Received February 4, 2020, accepted February 18, 2020, date of publication February 24, 2020, date of current version March 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2975920

Blockchain-Based Distributed Firmware Update Architecture for IoT Devices

SEOYUN CHOI AND JONG-HYOUK LEE, (Senior Member, IEEE)
Protocol Engineering Laboratory, Department of Software, Sangmyung University, Cheonan 31066, South Korea

Corresponding author: Jong-Hyouk Lee (jonghyouk@smu.ac.kr)

This work was supported by the research grant from Sangmyung University.

ABSTRACT The Internet of Things (IoT) which creates a hyper-connected society is playing a major role in the 4th industrial revolution. The IoT is being leveraged across various fields of business globally and the number of IoT devices is causing serious security concerns. Since the firmware update of an IoT device is necessary for its lifecycle, secure firmware update of the IoT device is being brought as the first step in IoT security. The Internet Engineering Task Force (IETF) Software Updates for Internet of Things (SUIT) working group has started to specify a software update architecture for IoT devices. However, the current SUIT working group adopts a traditional client-server model to distribute firmware images, which potentially causes security risks. The current approach of the SUIT working group is unable to solve a targeting issue and an author-disappearing issue, which is suggested in this paper. Therefore, in this work, we introduce a distributed firmware update architecture based on the SUIT firmware update architecture applying blockchain. Our update architecture can prevent the issues we concern through the characteristics of blockchain, such as decentralization, transparency, and irreversibility. The blockchain network has registration nodes that process registration of manifest and firmware image files from authors, and retrieval nodes that process downloading manifest and firmware image files. The firmware image files are stored in a distributed file system and the hash values of firmware image chunks are stored on the blockchain with manifest files. The proposed architecture in this paper enables the irreversible downloads even in the author-disappearing state and tolerant to a single point of failure.

INDEX TERMS Author-disappearing issue, blockchain, firmware update, Internet of Things (IoT).

I. INTRODUCTION

The Internet of Things (IoT) implies the status of network connection for interactions between things of human beings via embedded communication systems. The IoT is used in a very wide range of fields as it provides close connections between things and humans. Even though IoT technology is being expeditiously improved as a key technology in Industry 4.0, the security risks of IoT devices are concerned. IoT devices operate with low computing power using micro controller units (MCUs) for weight lightening, but this lightweight computing characteristic of the IoT device accompanies security limits.

A firmware update offers a new service or security patch to IoT devices. Since it is necessarily required for the IoT device, the recent cyber-attacks are focused on the firmware level. As the result, numerous researches for secure firmware updates have been conducted recently [1], [2]. The firmware update failures occur due to network issues or cyber-attacks. The supports for integrity and authentication of the firmware images thus required. The Software Updates for Internet of Things (SUIT) working group of the Internet Engineering Task Force (IETF) is on a process of creating standards for an IoT firmware update architecture and firmware manifest file. The traditional client-server model that uses a centralized database which is an easy and high-value target for various kinds of attacks. An attack on a server can violate the availability of all data stored on the server. Also, firmware updates for IoT devices are vulnerable against an author-disappearing issue that the IoT device manufacturers or firmware vendors are unable to provide firmware updates in time due to cyber-attacks or disappearing due to their funding problems.

Blockchain ensures that every node stores the same data based on an append-only distributed ledger, which provides integrity, decentralization, and irreversibility. In this paper, we propose a distributed SUIT firmware update architecture,

The associate editor coordinating the review of this manuscript and approving it for publication was Giovanni Pau.

which is based on blockchain. The proposed architecture is designed to provide a secure firmware update and also to address the author-disappearing issue.

The remainder of this paper is as follows: In Section II, the current SUIT architectures, blockchain, and existing blockchain-based firmware update architectures are explained. In Section III, we introduce our blockchain-based distributed firmware update architecture. Then, we present comparison results of the proposed architecture with the existing blockchain-based firmware update architecture in Section IV. In Section V, we conclude this paper.

II. RELATED WORK

Existing firmware updates are mostly executed in client-server architecture form. A server takes a charge of centralized management of the whole data in a client-server model. This leads a server to be a vulnerable target for an adversary. In addition, if excessive request from rapidly increasing IoT devices overwhelm the server, update processes can be failed. In recent years, blockchain, a distributed ledger offering decentralization, integrity and irreversibility of data has been considered as a mean to solve the limitations of client-server architecture and guarantee integrity of the firmware images.

A. BLOCKCHAIN

The concept of blockchain first appeared in the Bitcoin white paper written by Satoshi Nakamoto in 2008 [3]. Blockchain technology has gained substantial attention with a departure from the previous centralized network model. It enables that every participating node can verify, append, and store the data without any centralized entities. After the appearance of the Bitcoin, many kinds of cryptocurrencies have emerged based on blockchain [4]. The researches on security threats and mitigations of the blockchain are being actively conducted [5].

