electronics

MDPI

Article

Blockchain-Based Secure Firmware Update Using an UAV

Jong Wan Seo [1,†], Anik Islam [2,†], Md Masuduzzaman [1,†] and Soo Young Shin [1,*,†]

[1] Department of IT Convergence Engineering, Kumoh National Institute of Technology (KIT), Gumi 39177, Republic of Korea; sayo94@kumoh.ac.kr (J.W.S.); masud.prince@kumoh.ac.kr (M.M.)
[2] Department of Electrical and Software Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada; anik.islam@ucalgary.ca
* Correspondence: wdragon@kumoh.ac.kr
† These authors contributed equally to this work.

Abstract: This paper proposes a blockchain-based firmware update method using unmanned aerial vehicles (UAVs) to solve one of the security issues arising in the Internet of Things (IoT) environment, which is the firmware security problem. It has high scalability and transaction speed using private blockchains and solves the limitations of internet connections by updating the firmware using an UAV. The proposed firmware update system safely manages the IoT device and firmware information through four processes: participant registration, firmware registration/update, firmware update request, and firmware update. The verification of IoT devices and UAVs is performed using the IoT device's public key and Bloom filter, and firmware updates can be safely performed using public-key encryption communication. To prove the security of the proposed method, a security analysis based on the STRIDE model was conducted, and the performance of the blockchain network was analyzed by simulation on the Hyperledger.

Keywords: blockchain; firmware; unmanned aerial vehicle

Citation: Seo, J.W.; Islam, A.; Masuduzzaman, M.; Shin, S.Y. Blockchain-Based Secure Firmware Update Using an UAV. *Electronics* **2023**, *12*, 2189. https://doi.org/10.3390/electronics12102189

Academic Editor: Mehdi Sookhak

Received: 7 April 2023
Revised: 6 May 2023
Accepted: 9 May 2023
Published: 11 May 2023

1. Introduction

The Internet of Things (IoT) is a technology that connects to the internet by embedding sensors and communication functions in various objects. That is, it refers to a technology that connects various objects through wireless communication [1]. Recently, IoT technology has provided great advantages in various fields, such as agriculture [2–5], medical care [6–9], smart grids [10,11], and smart homes [12–14]. Experts predict that by 2020, about 30 billion IoT devices will be connected to the internet [15]. However, as interest in IoT technology increases, several vulnerabilities and security issues have also been raised. One of them is the firmware security issue of IoT devices. Firmware is a type of operating system responsible for the basic control and operation of devices and is stored in a non-volatile memory device such as a ROM. Cui and Prada-Delgado discussed attacks on the firmware of IoT devices [16,17]. When a firmware attack occurs, it can cause problems such as data manipulation and device operation interruption through infected firmware. If even a single device is subjected to a firmware attack, it is a threat in an IoT environment where numerous devices are connected. Therefore, firmware security in the IoT environment is an important issue that needs to be addressed.

To solve the firmware security problem, various studies have been conducted on firmware updates applying blockchain technology. Blockchain is a data-distributed processing technology that distributes and stores data, such as the transactions of all users participating in the network [18]. The blocks in which these transactions are stored are chained and cannot be modified [19]. Nowadays, blockchain technology is also being applied to various fields, including IoT technology. Blockchain is largely divided into public blockchain, private blockchain, and consortium blockchain. In public blockchain, everyone can participate in the process of confirming transactions and obtaining consensus.

Therefore, there is a high degree of decentralization. Conversely, only authorized users can participate in private blockchains and consortium blockchains. These have excellent transaction speeds and scalability because only the designated person participates in the consensus process. Thus, blockchain represents a tentative solution for securing firmware for IoT devices. However, it is difficult to connect with IoT devices where connectivity is limited.

The use of unmanned aerial vehicles (UAVs) is an emerging paradigm that has drawn attention both in academia and industry. UAVs have diverse functions such as extended coverage, ease of deployment, low-cost maintenance, and so on. Thus, UAVs can be beneficial both in urban and remote rural areas.

In this paper, a blockchain-based firmware update method is proposed using an UAV. Providing security in firmware updates in remote areas via UAVs has not been explored yet, to the best of our knowledge. The main contributions of this paper are as follows.

- A blockchain-based firmware update scheme using UAV is proposed.
- The limitation of the availability of an internet connection by updating the firmware using UAVs is resolved.
- The secure management of firmware and IoT device information in the blockchain network is achieved through four processes.
- Result analysis was obtained by analyzing the blockchain network performance using Hyperledger Fabric. Additionally, the verification of the UAV and IoT device was obtained through asymmetric encryption and the Bloom filter. Moreover, the security analysis was performed using a well-known security model called STRIDE.

The remainder of this paper is organized as follows. Related works are discussed in Section 2. Section 3 describes a firmware update system using the proposed UAV. Section 4 discusses the security of the proposed system based on the STRIDE model, and Section 5 analyzes the performance of the blockchain network through simulation. Finally, Section 6 concludes the paper.

2. Related Works

Among the existing researchers, Lee et al. proposed a blockchain-based firmware verification and update system for embedded devices in IoT environments [20]. They used blockchain technology to verify, distribute, and check firmware versions. In the proposed firmware update system, the blockchain network consists of a verification node, a general node, and a vendor node. Alexander et al. proposed a framework for firmware updates based on blockchain in the IoT environment [21–23]. The proposed framework devised a firmware update method based on PUSH, and the blockchain network consists of a vendor node and a passive node. The vendor node operated by the manufacturer creates, verifies, and distributes the firmware update-related smart contract to the blockchain network. Passive nodes do not participate in verification, and the vendor node receives a firmware update-related smart contract and can execute the firmware update process when conditions such as device name and version are met. Unlike previous studies where IoT devices saved a copy of the blockchain ledger and participated as a blockchain node, a gateway that stores information about the IoT devices participates in the blockchain network through a passive node. However, in the proposed framework, the gateway delivers the received firmware to the IoT device for firmware updates without confirmation. This is dangerous if the vendor sends a smart contract or other malicious code-inserted files. Pillai et al. proposed that when a new firmware update is released, the device vendor generates a hash using the firmware version and the time the firmware was created [24]. The hashes generated this way are connected to form a hash chain, which is used to verify the received firmware.

