electronics

MDPI

Article

Scalable and Secure Internet of Things Connectivity

Ye-Jin Choi [1], Hee-Jung Kang [2] and Il-Gu Lee [3,*]

[1] Department of Future Convergence Technology Engineering, Sungshin University, Seoul 02844, Korea
[2] Department of Computer Science, Sungshin University, Seoul 02844, Korea
[3] Department of Convergence Security Engineering, Sungshin University, Seoul 02844, Korea
* Correspondence: iglee@sungshin.ac.kr; Tel.: +82-2-920-7145

Received: 5 June 2019; Accepted: 1 July 2019; Published: 3 July 2019

Abstract: The Internet of things (IoT) technology, which is currently considered the new growth engine of the fourth industrial revolution, affects our daily life and has been applied to various industrial fields. Studies on overcoming the limitations of scalability and stability in a centralized IoT operating environment by employing distributed blockchain technology have been actively conducted. However, the nature of IoT that ensures connectivity with multiple objects at any time and any place increases security threats. Further, it extends the influence of the cyber world into the physical domain, resulting in serious damage to human life and property. Therefore, we aim to study a method to increase the security of IoT devices and effectively extend them simultaneously. To this end, we analyze the authentication methods and limitations of traditional IoT devices and examine cases for improving IoT environments by using blockchain technology. Accordingly, we propose a framework that allows IoT devices to be securely connected and extended to other devices by automatically evaluating security using blockchain technology and the whitelist. The method proposed in this paper restricts the extension of devices vulnerable to security risks by imposing penalties and allows only devices with high security to be securely and quickly authenticated and extended without user intervention. In this study, we applied the proposed method to IoT network simulation environments and observed that the number of devices vulnerable to security was reduced by 48.5% compared with traditional IoT environments.

Keywords: blockchain; IoT; scalability; authentication; whitelist; secure

1. Introduction

The Internet of things (IoT) era—in which various objects are connected through information and communication technology to collect, share, and process information—has emerged. IoT provides a more convenient environment by enabling various devices to communicate and share information with each other without human intervention [1]. IoT technology is developing rapidly every year and many companies and countries are making huge efforts toward vitalizing IoT. The International Data Corporation estimates that the global IoT market will achieve 15% year-on-year growth reaching $745 billion in 2019 and exceeding $1 trillion by 2022 [2,3]. However, hacking and cyber attacks targeting IoT devices are increasing every year and, as most of them are lightweight, low-power, and low-performance devices, it is difficult to apply security methods adopted for traditional PCs to IoT devices, thus making them vulnerable to cyber attacks [4,5]. As IoT devices collect and process various types of information including personal and sensitive information in everyday life, if IoT devices with vulnerabilities are exposed to such security threats, it can infringe on personal information, cause financial loss, and even threaten human life. If IoT security issues are not properly addressed, a hyper-connected society in which 5G technology is commercially available and everything is connected along with smart cities, smart factories, and smart cars becomes remote. As a solution

to these issues, the convergence of blockchain technology and IoT has drawn attention [6]. In this study, we investigate the blockchain using technology that can securely and automatically extend IoT devices by considering security when authenticating them. In addition, we design a mechanism that evaluates the security of devices by using the whitelist, which first defines the list of safety-proven software and then restricts things beyond the list [7,8]. Accordingly, the proposed model enables IoT devices to verify security automatically by using blockchain and smart contract technology; thus, they can be securely and automatically extended. Further, we ensure that secure extension is available by proposing a method that imposes a penalty on connection extension for low-security devices based on the security of the devices recorded in the blockchain.

The remainder of the paper is organized as follows. Section 2 describes the background and related works on the IoT connectivity tehchnologies and authentication methods. Section 3 analyzes limitations of IoT authentication for secure scalability. Section 4 present the proposed scheme for scalable and secure IoT connectivity. Section 5 describes the performance evaluation results. Finally, Section 6 draws conclusions.