The operation process of blockchain in the Bitcoin system is as follows. First, when a new transaction has occurred, it is broadcasted to every node. Then, full nodes verify the transaction with a digital signature. After the verification, the full nodes store the transaction in their memory pools. Then, the full nodes decide one block at regular intervals through a result of an agreed consensus algorithm [6]. The result of the consensus algorithm is then broadcasted to all nodes in the Bitcoin system, and the nodes link the result (i.e., the new block) to their own ledger. In the respect that every node contributes to consensus for creating new blocks and maintaining the same distributed ledger, blockchain ensures the transparency of transactions.

Every block has an own hash value for its identification and each block is linked through a previous block hash stored in the block header. Transaction IDs are paired and hashed, constructing a merkle tree. The root hash maintained in a block header and guarantees integrity of whole transactions stored in the block. Fig. 1 shows an example of a merkle tree structure ensuring integrity of the transactions in the Bitcoin system [7].

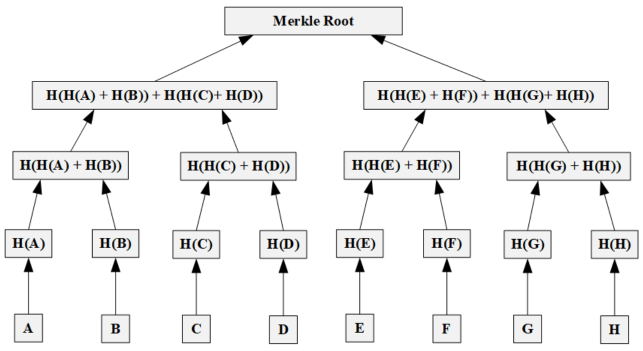

FIGURE 1. Merkle tree.

TABLE 1. Architecture types of blockchain networks.

	Public Blockchain	Consortium Blockchain	Private Blockchain
Major Operator	None	Consortium Members	Specific Operator
Trust Relationship	None	Consortium Members	Operator
Permission to Read Data	All Participating Nodes	Selective Authority or Consortium Members	Selective Authority or Specific Members or All Participating Nodes
Permission to Write Data	All Participating Nodes	Selective Authority Consortium Members	Selective Authority or Specific Members or All Participating Nodes
Permission to Create a Block	All Participating Nodes	Consortium Members	Operator
Transaction Throughput	Low	High	Very High
Network Delay	High	Low	Very Low

B. ARCHITECTURE TYPES OF BLOCKCHAIN

The blockchain platform can be divided into permissoinless and permissioned [8]. The permissionless platform includes a public blockchain (e.g., Bitcoin) in which every consisting node can participate to the network. The permissioned platform includes a private blockchain (e.g., Hyperledger Fabric) and a consortium blockchain. The private blockchain has a structure that a specific operator manages an authority service for all nodes. The consortium blockchain has a structure in which a plurality of organizations forms several private blockchain networks. Table 1 describes the features and illustrates the public, consortium, and private blockchain network.

C. SUIT FIRMWARE UPDATE ARCHITECTURE

The IETF SUIT working group is developing the firmware update architecture for IoT devices and manifest files for firmware updates at the IETF standardization organization [9], [10]. A server that firmware and manifest files are

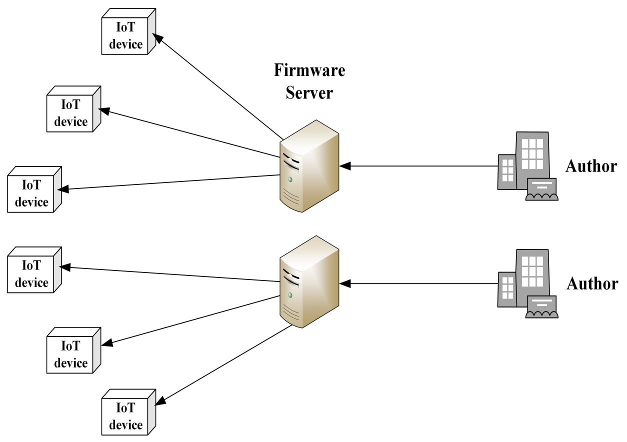

FIGURE 2. **SUIT architecture adopting the client-server model.**

uploaded is called a firmware server in the SUIT architecture. The manifest is a file that contains metadata about the firmware image. A manufacturer produces IoT devices or a vendor develops the firmware and manifest files are called authors.

The firmware update process is as follows: First, an author creates manifest and firmware file, signs them, and upload to the firmware server. The IoT device then queries the firmware server for the manifest and firmware file. When the firmware server sends the manifest and firmware file in order, the files verified each and downloaded.