Baza et al. [25] proposed a blockchain-based firmware update scheme for autonomous vehicles to ensure firmware updates' integrity, security, and traceability. It uses smart contracts and cryptographic techniques for traceability, accountability, and transparency, providing a high level of security and reliability in updating the firmware of an autonomous

vehicle. However, the authors demonstrated no practical experiments regarding the firmware update using the AVs. Seoyun et al. [26] proposed an architecture combining blockchain, smart contracts, and edge computing to ensure secure, efficient firmware updates for IoT devices. The architecture also includes data privacy and confidentiality mechanisms, allowing device owners to maintain control over their data. The authors demonstrate the feasibility and effectiveness of the proposed architecture through a prototype implementation using the Ethereum blockchain platform. However, the Hyperledger platform can be more feasible as it is faster than Ethereum and data can only be controlled among specific users. Hanqing et al. [27] proposed a solution to address the scalability and efficiency issues of existing blockchain-based supply chain traceability systems. The authors demonstrate the feasibility and effectiveness of the proposed solution through a prototype implementation using the Hyperledger Fabric blockchain platform. Later, Tsaur et al. [28] proposed a secure and efficient firmware update mechanism utilizing a blockchain. However, they primarily focused on minimizing the storage needed and enhancing the system's security only. Furthermore, Mingjin et al. [29] introduced some approaches to video surveillance and secure data storage techniques using blockchain and edge computing. Mingjin et al. [29] proposed a unique system that leverages collaborative edge intelligence, where multiple edge devices collectively analyze video data for real-time surveillance. Blockchain technology ensures data integrity, trustworthiness, and transparency in the surveillance process. Jiang et al. [30] also presented a privacy-focused data-sharing mechanism for intelligent transportation systems (ITS) using blockchain technology. The proposed system aims to address the challenges of data privacy, security, and efficiency in ITS by leveraging a combination of blockchain, encryption techniques, and data-sharing protocols. The authors demonstrate the feasibility and effectiveness of their approach through a prototype implementation. However, there is a lack of research on blockchain-based secure firmware updating mechanisms for remote areas using UAVs in the existing literature.

As shown in Table 1, although the above studies solved the problem of the performance of IoT devices due to their participation in the blockchain network through a gateway, it is difficult to update the firmware of IoT devices in areas where an internet connection is not possible or internet infrastructure is not established.

Table 1. Summary of the existing research along with its drawbacks.

Existing Schemes	Details	Issues
Lee et al. [20]	Blockchain-based secure firmware update for embedded devices in an Internet of Things environment	None of the existing research considered internet connection issues (i.e., whether the firmware can be updated according to the internet connection)
Yohan et al. [21]	Blockchain-based firmware update framework for Internet of Things environment	
Yohan et al. [22]	An over-the-blockchain firmware update framework for IoT devices	
Yohan et al. [23]	FOTB: a secure blockchain-based firmware update framework for IoT environment	
Pillai et al. [24]	Securing firmware in Internet of Things using blockchain	
Baza et al. [25]	Blockchain-based firmware update scheme for autonomous vehicles	
Seoyun et al. [26]	Blockchain-based efficient firmware updates for IoT devices	
Hanqing et al. [27]	Blockchain-based supply chain traceability systems	
Tsaur et al. [28]	Efficient firmware update mechanism utilizing blockchain	
Mingjin et al. [29]	Video surveillance and secure data storage technique using blockchain	
Jiang et al. [30]	Privacy-focused data-sharing mechanisms for intelligent transportation	

3. System Model

This paper proposes a blockchain-based firmware update system using an UAV. The firmware information is managed on the blockchain network, and based on that information, the firmware can be safely updated using an UAV. The proposed system is shown in Figure 1 and is composed of the following four entities:

- Blockchain network: A blockchain network that records and shares firmware information. Anything related to the firmware in the blockchain is stored in the block, and the information cannot be modified. This paper proposes a private blockchain for high transaction speed and scalability, and this network is managed by a trusted certification authority.
- Gateway and IoT device: As a participant in the blockchain network, it is a subject that needs a firmware update. To participate in the blockchain network, IoT devices must register information such as device name, installed firmware, and location (latitude, longitude). By participating in the blockchain network, the gateway can obtain information about the firmware and request a firmware update. In the proposed system, even the case where an internet connection is impossible is taken into consideration.
- Vendor: In order to participate in the blockchain network, information such as vendor name, owned device, and owned firmware must be registered. As a participant, it provides firmware information. When a new firmware is developed, information about the target device and firmware is registered in the blockchain network to store and share information securely.
- UAV: It uses the firmware file developed by the manufacturer and the information of IoT devices registered in the blockchain network to securely communicate with IoT devices to update the firmware.

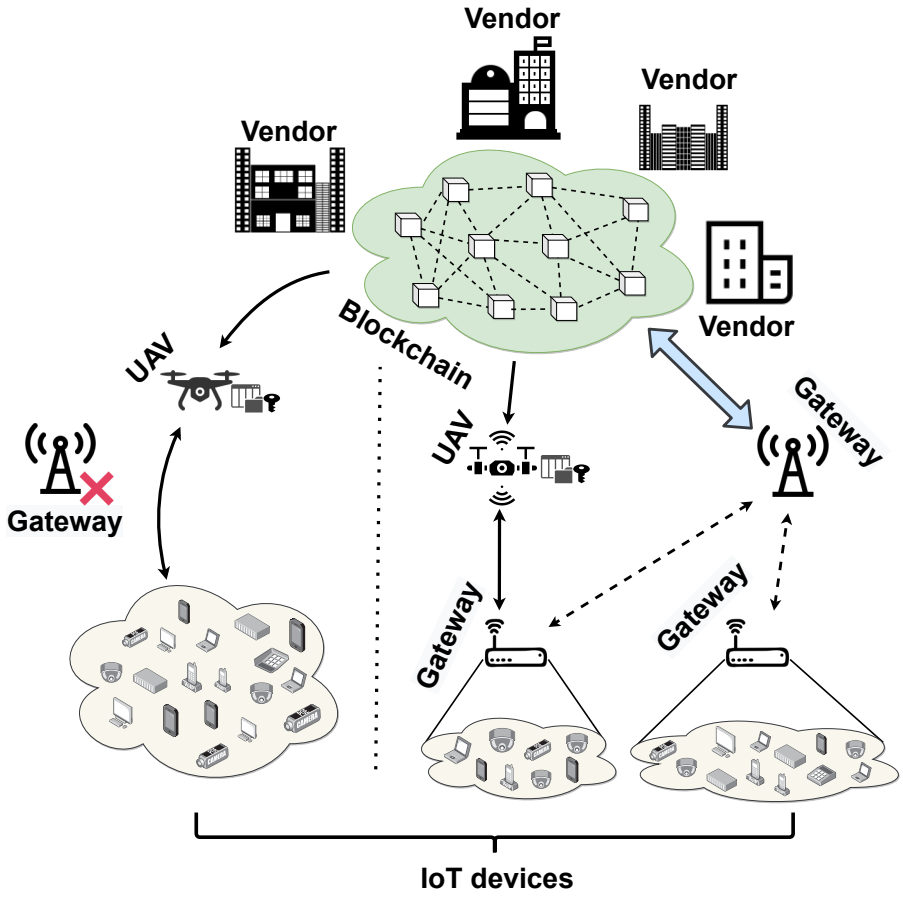


Figure 1. Proposed system model.

