therefore, Invoke is mainly used when the Ledger is updated. However, there is problem as a delay time exists until the result is output.

Since the delay time of Invoke does not affect the entire process significantly in our architecture, when important data are fetched from the Ledger, invoke is be used instead of query to verify the relevant data.

IoT devices connected to peers send queries to the connected peers at intervals of a certain time to inquire whether they have the latest firmware. If they do not have the latest firmware, they get a URI that enables them to download the latest firmware from the peer through invoke and leave a IoT device firmware requesting log in the Ledger. This enables the IoT devices to identify the update situation of each device later.

Since the IoT devices have IPFS peer information in the registration process, they can obtain the latest firmware IPFS URL that uses the IPFS peer as the router by combining the IP address of the IPFS and the URI of the firmware and become

to be capable of proceeding with update through the foregoing.

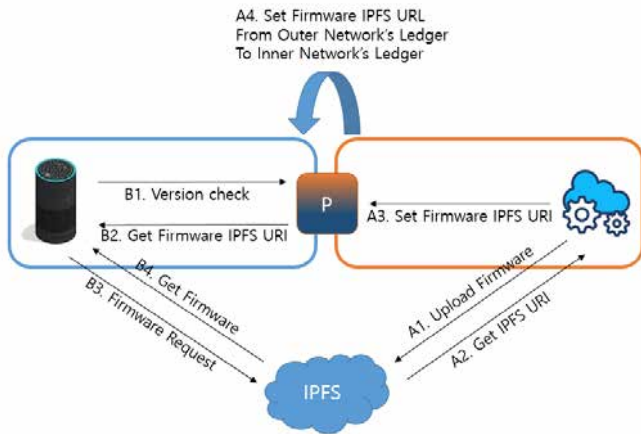


Figure 3. Overall operation process

E. Implementation

To implement the above architecture, we made three different blockchain networks (one firmware provider, two firmware requestors) and configured each of the blockchain networks to have four peers and two organizations and configured the IPFS to be connected to the main net. The operating systems of all peers and IoT devices were based on ubuntu.

From	November 6, 2018 6:07 AM	To	November 7, 2018 6:07 AM	Select Orgs	Search	Reset	Clear
Creator	Channel Name	Tx Id	Type	Chaincode	Timestamp		
Org1MSP	mychannel	c4c9ba...	ENDORSE...	mycc	2018-11-06T21:06...		
Org1MSP	mychannel	e4864b...	ENDORSE...	mycc	2018-11-06T21:06...		
Org1MSP	mychannel	1d88b5...	ENDORSE...	mycc2	2018-11-06T21:05...		
Org1MSP	mychannel	c04797...	ENDORSE...	mycc2	2018-11-06T20:58...		
Org1MSP	mychannel	b5f72b...	ENDORSE...	mycc2	2018-11-06T20:57...		
Org2MSP	mychannel	48451b...	ENDORSE...	lsc	2018-11-06T20:56...		
Org2MSP	mychannel	a13494...	ENDORSE...	lsc	2018-11-06T20:56...		

Figure 4. Transactions list

Creator MSP: Org1MSP

Endorser: ["Org1MSP"]

Chaincode Name: mycc2

Type: VALID

Time: 2018-11-06T21:05:03.000Z

Reads:

```

root: {} 3 items
  0: {} 2 keys
    chaincode: "mycc"
    set: {} 1 item
      0: {} 2 keys
        key: "test_device"
        version: {} 2 keys
  1: {} 2 keys
    chaincode: "mycc2"
    set: {} 1 item
      0: {} 2 keys
        key: "test_device"
        version: {} 2 keys
  2: {} 2 keys
    chaincode: "mycc2"
    set: {} 1 item
      0: {} 2 keys
        key: "test_device"
        version: {} 2 keys

```

Writes:

```

root: {} 3 items
  0: {} 2 keys
  1: {} 2 keys
  2: {} 2 keys

```

Figure 5. Contents of the transaction

```

Transaction proposal was good
status - 200, message = ""
result : https://...
transaction commit Complete.