If a firmware update is aborted or an IoT device fails to download the latest version of firmware, the IoT devices are considered as a vulnerable device due to lack of security services or secure patch, which may lead the IoT devices to be exposed to an attack. Therefore, the firmware update is a crucial security process for the IoT devices. However, the traditional client-server architecture leads to serious overload. Plus, if a server is attacked or goes down, all clients that need to get a firmware image from the server are exposed to threats. Figure 2 shows the SUIT architecture that adopts the client-server model.

D. BLOCKCAHIN-BAESD FIRMWARE UPDATES

Here, we introduce and analyze the existing studies of blockchain-based firmware updates.

1) SCHEME 1 BY B. Lee AND J.-H. Lee

The paper written by Lee and Lee [1] was the first paper that suggests a blockchain-based firmware update for IoT devices. The architecture suggested in the paper consists of request node, response nodes, and verifying nodes. A verification node is directly managed by a manufacturer and always aware of the latest version of the firmware. A request node sends its firmware version to a verifying node, then the verifying node compares the version of the firmware whether the requested firmware version is the latest. If the version of the requested device's firmware is not the latest, the request node requests again to a neighbor node (besides verifying nodes). Between the request node and the response node, a node that has an older version of firmware downloads the firmware from the counterpart node. The repetition of this comparison process finally leads all nodes to be updated.

However, the architecture proposed in the paper has limitations that the blockchain network consists of IoT devices as nodes and applies a consensus algorithm that requires heavy resource consumption such as Proof-of-Work (PoW). As IoT devices operate with limited resources, it is difficult to perform such a consensus algorithm and digital signature verification. In addition, the architecture only supports updates from one manufacturer, so a countermeasure for multiple vendors is demanded.

2) SCHEME 2 BY A. Boudguiga et al.

The paper was proposed in 2017 by Boudguiga *et al.* [11]. The update procedure suggested in this paper is as follows. A manufacturer selects the IoT devices to update and uploads the software to the manufacturer's web portal. The IoT devices receive an update notification from the manufacturer.

The suggested architecture offers two methods of the firmware update. The first method is that manufacturers upload the firmware binaries within the transaction and issue the transaction to the blockchain network. In this case, the size of blocks will get larger, but the firmware binaries are preserved eternally. Another method is configuring innocuousness nodes which are very reliable (e.g., nodes affiliated to a national agency or security company). In the second case, manufacturers do not put firmware binaries within the transaction, but innocuousness nodes get the firmware binaries directly from manufacturers to check integrity or bugs of the firmware file.

However, this architecture has drawbacks that the block size becomes larger as the manufacturers transmit the firmware binaries in transactions to the blockchain network.

3) SCHEME 3 BY A. Yohan AND N. Lo

The paper was proposed by A. Yohan and N. Lo in 2018 [12]. A vendor repository and a broker repository are storing firmware binaries and associated firmware information. A broker repository distributes the traffic overhead of vendor repositories. Vendor nodes, full nodes, and light nodes are constituting of a blockchain network for the firmware update. When a vendor repository announces the latest firmware to the vendor node, it deploys a smart contract requesting full nodes to verify the latest firmware. When the verification is completed, gateway nodes find the devices belong to the manufacturer through the gateways and sends URI to the gateway. The gateway downloads the firmware binary from the vendor repository and distribute the firmware binary to the devices.

However, this architecture arouses concerns about excessive traffic and costs because the manufacturers must deploy smart contracts whenever they distribute new firmware files or duplicate firmware files to the broker nodes. In addition, an update failure may occur if the gateway is compromised.

III. DISTRIBUTED FIRMWARE UPDATE ARCHITECTURE

The blockchain-based firmware update architecture brought up in this paper is based on the firmware architecture model currently being standardized by the IETF SUIT.

The proposed architecture contains a blockchain network that can distribute firmware files in a Peer-to-Peer (P2P) manner without using a firmware server [13], [14]. The manifest file is distributed through the blockchain network and contains a unique hash value of the firmware file. The firmware files are stored in a distributed file system and the hash values of firmware images are stored in the blockchain. Since IoT devices have very limited computational resources, we ensure that all verifications are performed by blockchain nodes, not IoT devices. In order to solve the bottleneck problem in the blockchain network, registration and retrieval nodes are configured in our architecture. The following entities are defined in the proposed architecture.

- Author: An author is a manufacturer produces IoT devices or a vendor develops the firmware and the manifest files. An author has to upload the manifest file and the firmware image to the blockchain network to distribute.
- IoT Device: An IoT device can be any IoT products operating with MCUs, sensors or actuators [10]. An IoT device needs the manifest file and the firmware image to be updated.
- Registration Node: A registration node is a blockchain node that handles the author's registration. An author should be registered to the blockchain network via registration nodes to upload the manifest and firmware image files.
- Retrieval Node: A retrieval node is a blockchain node that handles IoT devices downloading manifest and firmware image files via URIs contained in the manifest file.
- General Node: A general node is a normal node constituting the blockchain network besides registration nodes and a retrieval nodes.
- Blockchain Nodes: Blockchain nodes mean all of the nodes constituting the blockchain network, which are registration nodes, retrieval nodes, and general nodes. Blockchain nodes can be separated into two different types that are author nodes and non-author nodes. The author node is a blockchain node which is managed by the author, and the non-author node is a blockchain node which is a voluntary or a selected node that certified by the author. The distribution file system for firmware images are configured with blockchain nodes.