The proposed system is designed based on the Hyperledger Fabric architecture. Only participants who have been verified with a private blockchain can participate in the blockchain network. The proposed system includes a participant registration process, a firmware registration and update process, a firmware update request process, and a firmware update process. The data generated by all processes are written as blocks. Participants in the blockchain are verified with the Membership Service Provider (MSP), and the UAV and IoT devices verify each other using a public key and Bloom filter.

Figure 2 shows the participant registration process for IoT devices and manufacturers to participate in the blockchain network. The participant registration process is as follows.

1. To participate in the blockchain network, participants must register information on the blockchain network. The information provided by participants is as follows.
   - IoT device: device name, installed firmware, location (latitude, longitude), and public key.
   - Vendor: vendor name, owned device, owned firmware, and public key.

   In the case of IoT devices, the process is divided into two types, as follows, depending on whether there is an internet connection.
   - IoT device connected to the internet: In this case, IoT devices can participate in the blockchain network through the gateway. Here, the gateway becomes a participant in the blockchain network by collecting information on connected devices, creating a device list, and registering it.
   - IoT device not connected to the internet: In this case, before placing the IoT device, register the device's information and public key in advance on the blockchain network.
2. The blockchain network that received the information checks whether the participant is registered in the participant storage through the received information.
3. The result of whether a participant is registered in the blockchain network is returned, as shown in Algorithm 1.
4. When there is no new participant information in the participant registry, the information received is added to the registry.
5. The participant registration result is delivered to the requester.

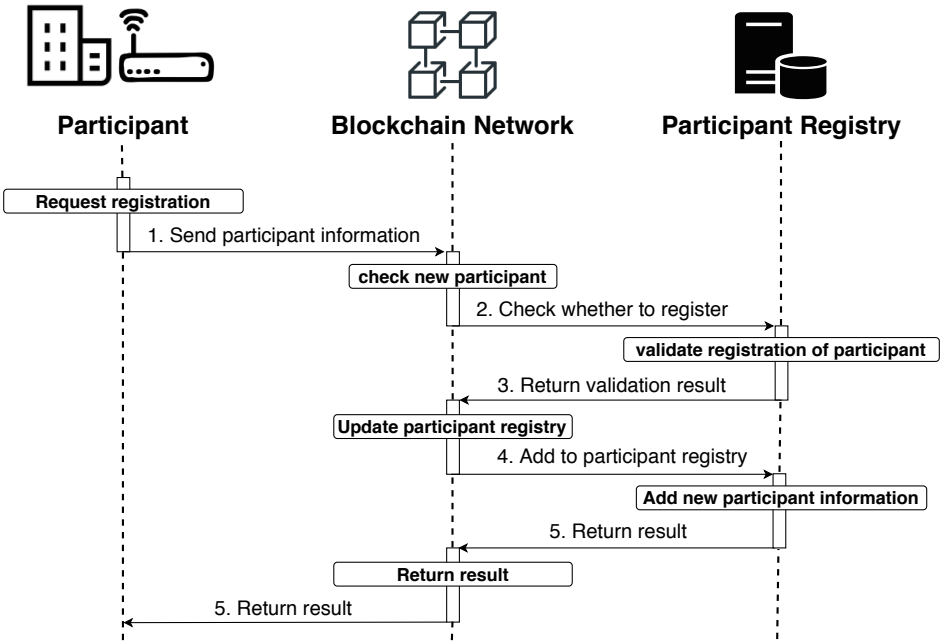


Figure 2. Participant registration process.

Algorithm 1: Participant Registration.

```
Input: I
[Information of participants]
for i = 1 to number of participants do
    if I == Participant Registry[i] then
        terminate;
    end
end
Add new participant information;
number of participants ++;
terminate;
```

Each registered participant is given a different ID, which is stored in the MSP. Now, IoT devices that cannot connect to the internet are stored separately. Participants are given different permissions depending on their roles. For example, as the manufacturer must manage firmware information, all functions (create, read, update, delete) are permitted for the firmware information, but only the read function is permitted for the gateway. Figure 3 shows the process of registering and updating information on the new firmware by a registered vendor. The firmware registration and update processes are as follows.

1. The manufacturer delivers firmware information to register/update the firmware on the blockchain network, which is as follows.
   - Firmware registration: vendor name, target device, and firmware information.
   - Firmware update: vendor name, registered firmware ID, and firmware information.
2. Before registering/updating the firmware, check if the requester has permission from the MSP, as shown in Algorithm 2.
3. Requester's permission check result is returned.
4. When the requestor has permission, register/update the firmware.
   - Firmware registration: information about new firmware is added to the firmware registry.
   - Firmware update: update the information on the new firmware by using the registered firmware ID to retrieve its information from the firmware registry.
5. When registration/update is completed, information about the firmware is notified to the participants of the blockchain network, and the registration/update result is delivered to the requester.

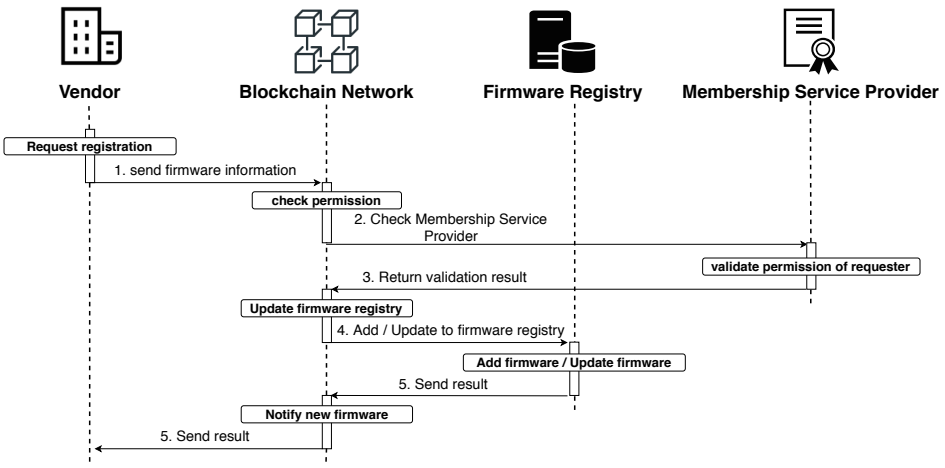


Figure 3. Firmware registration/update process.

Algorithm 2: Firmware Registration/Update.

```
Input: V, F
[Information of Vendor], [Information of Firmware]
if Check that V has the authority then
    for i = 1 to number of firmware do
        if F == Firmware Registry[i] then
            Firmware update;
            terminate;
        end
    end
    Add new firmware;
    number of firmware ++;
    terminate;
else
    terminate;
end
```

When information on new firmware is registered/updated, a firmware registration/update event occurs and is transmitted to the participants in the network. Thus, the gateway can quickly obtain information about the new firmware.