2. Background and Related Works

2.1. IoT Connectivity Technologies

The idea of blockchain technology was described by Stuart Haber and W. Scott Stornetta in 1990, and was introduced in a paper on Bitcoin, a decentralized cryptocurrency, developed by Satoshi Nakamoto in 2008 [9,10]. A blockchain is distributed ledger technology that enables all users participating in the network to share transaction records jointly and e ㅅsures reliability without any accredited third party [11–13]. As everyone can openly obtain access to data in a blockchain and all the network participants own the ledger, it cannot be modified or deleted once information is recorded, thus guaranteeing a higher level of integrity than traditional centralized systems [14]. In addition, as transaction details and data are managed and stored in a distributed manner, problems in some parts of networks do not affect the entire system [15]. Blockchains are classified into public and private blockchains depending on the participation method of participants. A public blockchain is open to the public so that anyone can freely participate in it, and Bitcoin, Etherium, and decentralized operating system (EOS) blockchains are its typical examples [16]. A private blockchain allows only pre-agreed and permissioned users to participate in it, has a relatively high speed compared with a public blockchain because of the small number of nodes; private blockchains are further divided into a permissioned blockchain and a consortium blockchain [17]. IoT is implemented as public or private blockchains depending on the field and purpose. If blockchain technology is applied to an IoT environment, it is possible to not only solve cost, scalability, and security issues but also expect high efficiency by ensuring the integrity and transparency of data produced in IoT devices [18–22]. Table 1 shows an example of the benefits of applying the blockchain technology to the IoT environment compared to the centralized system [23].

Table 1. Comparison between traditional the Internet of things (IoT) and blockchain-based IoT.

Feature	Traditional IoT	Blockchain-Based IoT
Scalability	As a central system, there is a limited number of nodes that can be managed and added.	As a distributed system, there are no great limitations on the number of nodes that can be managed and added
Efficiency	It is expensive to process and store data in a central system.	As a distributed system, it can reduce data processing and storage cost compared with a central system.
Stability	If a central server or network fails, the connected devices cannot be used.	Some problems with the network do not affect the entire system.
Security	It is easy to manipulate data, but it is difficult to verify forgery/falsification and restore.	The recorded data in the blockchain is difficult to forge/falsify.

Many projects using blockchain technology in IoT environments are underway to resolve security problems and problems arising from traditional centralized systems. The Tangle-based Internet of Things Application (IOTA) project using the directed acyclic graph protocol has been developing a blockchain platform suitable for IoT environments by addressing the fee issues of traditional blockchain platforms and processing transactions promptly [24,25]. In addition, the Hyperledger Fabric project hosted by the Linux Foundation has developed a platform suitable for IoT environments based on a private blockchain, and applied it to various fields such as logistics, distribution, manufacturing, and finance [26]. Further, other projects such as Streamer, IoT Chain, and Walton Chain are still in progress [27–29].

Blockchain technology is also utilized for the data management, data transactions, access control, and authentication of IoT devices [30]. Low-cost, compact, and lightweight IoT devices such as home IoT or wearable devices have limited computing power and are not suitable for processing encryption protocols or certificates [31]. In this case, the use of blockchain technology can be an effective solution to authenticate low-performance IoT devices securely.

2.2. *Authentication Methods of IoT Devices*

To ensure secure device connections without user intervention in IoT environments, each device must demonstrate its legitimacy and integrity. As for the technologies used for IoT devices, they include Identification (ID)/Password (PW) based authentication, medium access control (MAC) address-based authentication, encryption algorithm-based authentication, challenge response-based authentication, one-time password-based authentication, and certificate-based authentication [32,33].

- ID/PW-based authentication: This is the most general and basic method, and a username and password are used for authentication. It is simple and easy to implement, yet its authentication intensity is low and is easy for attackers to bypass.
- MAC address-based authentication: This method utilizes the unique identification address of a device, the medium access control (MAC) address. After registering the MAC address of devices in an authentication server, authentication is performed by verifying the registered MAC address of devices when authenticating them. Despite being relatively fast compared with other methods, security is low as it is vulnerable to address capture and forgery/falsification attacks.
- Encryption algorithm-based authentication: Authentication is performed based on a public key encryption algorithm or a secret key (symmetric key) encryption algorithm. There are various features owing to the use of multiple protocols or algorithms. A suitable method should be applied by considering complexity and time delay.
- Challenge response-based authentication: It encrypts the random challenge value provided by the server using an algorithm or a secret key, and then transmits it as a response value for authentication. As it uses random values, it provides relatively high security.
- One-time password-based authentication: This is an authentication method using a one-time password (OTP). It uses the same algorithm for a device and an authentication server, and authentication is performed by verifying the match. It is robust against reuse attacks as it generates a new password at the time of authentication; thus, its security is very strong.