Every node constructing the blockchain network performs the designated role as a blockchain node. The firmware image files stored distributed on the blockchain network. The IoT devices are required to have a public key or a pre-shared key of the device's author. Figure 3 shows the proposed decentralized firmware update architecture based on blockchain.

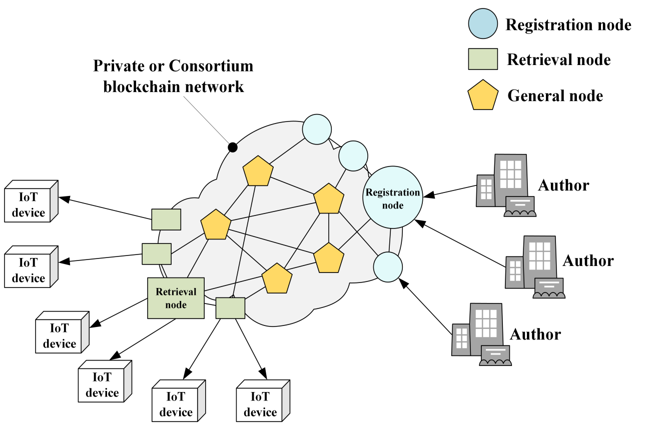

FIGURE 3. Proposed blockchain-based firmware update architecture.

A. PROCEDURE DESCRIPTION

We explain the procedure of our architecture with Figure 4.

1) AUTHOR REGISTRATION

The manifest file contains the firmware information and a URI for downloading the firmware image which is the firmware binary of a complete software or a subset of a software. When an author creates a manifest file and a firmware image, the first step is to request to the registration nodes for uploading the firmware files to the blockchain network. Thus, an author should create a registration request transaction.

2) REGISTRATION CHECK

A registration node checks whether the author already has been registered by the author ID and public key from the request transaction. If the author's request is the first time, the author ID should be registered. The blockchain nodes then validate the manifest file with the author's public key or pre-shred key in the transaction.

3) MANIFEST STORED ON THE BLOCKCHAIN

As the manifest file gets verified and stored on the blockchain, the manifest file can never be deleted or tampered; in other words, it become irreversible. Furthermore, integrity of the manifest file itself is guaranteed. A manifest integrity is crucial as it contains the unique hash value and other information about the firmware image.

4) FIRMWARE IMAGE STORED ON THE DISTRIBUTED FILE SYSTEM

Firmware images registered and verified in Step (2) are stored distributed in the P2P-based file system such as IPFS [15]. The hash values of the firmware image chunks are stored on the blockchain.

5) IOT DEVICES PERIODICALLY CHECK THE FIRMWARE VERSION

The IoT devices check the blockchain network periodically. A blockchain node responds the latest firmware version