Figure 4 shows the process of requesting a firmware update by a gateway when information about the new firmware is updated on the blockchain network. The firmware update request process is as follows.

1. The gateway receives information about the firmware when registering/updating the new firmware.
2. The gateway determines whether to update the firmware by comparing the received firmware information with its list of devices, as shown in Algorithm 3. When a firmware update is required, it requests the same from the blockchain network.
3. To verify the gateway that was requested for the firmware update, the blockchain network retrieves the device list of the gateway using the gateway's ID in the participant registry. It checks whether there is an IoT device requiring a firmware update by comparing information such as the target device and the firmware version of the registered/updated firmware in the device list.
4. The verification result is returned.
5. If the firmware update is required, information on the device is sent to the vendor.

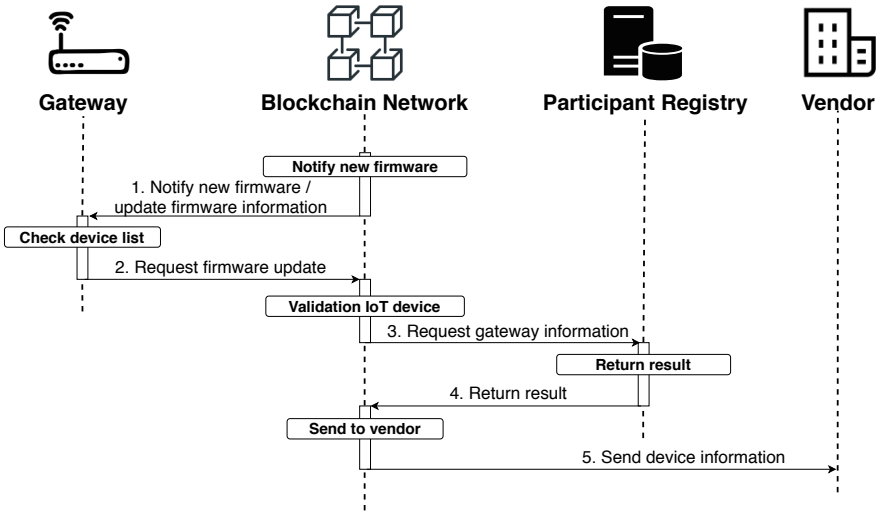


Figure 4. Firmware update request process.

Algorithm 3: Firmware Update Request.

```
Input: G, F
[Information of Gateway]
[Information of Firmware]
if Check G and F for firmware update then
    for i = 1 to number of participants do
        if G == Participant Registry[i] then
            send information;
            terminate;
        end
    end
    terminate;
else
    terminate;
end
```

In the case of IoT devices that cannot be connected to the internet, when information on new firmware related to the IoT device is updated on the blockchain network, the information on the IoT device is always delivered to the vendor to update the firmware. Figure 5 shows the process of updating the firmware of IoT devices that have requested an update, performed by a manufacturer using an UAV. The firmware update process is as follows.

1. To update the firmware, a list containing information about the firmware and IoT devices is stored in the UAV. The Bloom filter is used to block unnecessary IoT device access and verify those devices that have requested a firmware update, as shown in Algorithm 4. The Bloom filter is created using the information from IoT devices to be updated by the UAV.
2. The UAV checks the location information of the devices, moves to the location, and requests communication from the IoT device. Now, the UAV is verified using the public key registered by the IoT device to participate in the blockchain network.
3. After the UAV is verified, the UAV uses its own Bloom filter to verify the IoT device.
4. After the verification of the UAV and the IoT device is completed, the IoT device uses a private key and the UAV uses a public key to securely communicate with a public key encryption method to update the firmware.
5. When the firmware update of all IoT devices in the firmware update list is completed, this result is recorded by the manufacturer on the blockchain network. Now, information about the UAV is also recorded. The blockchain network uses the received firmware update result to update the information of IoT devices that have completed the firmware update.

If the verification between the UAV and IoT device fails, the firmware update will not proceed, and the result will also be recorded.

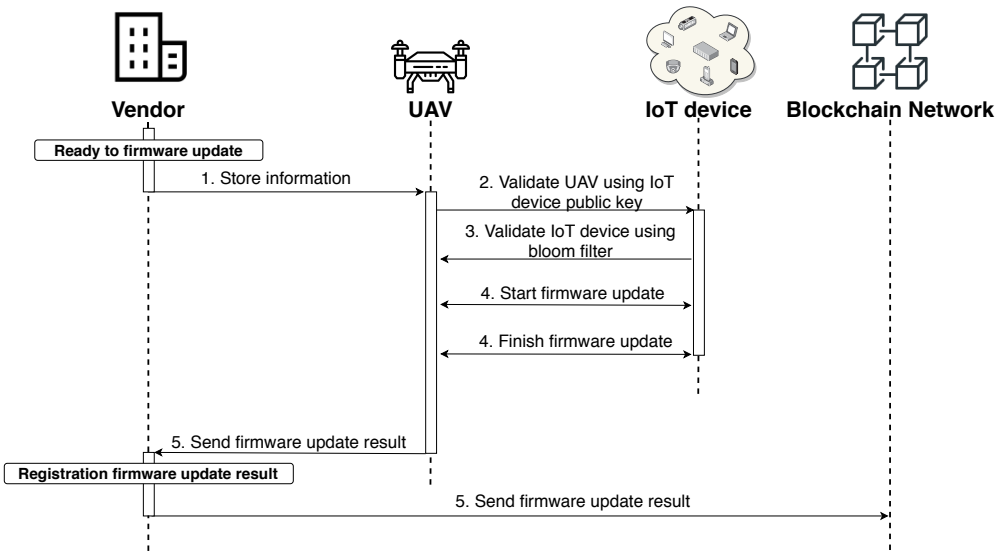


Figure 5. Firmware update process.

Algorithm 4: Firmware Update.

Input: P, B
[Public key of the target device]
[Bloom filter]
if *Use P to verify the identity of the drone* **then**
  if *Use B to verify the IoT device* **then**
    Start firmware update;
  else
    terminate;
  end
else
  terminate;
end

4. Security Analysis

In this paper, security analysis was performed based on the STRIDE (spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege) model. The possible attack and protection structures are discussed below.

- Spoofing—Spoofing means false identity. There is a possibility of transferring the secret message to wrong devices that can appear in the network under a false identity. Therefore, in this paper, public key cryptography is used to prevent a false identity. Here, a digital signature is produced to verify the authentication. Therefore, spoofing issues have been resolved using the digital signature after receiving the data.
- Tampering—Tampering means modifying the information. Attackers, such as hackers, can manipulate data inside the network. Blockchain technology is used here to store the information securely and to prevent illegal data from being tampered with inside the network.
- Repudiation—Repudiation means the denial of any activity by any user inside a network. As the blockchain is used in the proposed system, all of the data are stored in a blockchain ledger. Once the data is stored with the verification of other users of the network, a third party or attacker will not be able to modify it by entering the blockchain network.
- Information disclosure—Information disclosure means the leakage of information to unauthorized outsiders. In this proposed system, public key cryptography is

used to transfer the data, and the digital signature is used to verify the authentication of the user. Therefore, the data leakage problem is solved by using the digital signature technique.