```

Figure 6. Result of the transaction

The Hyperledger Explorer, which is one of the Hyperledger projects, was used to observe transactions when version information was sent to peers in an IoT Device environment. You can see that transactions are accumulated every time device check the version through Invoke via Hyperledger Explorer. (Figure 4) and on reviewing the inside of the transactions, the details of identification of IoT device versions under the title "test_device" can be seen (Figure 5). In addition, Also, when the device sends a transaction containing its firmware version information to the blockchain network to check its firmware version, if it is determined by the chaincode that it is not the latest version, the view of transmission of the IPFS URL can be seen. The version check transaction uses invoke to prepare the worldstate database tampering of the peer. Therefore, for this reason, the transaction is stored in a block chain. so we wrote the code to output a "transaction commit complete" message. (Figure 6).

IV. CONCLUSION

The proposed architecture improved data integrity by using blockchain to provide distributed database with guaranteed reliability to the firmware provider and the firmware requestor thereby preventing data falsification when information is transmitted and enabled firmware version management through continuous communication between IoT devices and blockchain networks.

Therefore, when the proposed architecture is used, Because IPFS guarantees the integrity of the firmware, and the block chain ensures the integrity of the IPFS URL, safe firmware update is possible because firmware falsification through

firmware roll-back or man-in-the-middle attacks, which is a problem in existing firmware systems, can be prevented. In addition, the firmware requestor manager can manage devices through IoT device update logs.

ACKNOWLEDGMENT

This research was supported by Ministry of Science ICT Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2018R1C1B6002903).

REFERENCE

- [1] Lee Hyeon-gyu, Smart Factory Industry and Market Trends, Korea Technology Transfer Centre for National R&D Programs .2018
- [2] Sanghyoen Park, Sungjun Jo, "Smart Home Hacking in Real World.", 2018. [Online].Available: <http://codegate.org/assets/kr/files/Smart%20Home%20Hacking%20in%20Real%20World.pdf>
- [3] Changhyun Park, Hwiwon Lee, "IoT Hub Vulnerability Analysis and Home Service Security Threats and Countermeasures.", 2018. [Online].Available: <http://codegate.org/assets/kr/files/IoT%20Hub%20Vulnerability%20Analysis%20and%20Home%20Service%20Security%20Threats%20and%20Countermeasures.pdf>
- [4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system, 2009," 2012. [Online]. Available: <http://www.bitcoin.org/bitcoin.pdf>
- [5] Nir Kshetri, "Can Blockchain Strengthen the Internet of Things", IEEE IT Pro, July/August 2017.
- [6] Lee, B., & Lee, J. H. (2017). Blockchain-based secure firmware update for embedded devices in an Internet of Things environment. The Journal of Supercomputing, 73(3), 1152-1167.
- [7] (2018)Hyperledger-fabric read the docs website.[Online].Available:https://hyperledger-fabric.readthedocs.io/en/release-1.3/
- [8] J. Benet, "IPFS - Content Addressed, Versioned, P2P File System." CoRR, vol. abs/1407.3561, 2014. [Online].Available: <https://github.com/ipfs/papers/raw/master/ipfs-cap2pfs/ipfsp2p-file-system.pdf>



Minsung Son is in the undergraduate course of Computer Science Department, Kyonggi University , His major research interests include security and blockchain.



Heeyoul Kim received the B.E. degree in Computer Science from KAIST, Korea, in 2000, the M.S. degree in Computer Science from KAIST in 2002, and the Ph.D. degree in computer science from KAIST in 2007. From 2007 to 2008, with the Samsung Electronics as a senior engineer. Since 2009 he has been a faculty member of Department of Computer Science at Kyonggi University. His major research interests include cryptography, security and blockchain.