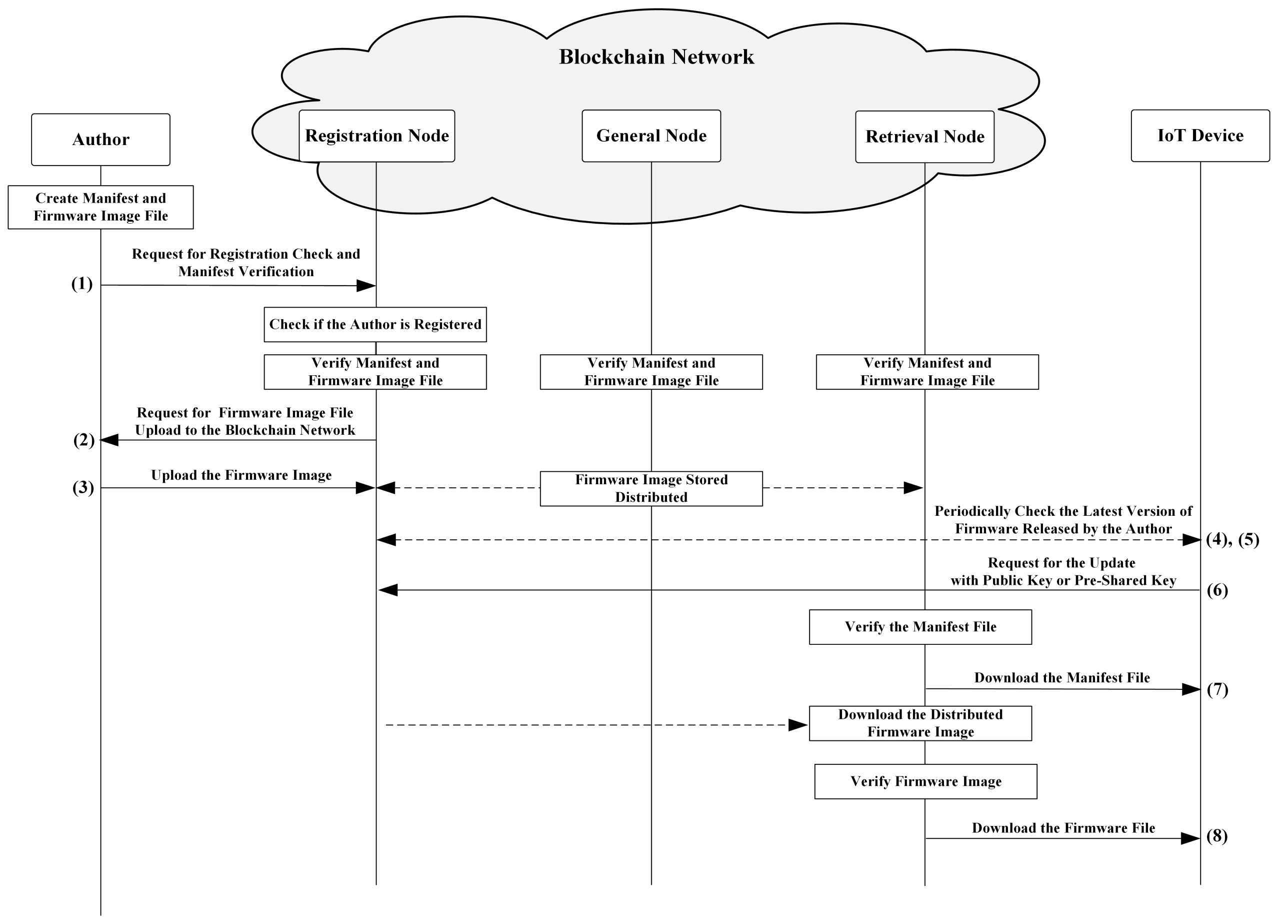
Blockchain Network
Author
Registration Node
General Node
Retrieval Node
IoT Device
Create Manifest and Firmware Image File
Request for Registration Check and Manifest Verification
(1)
Check if the Author is Registered
Verify Manifest and Firmware Image File
Verify Manifest and Firmware Image File
Verify Manifest and Firmware Image File
Request for Firmware Image File Upload to the Blockchain Network
(2)
Upload the Firmware Image
(3)
Firmware Image Stored Distributed
Periodically Check the Latest Version of Firmware Released by the Author
(4), (5)
Request for the Update with Public Key or Pre-Shared Key
(6)
Verify the Manifest File
Download the Manifest File
(7)
Download the Distributed Firmware Image
Verify Firmware Image
Download the Firmware File
(8)

FIGURE 4. Procedure of the proposed blockchain-based firmware update.

released by the device's author. Then, the IoT device compares the firmware version it has now, and the latest firmware version released by the author. If the firmware version that the IoT device owns is older, the IoT device requests for the latest manifest file to the retrieval node. Note that this process can be operated by a specific device operator such as status tracker as in existing SUIT architecture [10].

6) MANIFEST DOWNLOADS

After the IoT devices request for the manifest file, the retrieval node responds to the request by sending the manifest file that was stored on the blockchain network to the requesting IoT device. We assume that the retrieval node and the IoT device have a trust relationship so that the manifest or firmware image files sent from a retrieval node to an IoT device are securely protected.

7) FIRMWARE DOWNLOADS

The retrieval node sends the manifest file to the IoT device in Step (6) and then downloads the distributed firmware image from the neighbor nodes with the information stored in the manifest. The blockchain nodes verify the downloaded firmware image with the public key or a pre-shared key of an author. The data sent from the retrieval node to the IoT device should be securely protected.

B. TRANSACTION COMPOSITIONS

In our architecture, 8 different types of transactions are defined as follows ('A → B' implies 'sender → recipient', and ***Request*** or ***Response*** implies the type of the transaction):

(1) Author → Registration Node : ***Request***
An author requests for registration check and verification of the manifest file.

(2) Registration Node → Author : ***Request***
After the manifest verification, Registration node requests to Author to upload the firmware image.

(3) Author → Blockchain Node : ***Response***
An author uploads the firmware image to the distributed file system as a response to the request.

(4) IoT Device → Blockchain Node : ***Request***
An IoT device periodically check if the latest version of firmware is released from its author to the blockchain nodes.

(5) Blockchain Node → IoT Device : ***Response***
A blockchain node responses to the IoT device the latest version of the firmware released by the IoT device's author.

TABLE 2. Compositions constituting transactions.