- Denial of service (DoS)—Denial of service means denying access to system resources to the legal users of the network. DoS attacks are made by outsiders to hamper system performance. In this proposed system, if a user tries to transfer the data multiple times, it is proposed to be blocked so that it can no longer transfer the data on the network with the intention of any DoS attack.
- Elevation of privilege—Elevation of privilege means access to system resources without permission. As the registration process is completed for every device and the authentication process is maintained using a digital signature, unauthorized users will not be able to access the system resources or the data inside the network.

5. Performance Evaluation

This section evaluates the performance of the proposed blockchain network. The implementation environment is shown in Table 2. To evaluate our proposed scheme, a scenario was considered in which an UAV tried to upgrade the firmware of multiple IoT devices, and the firmware was securely mined under the observation of 10 mining nodes. An onboard Wi-Fi with the drone was considered in the experiments. A lightweight pocket-size Beryl Router (model: GL-MT1300) was attached with the customized tarot x6 UAV to connect and transfer the firmware to the IoT devices, as shown in Figure 6. The NVIDIA Jetson Xavier NX was considered as the onboard computing module, and Holybro SiK Telemetry Radio V3 was considered for controlling the drone. The experiments were conducted for 300 s.

Figure 6. UAV for performing the firmware update.

Table 2. Implementation environment.

OS	**Ubuntu 16.04 LTS**
Hyperledger Fabric	v1.4.1
Consensus	Raft
Number of node	2, 3, . . ., 10

Hyperledger is an open-source blockchain platform that is designed for use by private organizations that know and trust each other. Hyperledger is primarily concerned with consortium networks connecting several parties to speed up crucial, frequently confidential operations and transactions. The primary benefit of Hyperledger is that only authorized parties are permitted to participate in and access the data of the blockchain network [31]. Moreover, Hyperledger acts as a junction for various distributed ledger

frameworks to improve efficiency, performance, and transactions. It can support permitted, permissionless, or hybrid networks, balancing the overall performance and privacy of a blockchain network. Therefore, an organization's sensitive and internal data can be protected and restricted to only some permitted individuals by deploying Hyperledger. Hyperledger supports a variety of consensus algorithms, which allows for better scalability of the network, and offers a range of privacy features that enable participants to keep their transactions private while still allowing for the auditability and transparency of the blockchain network [32]. It also provides several tools and frameworks to build custom blockchain applications quickly and efficiently. As only verified participants can engage in updating the firmware in this proposed system, the Hyperledger platform is utilized to implement the idea. Hyperledger Fabric is one of the most well-liked enterprise-grade blockchain frameworks and is highly customizable and scalable, able to handle thousands of transactions per second [33]. Moreover, Fabric-based blockchain offers superior security, scalability, confidentiality, and performance. Designers may expand upon what they built using Hyperledger Fabric, making the systems scalable. With the capacity to manage large transaction volumes, Hyperledger Fabric offers a variety of privacy features, including private channels and secret transactions, which are required for our proposed UAV-based secure firmware update system. In addition, Hyperledger Fabric enables our system to protect sensitive information by providing a robust governance model, allowing the proposed system to establish its own rules and policies for the network. Therefore, the blockchain network was implemented using Hyperledger Fabric. To deploy blockchain on different machines, the Amazon web server (AWS) was considered. In addition to npm, node.js, git, python, and docker-compose are installed as prerequisites for using Hyperledger Fabric. All nodes are considered to have similar configurations, and details are provided in Table 3. RAFT was considered as a consensus algorithm. In RAFT, nodes in the network elect a leader responsible for managing the blockchain ledger's replication across all network nodes. The leader notifies all other nodes for confirmation when a new block is added to the blockchain, and it is added to the blockchain once the block has received confirmation from the majority of nodes [34]. One of the primary advantages of using the RAFT algorithm is its fault-tolerant capabilities, which are capable of operating even if some network nodes fail [35]. Moreover, the RAFT assures that only one leader node at a time may modify the blockchain, lowering the possibility of errors and guaranteeing long-term, trouble-free operation in the blockchain network. The middleware and each of the components were built using Python. The following performances were measured and analyzed.

- Transaction rate.
- Transaction throughput.
- Transaction latency.
- Resource consumption.

Table 3. AWS node environment.

Processor	**Intel Xeon Platinum 8000 Series**
vCPUs and Memory	4 and 16 GiB
Clock Speed	Up to 3.1 GHz
Network Performance	Up to 10 Gbps
Operating System	Ubuntu 18.04
Storage Type	Amazon EBS (Elastic Block Store)
Storage Size	100 GB
Network	Virtual Private Cloud
Public IP Address	Disabled

The performance analysis measured the performance of the blockchain network when there were 2, 3, . . ., and 10 nodes. Figure 7a shows transaction throughput over time. Transaction throughput is the percentage of valid transactions executed during a defined period, and this is observed to decrease over time. Network conditions change over time, and the responsiveness of nodes is also associated with a decrease in throughput. Throughput is related to packet loss, latency, and zippers. As the number of nodes increases, it takes longer to verify and share transactions, so the response time of nodes increases. It can be seen that, as the response time increases, a time delay occurs and the throughput decreases.

Figure 7c shows the number of successful transactions over time, which naturally increases. It can be seen that, as the number of nodes increases, the number of successful transactions decreases. As mentioned earlier, it can be seen that this decreases because a time delay occurs due to the verification and sharing of nodes.

Figure 7c shows the delay time according to the number of verification nodes in the blockchain network. As the number of nodes increases, the number of nodes that need to be verified and shared also increases, so the time required for verification and sharing also increases. This causes a delay in the network, and the time delay increases due to an increase in the network delay.

Figure 7d shows the CPU usage over time when executing creation transactions (left) and query transactions (right). You can see that the CPU usage when executing the creation transaction is higher than when executing the query transaction. This is because the creation transaction has a process of verification and sharing, but the query transaction only needs to read the data in the copy of the ledger of one node. You can see that the CPU usage decreases as the number of nodes increases because the nodes divide and execute the required transaction.