Transaction Identifier	Transaction Compositions
1	Identifier, Author ID, Version of Firmware, Encrypted Manifest, Public Key or Pre-Shared Key, Hash Value of Firmware Image
2	Identifier
3	Identifier, Author ID, Encrypted Firmware Image, Hash Value of Firmware Image
4	Identifier, Author ID, Device Name
5	Identifier, Author ID, Latest Version of Firmware Image
6	Identifier, Author ID, Current Firmware of Device, Public key or Pre-Shared Key of Author
7	Identifier, Verified Manifest
8	Identifier, Verified Firmware Image

(6) IoT Device $\rightarrow$ Blockchain Node : ***Request***
If an IoT device confirms the firmware it has is not the latest version, it requests for an update to the blockchain nodes.

(7) Retrieval Node $\rightarrow$ IoT Device : ***Response***
A retrieval node sends the manifest file that is stored in the blockchain network to the requesting IoT device.

(8) Retrieval Node $\rightarrow$ IoT Device : ***Response***
Retrieval nodes download the distributed firmware image that is stored in the blockchain network and sends the firmware image to the requested IoT device.

Each transaction should contain its transaction type number as a transaction identifier. Note that transaction can be formed in variable ways by platforms. The compositions that each transaction contains are presented in Table 2.

C. SAFETY MITIGATION

Our proposed architecture is able to mitigate 'targeting issue' and 'author-disappearing issue' which can occured in the existing SUIT architecture. The description of each issues is as follows.

- Targeting Issue: In the client-server model, a server manages all the data it stores, so a firmware file can be forged into a malicious file before distributed. An adversary may also delete the latest firmware file before distributed if the server is stolen. An adversary may also delete the latest firmware file. The traditional client-server structure offers a highly efficient and valued target for adversaries since the attack on a server can damage every IoT device requesting firmware update to the server. In addition, as the number of IoT devices increases exponentially, the firmware update requests may not be properly processed by the server due to the server overload [16]. This centralized condition provides easier environment for an adversary to conduct a network attack such as distributed denial of service attack.
- Author-Disappearing Issue: Authors distributing firmware are not guaranteed to be permanent. Authors may disappear due to cyber-attacks or funding problems. If the author disappears before an IoT device downloads the latest version of firmware released, the IoT device will never be able to update the firmware indefinitely. This issue can lead damage of notorious cyber-attacks more serious. For example, if the latest version of firmware image contains a patch for the recent cyber attack, an attacker could make author disappeared and prevent the firmware download permanently to maintain the IoT devices vulnerable. Also, this issue means that companies with less funding power are more likely to be attacked. This issue must be resolved because it can lead to financial polarization in terms of security and hinder the development of the IoT firmware industry.

Both issues are single points of failure and our proposed architecture mitigate these issues with the characteristics of blockchain such as decentralization. integrity, and irreversibility of data.

The decentralization architecture provided by the blockchain network resolves the targeting issue. Every node shares a same ledger in the blockchain network. Therefore, a targeting attack on a particular node does not act as a single point failure. Also, even if some malicious nodes send unneeded transactions repeatedly, the transactions will be distributed to the blockchain nodes by their roles. The registration nodes and retrieval nodes in our proposed architecture enable efficient distribution, and prevent bottleneck problem. In addition, if any nodes send same transactions or unverifiable transactions repeatedly, a function that determine the node as a malicious node through the consensus of the blockchain nodes can be added.

If a server operated or managed by an IoT device manufacturer or a firmware image vendor disappears due to cyber-attacks or funding problems, all data stored on the server also become unavailable. However, the proposed architecture ensures integrity and irreversibility for the hash values of the firmware image and manifest files stored on the blockchain so that provides a permanent download of firmware images even after the author is disappeared. This resolution can function differently in the consortium blockchain and the private blockchain.

- Consortium Blockchain: The consortium blockchain is operated jointly by several authors. Therefore, even if the author disappearing issue occurs in the consortium blockchain architecture, the nodes that linked to other co-authors and the non-author nodes of disappeared author retain their firmware and manifest files. The remaining nodes therefore provide irreversible download after the author-disappearing issue. Figure 5 shows the resolution of an author-disappearing issue in the consortium blockchain.
- Private Blockchain: The private blockchain is suitable for large enterprises that are responsible for managing numerous types of devices to manage and reserved

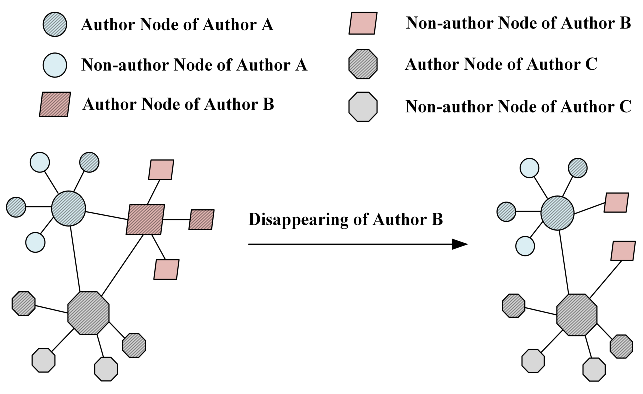

FIGURE 5. Resolution of author-disappearing issue in the consortium blockchain.

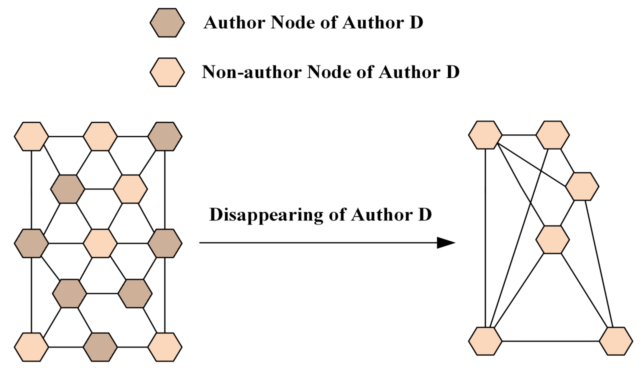

FIGURE 6. Resolution of Author-Disappearing Issue in the Private Blockchain.

enough credibility. In the private blockchain, if the author node are disappeared, the blockchain network only consists of non-author nodes. Therefore, a sufficient number of non-author nodes must be guaranteed. In other words, all different types of firmware images of the author should be able to be downloaded only from the non-author nodes. Figure 6 shows the resolution when an author-disappearing issue occurs in the private blockchain.

IV. DISCUSSION

In this section, we provide comparison results between existing blockchain-based firmware update architectures and our architecture in Table 3.

In our proposed architecture, we do not require to apply smart contracts. However, Scheme 3 applies smart contracts that can incur excessive traffic and cost. Also, the proposed architecture does not require the IoT devices to be operated as part of the blockchain network. On the other hand, Scheme 1 is consisting of the blockchain network with IoT devices even though performing verification and consensus algorithms as blockchain nodes require immoderate computing power for IoT devices. Also, Scheme 1 applies the PoW consensus algorithm which requires too many resources.

As the number and types of IoT devices have increased, the number of firmware files that vendors have to

TABLE 3. Comparison results.

	Scheme 1 [1]	Scheme 2 [11]	Scheme 3 [12]	Our Proposed Architecture
Using Smart Contract	No	No	Yes	No
Configuring IoT Devices as Blockchain Nodes	Yes	No	No	No
Bottleneck Avoidance	No	No	Gateway Node and Broker Storage	Registration Node and Retrieval Node
Multi Vendors Support	No	No	Yes	Yes
Applied Blockchain Platform	PoW based Platform	MultiChain (Private Blockchain)	Ethereum	Private or Consortium Blockchain
Firmware Storage	Vendor Database	Blockchain Network	Vendor Repository and Broker Repository	Distributed in Blockchain Network
Server Targeting Attack Resolving	No	No	No	Yes
Author -Disappearing Issue Resolving	No	Depends on Cases	No	Yes

manage, or register is also getting expanded. If a firmware file is uploaded and all corresponding devices request for downloads at once, firmware uploads from some other vendors would fail. Therefore, the proposed architecture configures the registration node and the retrieval node to prevent this bottleneck issue.

The architecture we propose in this paper suggests both private and consortium depending on the size of the company. In other words, the proposed architecture supports both single-manufacturer architecture, and multi-vendors architecture. But, Scheme 1 suggests an architecture that is only suitable for a single vendor. Since the number of manufacturer or vendors are rapidly growing, firmware update architectures should support multi-vendors.

There are various platforms in the blockchain platform, depending on the network architecture or the type of supporting language. However, PoW-based blockchain platforms take too much time due to very low transaction throughput and 6 confirmation [17]. Besides, blockchain platforms in public environments are more vulnerable than the permissioned blockchain because the blockchain participants have the right to read, write data, and create blocks [18].

The architecture proposed in this paper has a low risk of server targeting attacks because the firmware images are stored distributed on the blockchain network. However, Schemes 1, 2 and 3 store the firmware images in the cloud or storage managed by the manufacturers in centralized architecture. Therefore, Scheme 1, 2 and 3 are vulnerable to server targeting attacks. Furthermore, the centralized management of data is not immune to the author-disappearing issue. Scheme 2 is immune depend on cases because only the first of the two methods proposed in Scheme 2 stores the firmware binaries in the blockchain. In our proposed architecture, the author-disappearing issue is solved through distributed storage in the blockchain network.