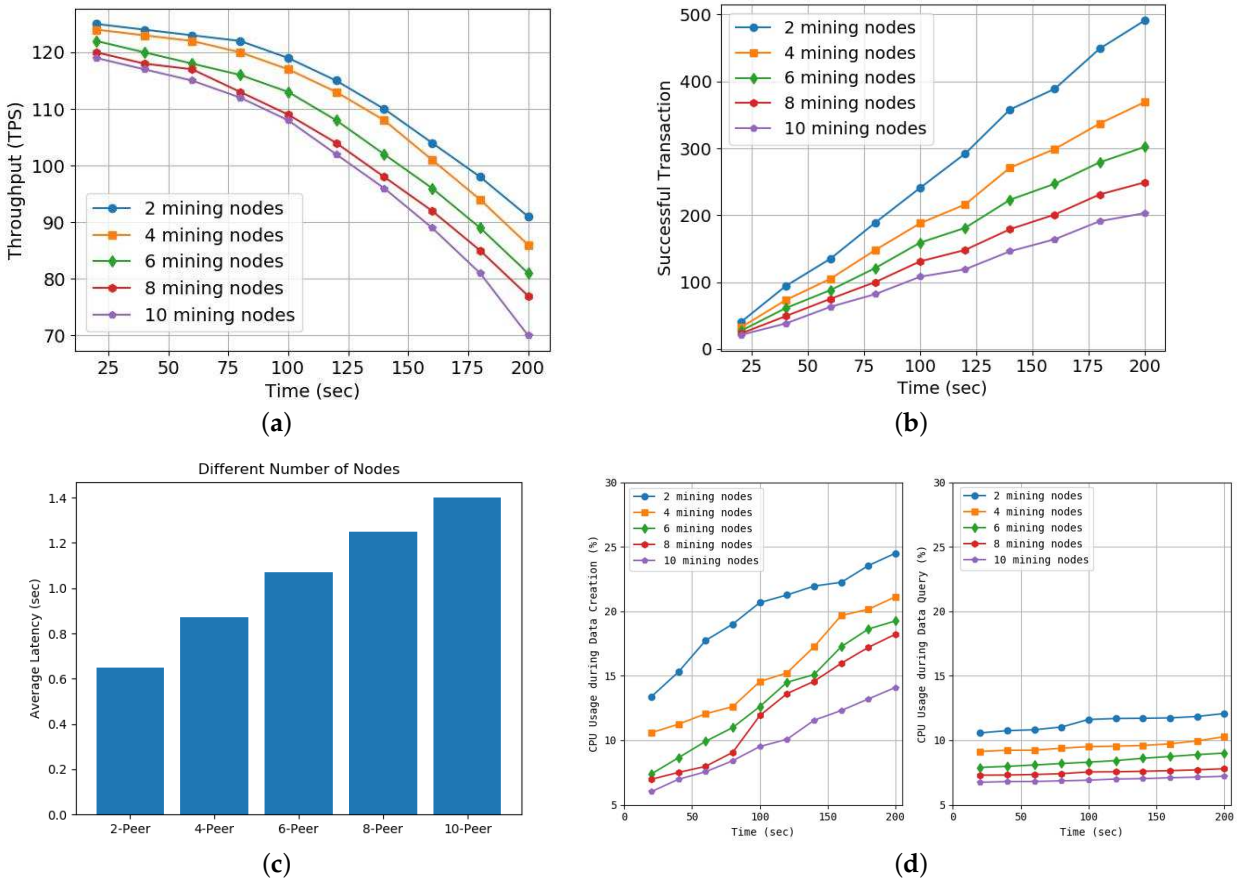
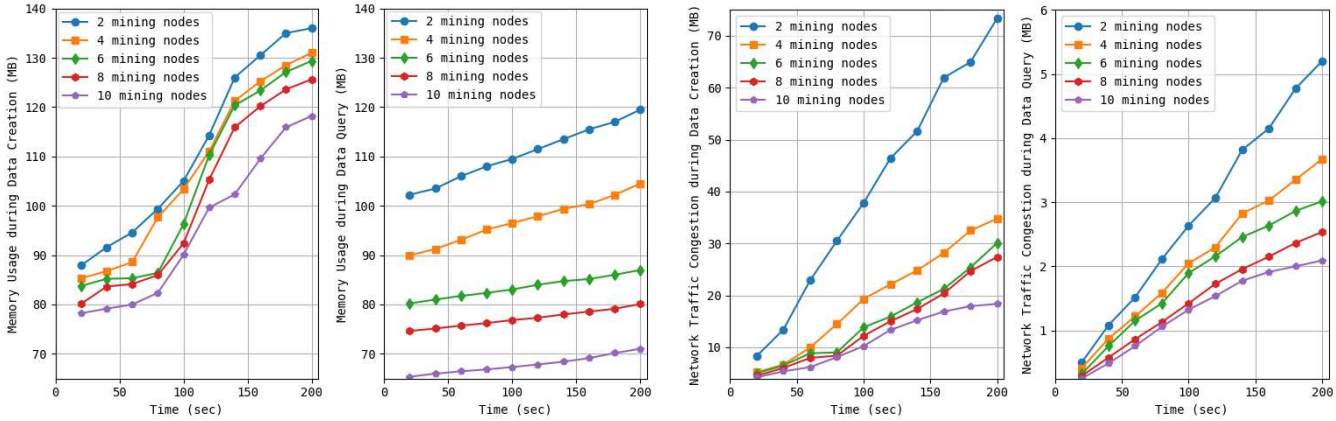


Figure 7. *Cont.*



(**e**) (**f**)

Figure 7. Experimental results performed in blockchain: (**a**) Transaction throughput over time, (**b**) Number of successful transactions over time, (**c**) Latency according to the number of nodes, (**d**) CPU usage over time for data creation (**left**) and data query (**right**), (**e**) Memory usage over time for data creation (**left**) and data query (**right**), and (**f**) Network traffic over time for data creation (**left**) and data query (**right**).

Figure 7e shows the memory usage over time when executing the creation transaction (left) and a query transaction (right). It can be seen that memory usage decreases as the number of nodes increases. If there are two or four nodes up to 50 s, the memory usage of the query transaction is seen to be higher than that of the creation transaction. It can be seen that, out of a data list of 200, query transactions reading specific data used more memory. However, in due course, it can be seen that the memory usage of the creation transaction, which requires verification and sharing, increases further.

Figure 7f shows network traffic over time when executing a creation transaction (left) and query transaction (right). Network traffic refers to the amount of data flowing through the network within a certain time. It can be observed that more data flows through the network when the network traffic of the creation transaction is higher than that of the query transaction. Network delays can occur if there is a significant amount of data flow. It can be seen that the network traffic decreases as the number of nodes increases because multiple nodes divide and process the data.

Figure 8 shows the time required to process data security tasks in the UAV. We considered encryption (symmetric), decryption (symmetric), signing, verification, and private/public (pri/pub) encryption and decryption (asymmetric). As the size of data increase, the processing time of each operation increases.

A comparative study of the proposed system and previous studies was performed, as shown in Table 4, where (O) means that the existing problem has been solved and (X) means that the existing problem has not been solved.

- Low resources: This problem is the possibility of implementation in an IoT environment with low resources. Lee et al. [20] and Mingjin et al.'s [29] research on storing a blockchain ledger in local storage is difficult to implement.
- Firmware verification: This is a verification problem for the deployed firmware. Lee et al. [20], Yohan et al. [21–23], Hanqing et al. [27], Tsaur et al. [28], Mingjin et al. [29], and Jiang et al. [30] used blockchain in terms of data security but did not consider verification when deploying firmware to IoT devices.
- Internet connection: This problem is whether the firmware can be updated according to the internet connection. Existing studies require an internet connection, but the proposed system can update the firmware using UAVs at any time as long as devices that cannot connect to the internet are registered in advance.

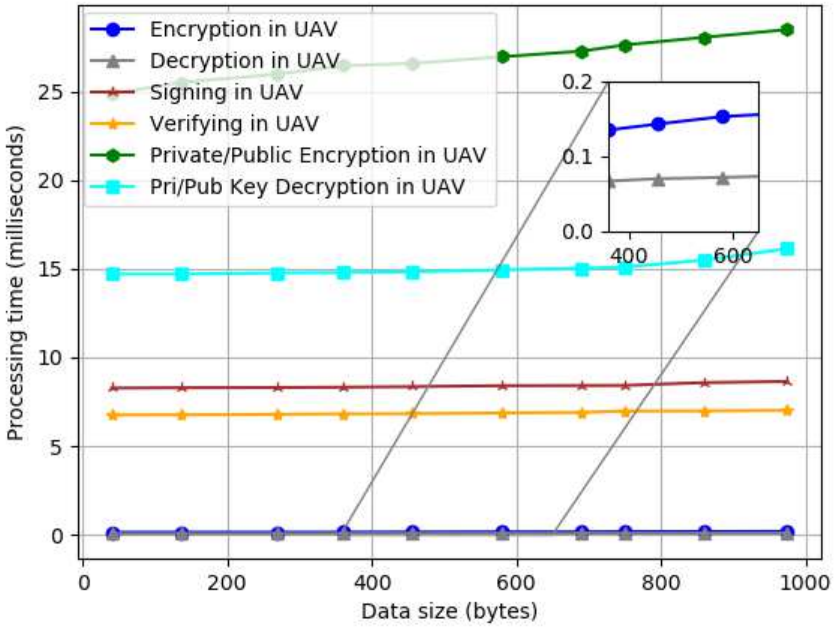


Figure 8. Individual security processing time by data size.

Table 4. Performance comparison.

Features	Schemes		
	Low Resource	**Firmware Verification**	**Internet Connection**
Lee et al. [20]	X	X	X
Yohan et al. [21–23]	O	X	X
Pillai et al. [24]	O	O	X
Baza et al. [25]	O	O	X
Seoyun et al. [26]	O	O	X
Hanqing et al. [27]	O	X	X
Tsaur et al. [28]	O	X	X
Mingjin et al. [29]	X	X	X
Jiang et al. [30]	O	X	X
Proposed System	O	O	O

6. Conclusions

This paper proposes a firmware update method to solve the firmware security problem, which is often raised and with increasing interest in IoT technology. To overcome the low-performance issue of IoT devices and the limitations of internet connections in existing blockchain-based firmware update-related studies for data integrity, we propose a blockchain-based firmware update method using an UAV. Through the participant registration process, the firmware registration and update process, and the firmware update request process, the information on IoT devices and firmware is securely registered and shared. The firmware update process uses a public key and the Bloom filter to verify UAV and IoT devices and provides secure firmware updates using public key encryption. The proposed system guarantees data integrity through blockchain technology, and all processes are recorded in blocks. Participants are given IDs so that when an issue occurs, the cause can be quickly discovered. Because the firmware is updated using an UAV based on the registered information, the load on the network can be reduced, and the IoT device where the internet infrastructure is not established can also update the firmware. In addition, this

paper has completed a security analysis based on the STRIDE model and a performance analysis of the blockchain network using Hyperledger. However, consensus algorithms can consume resources and energy. Thus, choosing a suitable consensus algorithm is necessary to bring efficiency to the system, which has been reserved for future investigation. Sixth-generation wireless communication is planned to overcome connectivity issues with better coverage that can increase the efficiency of the proposed scheme, which will be the subject of future work.

Author Contributions: Conceptualization, J.W.S., A.I. and S.Y.S.; Methodology, J.W.S. and S.Y.S.; Software, J.W.S. and M.M.; Validation, S.Y.S.; Formal analysis, J.W.S., M.M. and S.Y.S.; Investigation, A.I.; Writing—original draft, J.W.S. and A.I.; Writing—review & editing, A.I. and S.Y.S.; Visualization, M.M.; Supervision, S.Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ICAN(ICT Challenge and Advanced Network of HRD) program(IITP-2022-RS-2022-00156394) supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Asghari, P.; Rahmani, A.M.; Javadi, H.H.S. Internet of Things applications: A systematic review. *Comput. Netw.* **2019**, *148*, 241–261. [CrossRef]
2. Dagar, R.; Som, S.; Khatri, S.K. Smart farming–IoT in agriculture. In Proceedings of the 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 11–12 July 2018; pp. 1052–1056.
3. Balakrishna, S.J.; Marellapudi, H.; Manga, N.A. IoT based status tracking and controlling of motor in agricultural farms. In Proceedings of the 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Gorakhpur, India, 2–4 November 2018; pp. 1–5.
4. Zyrianoff, I.; Heideker, A.; Silva, D.; Kamienski, C. Scalability of an Internet of Things platform for smart water management for agriculture. In Proceedings of the 2018 23rd Conference of Open Innovations Association (FRUCT), Bologna, Italy, 13–16 November 2018; pp. 432–439.
5. Ramachandran, V.; Ramalakshmi, R.; Srinivasan, S. An automated irrigation system for smart agriculture using the Internet of Things. In Proceedings of the 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 18–21 November 2018; pp. 210–215.
6. Alam, M.M.; Malik, H.; Khan, M.I.; Pardy, T.; Kuusik, A.; Le Moullec, Y. A survey on the roles of communication technologies in IoT-based personalized healthcare applications. *IEEE Access* **2018**, *6*, 36611–36631. [CrossRef]
7. He, D.; Ye, R.; Chan, S.; Guizani, M.; Xu, Y. Privacy in the Internet of Things for smart healthcare. *IEEE Commun. Mag.* **2018**, *56*, 38–44. [CrossRef]
8. Islam, A.; Young Shin, S. A blockchain-based secure healthcare scheme with the assistance of unmanned aerial vehicle in Internet of Things. *Comput. Electr. Eng.* **2020**, *84*, 106627. [CrossRef]
9. Luo, E.; Bhuiyan, M.Z.A.; Wang, G.; Rahman, M.A.; Wu, J.; Atiquzzaman, M. Privacyprotector: Privacy-protected patient data collection in IoT-based healthcare systems. *IEEE Commun. Mag.* **2018**, *56*, 163–168. [CrossRef]
10. Lee, C.K.; Liu, H.; Fuhs, D.; Kores, A.; Waffenschmidt, E. Smart lighting systems as a demand response solution for future smart grids. *IEEE J. Emerg. Sel. Top. Power Electron.* **2020**, *8*, 2362–2370. [CrossRef]
11. Li, Y.; Cheng, X.; Cao, Y.; Wang, D.; Yang, L. Smart choice for the smart grid: Narrowband Internet of Things (NB-IoT). *IEEE Internet Things J.* **2017**, *5*, 1505–1515. [CrossRef]
12. Meng, Y.; Zhang, W.; Zhu, H.; Shen, X.S. Securing consumer IoT in the smart home: Architecture, challenges, and countermeasures. *IEEE Wirel. Commun.* **2018**, *25*, 53–59. [CrossRef]
13. Elsts, A.; Fafoutis, X.; Woznowski, P.; Tonkin, E.; Oikonomou, G.; Piechocki, R.; Craddock, I. Enabling healthcare in smart homes: The SPHERE IoT network infrastructure. *IEEE Commun. Mag.* **2018**, *56*, 164–170. [CrossRef]
14. Wang, P.; Ye, F.; Chen, X. A smart home gateway platform for data collection and awareness. *IEEE Commun. Mag.* **2018**, *56*, 87–93. [CrossRef]
15. Holler, J.; Tsiatsis, V.; Mulligan, C.; Karnouskos, S.; Avesand, S.; Boyle, D. *Internet Things*; Academic Press: Cambridge, MA, USA, 2014.
16. Cui, A.; Costello, M.; Stolfo, S. When firmware modifications attack: A case study of embedded exploitation. In Proceedings of the 20th Annual Network & Distributed System Security Symposium, San Diego, CA, USA, 24–27 February 2013.
17. Prada-Delgado, M.A.; Vázquez-Reyes, A.; Baturone, I. Trustworthy firmware update for Internet-of-Thing Devices using physical unclonable functions. In Proceedings of the 2017 Global Internet of Things Summit (GIoTS), Geneva, Switzerland, 6–9 June 2017; pp. 1–5.