V. CONCLUSION

The IoT technology, which realizes a hyper-connected society through the interaction of things and humans, is regarded as a core technology of the 4th Industrial Revolution. However, there is a lot of concern in terms of security because of its low computing power. Since the firmware update for IoT devices must be performed to the IoT devices, the secure firmware update is necessarily required. In this paper, we stated a decentralized firmware update architecture based on blockchain. The expressed architecture can distribute network load and guarantee integrity of firmware images. The blockchain network presented in this paper consists of registration nodes that process registration of manifest and firmware image files from authors, and retrieval nodes that process downloading manifest and firmware image files. The firmware files are stored in the distributed file system and the hash values of the files downloaded are stored on the blockchain. It can also resolve the server targeting and author-disappearing issues. For future work, we would like to design specific message procedures for the transactions. Also, we look forward to implementing the architecture in both private and consortium blockchain and obtain performance evaluation and security analysis.

REFERENCES

[1] 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, Sep. 2016.

[2] B.-C. Choi, S.-H. Lee, J.-C. Na, and J.-H. Lee, "Secure firmware validation and update for consumer devices in home networking," *IEEE Trans. Consum. Electron.*, vol. 62, no. 1, pp. 39–44, Feb. 2016.

[3] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[4] J.-H. Lee, "Rise of anonymous cryptocurrencies: Brief introduction," *IEEE Consum. Electron. Mag.*, vol. 8, no. 5, pp. 20–25, Sep. 2019.

[5] S. Zhang and J.-H. Lee, "Smart contract-based secure model for miner registration and block validation," *IEEE Access*, vol. 7, pp. 132087–132094, 2019.

[6] S. Zhang and J.-H. Lee, "Analysis of the main consensus protocols of blockchain," *ICT Express*, to be published.

[7] C. Merkle, "A Digital signature based on a conventional encryption function," in *Proc. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2000, pp. 369–378.

[8] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congr. Big Data (BigData Congress)*, Honolulu, HI, USA, Jun. 2017, pp. 557–564.

[9] B. Moran, H. Tschofenig, and H. Birkholz, *SUIT CBOR Manifest Serialization*, document draft-moran-suit-manifest-05, 2019. [Online]. Available: https://tools.ietf.org/html/draft-moran-suit-manifest-05

[10] B. Moran, H. Tschofenig, D. Brown, and M. Meriac, *A Firmware Update Architecture for Internet of Things*, document draft-ietf-suit-architecture-08, 2019. [Online]. Available: https://tools.ietf.org/html/draft-ietf-suit-architecture-08

[11] A. Boudguiga, N. Bouzerna, L. Granboulan, A. Olivereau, F. Quesnel, A. Roger, and R. Sirdey, "Towards better availability and accountability for IoT updates by means of a blockchain," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops (EuroS&PW)*, Apr. 2017, pp. 50–58.

[12] A. Yohan and N.-W. Lo, "An over-the-blockchain firmware update framework for IoT devices," in *Proc. IEEE Conf. Dependable Secure Comput. (DSC)*, Kaohsiung, Taiwan, Dec. 2018, pp. 1–8.

[13] S. Wang, Y. Zhang, and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," *IEEE Access*, vol. 6, pp. 38437–38450, 2018.

[14] S. Muralidharan and H. Ko, "An InterPlanetary file system (IPFS) based IoT framework," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Las Vegas, NV, USA, Jan. 2019, pp. 1–2.

[15] J. Benet, "IPFS–content addressed, versioned, P2P file system," 2014, *arXiv:1407.3561*. [Online]. Available: http://arxiv.org/abs/1407.3561

[16] K. Zandberg, K. Schleiser, F. Acosta, H. Tschofenig, and E. Baccelli, "Secure firmware updates for constrained IoT devices using open standards: A reality check," *IEEE Access*, vol. 7, pp. 71907–71920, 2019.

[17] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication," in *Proc. Int. Workshop Open Problems Netw. Secur.* Cham, Switzerland: Springer, 2015.

[18] S. Zhang and J.-H. Lee, "Double-spending with a sybil attack in the bitcoin decentralized network," *IEEE Trans Ind. Informat.*, vol. 15, no. 10, pp. 5715–5722, Oct. 2019.

SEOYUN CHOI is currently pursuing the bachelor's degree in computer engineering with Sangmyung University, Cheonan, South Korea. She is also a Research Assistant with the Protocol Engineering Laboratory, Sangmyung University. Her research interests include protocol engineering, network security, and blockchians.

JONG-HYOUK LEE (Senior Member, IEEE) received the Ph.D. degree in computer engineering from Sungkyunkwan University, Suwon, South Korea. He was with INRIA, France, for IPV6 vehicular communication and security research. He was an Assistant Professor with the TELECOM Bretagne, France. In 2013, he moved to Sangmyung University, Cheonan, South Korea. He has authored the Internet Standards: IETF RFC 8127, IETF RFC 8191, and IETF RFC 8691. His research interests include protocol engineering and performance analysis. He was a recipient of the Best Paper Award from the IEEE WiMob 2012, the 2015 Best Land Transportation Paper Award from the IEEE Vehicular Technology Society, the Haedong Young Scholar Award, in 2017, and the IEEE Systems Journal Best Paper Award from the IEEE Systems Council, in 2018.