18. Islam, A.; Shin, S.Y. BUS: A Blockchain-Enabled Data Acquisition Scheme With the Assistance of UAV Swarm in Internet of Things. *IEEE Access* **2019**, *7*, 103231–103249. [CrossRef]
19. Islam, A.; Shin, S.Y. BUAV: A blockchain based secure UAV-assisted data acquisition scheme in Internet of Things. *J. Commun. Netw.* **2019**, *21*, 491–502. [CrossRef]
20. Lee, B.; Lee, J.H. Blockchain-based secure firmware update for embedded devices in an Internet of Things environment. *J. Supercomput.* **2017**, *73*, 1152–1167. [CrossRef]
21. Yohan, A.; Lo, N.W.; Achawapong, S. Blockchain-based firmware update framework for internet-of-things environment. In Proceedings of the Conference Information and Knowledge Engineering, Mashhad, Iran, 25–26 October 2018; pp. 151–155.
22. Yohan, A.; Lo, N.W. An over-the-blockchain firmware update framework for IoT devices. In Proceedings of the 2018 IEEE Conference on Dependable and Secure Computing (DSC), Kaohsiung, Taiwan, 10–13 December 2018; pp. 1–8.
23. Yohan, A.; Lo, N.W. FOTB: A secure blockchain-based firmware update framework for IoT environment. *Int. J. Inf. Secur.* **2020**, *19*, 257–278. [CrossRef]
24. Pillai, A.; Sindhu, M.; Lakshmy, K. Securing firmware in Internet of Things using blockchain. In Proceedings of the 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 15–16 March 2019; pp. 329–334.
25. Baza, M.; Nabil, M.; Lasla, N.; Fidan, K.; Mahmoud, M.; Abdallah, M. Blockchain-based Firmware Update Scheme Tailored for Autonomous Vehicles. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 15–18 April 2019; pp. 1–7. [CrossRef]
26. Choi, S.; Lee, J.H. Blockchain-Based Distributed Firmware Update Architecture for IoT Devices. *IEEE Access* **2020**, *8*, 37518–37525. [CrossRef]
27. Wu, H.; Jiang, S.; Cao, J. High-Efficiency Blockchain-Based Supply Chain Traceability. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 3748–3758. [CrossRef]
28. Tsaur, W.J.; Chang, J.C.; Chen, C.L. A Highly Secure IoT Firmware Update Mechanism Using Blockchain. *Sensors* **2022**, *22*, 530. [CrossRef]
29. Zhang, M.; Cao, J.; Sahni, Y.; Chen, Q.; Jiang, S.; Yang, L. Blockchain-based Collaborative Edge Intelligence for Trustworthy and Real-Time Video Surveillance. *IEEE Trans. Ind. Inform.* **2023**, *19*, 1623–1633. [CrossRef]
30. Jiang, S.; Cao, J.; Wu, H.; Chen, K.; Liu, X. Privacy-preserving and efficient data sharing for blockchain-based intelligent transportation systems. *Inf. Sci.* **2023**, *635*, 72–85. [CrossRef]
31. Masuduzzaman, M.; Islam, A.; Sadia, K.; Shin, S.Y. UAV-based MEC-assisted automated traffic management scheme using blockchain. *Future Gener. Comput. Syst.* **2022**, *134*, 256–270. [CrossRef]
32. Masuduzzaman, M.; Rahim, T.; Islam, A.; Shin, S.Y. UxV-Based Deep-Learning-Integrated Automated and Secure Garbage Management Scheme Using Blockchain. *IEEE Internet Things J.* **2023**, *10*, 6779–6793. [CrossRef]
33. Chen, C.L.; Yang, J.; Tsaur, W.J.; Weng, W.; Wu, C.M.; Wei, X. Enterprise Data Sharing with Privacy-Preserved Based on Hyperledger Fabric Blockchain in IIOTś Application. *Sensors* **2022**, *22*, 1146. [CrossRef] [PubMed]
34. Alexandridis, A.; Al-Sumaidaee, G.; Alkhudary, R.; Zilic, Z. Making Case for Using RAFT in Healthcare Through Hyperledger Fabric. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–18 December 2021; pp. 2185–2191. [CrossRef]
35. Piao, X.; Li, M.; Meng, F.; Song, H. Latency Analysis for Raft Consensus on Hyperledger Fabric. In Proceedings of the Blockchain and Trustworthy Systems, Chengdu, China, 4–5 August 2022; Svetinovic, D., Zhang, Y., Luo, X., Huang, X., Chen, X., Eds.; Springer: Singapore, 2022; pp. 165–176.