3. Limitations of IoT Authentication for Secure Scalability

The authentication methods of traditional IoT devices mainly confirm and verify their legitimacy and integrity. However, they require user or server intervention and do not consider the security level of devices at the time of determining their extension. Attackers can easily bypass ID/PW-based and MAC address-based authentication methods, and the encryption algorithm-based, challenge response-based, and OTP-based authentication methods are not suitable for low-performance IoT devices [34]. Security verification for maintaining the security of IoT devices is time consuming and expensive, and the need for intervention by specialists or accredited agencies rather than by general users reduces its

efficiency [35]. Moreover, there are no security authentication and evaluation systems that satisfy the characteristics of IoT devices legally and institutionally but are only authentication and evaluation systems for each product in most cases. If IoT device's security validation is performed through a smart contract, verification can be relied on without third party intervention. It is also convenient to check the integrity of the white list or the security information of the device and can provide compensation for security maintenance in cryptocurrency. Therefore, there is an urgent need for technology to verify security for known malicious code and security vulnerabilities, and to evaluate the security of IoT devices continuously and securely extend them. For an IoT environment development, a technical solution that can interoperate IoT devices securely and promptly by automatically analyzing security vulnerabilities and verifying security should be provided. Therefore, in this study, we automatically verify and evaluate the security of IoT devices and then record it in the blockchain and propose a method to extend them securely by automating authentication and connection between devices accordingly.

By setting the security level required for IoT devices at the time of authenticating and connecting them, only the devices that satisfy a certain security level are allowed to be connected automatically while connection to devices with low security is restricted, thus inducing manufacturers to maintain security through patches and updates. By determining whether there are some security vulnerabilities of target devices, devices are automatically connected, and the connection is completed after verifying whether the security of the connected devices is suitable, thus ensuring secure interoperability between IoT devices.

4. Scalable and Secure Internet of Things Connectivity

IoT usage can be divided into two types. One is to offer user-centered entertainment and various services by installing applications according to users' convenience and preferences, such as smartphones and smart TVs. The other is to network the social infrastructure and environment such as smart grid, smart city, and smart home. IoT devices can be hacked owing to various vulnerabilities, but malicious programs installed by the user's carelessness or by hackers' attacks account for the most frequent infiltration routes [36]. IoT devices infected with malicious programs such as Trojan viruses, ransomware, and rootkits as well as IoT devices on the connected networks are at high risk of being exposed to attacks. In the upcoming IoT era, a single object will be dynamically connected with surrounding objects in a distributed ad-hoc network. A secure and scalable IoT connection method is required in this environment.

For automated authentication and extension of IoT devices, there must be a trust on the security level and the security evaluation system of IoT devices. The method proposed in this paper evaluates the security level based on the verification of software installed in IoT devices using the whitelist and records it in the blockchain through the smart contract. Assuming that IoT devices adopt an Agent stored in a secure area that is resilient against physical tampering, reverse engineering or compromising by malicious softwares, Figure 1 shows the system configuration for evaluating the security of IoT devices, which is described in the following steps.

Step 1: IoT device manufacturers verify software to be installed on IoT devices in advance and prepare the whitelist. IoT devices of the same type can have the same whitelist and have the whitelist that satisfies the required conditions for each IoT environment. A smart contract is created by including the whitelist prepared by manufacturers and the initial agent hash value (IAHV) of the agent embedded in an IoT device. IoT device manufacturers can access smart contract through Decentralized Applications (dApps), such as web pages. Internally, they can easily access the blockchain through an Application Programming Interface (API). IoT device manufacturers constantly update White List via smart contract.

Step 2: Manufacturers record the whitelist and the IAHV of the agent embedded in an IoT device in the blockchain through the whitelist smart contract (WSC). The IoT device can verify that the IAHV of the agent matches the device agent hash value (DAHV) of the installed agent by inquiring the information recorded in the blockchain through the WSC. It verifies that the agent has not been forged

or falsified by comparing it with the agent hash value stored in the blockchain; thus, the security evaluation process of IoT devices through the agent can be reliable.

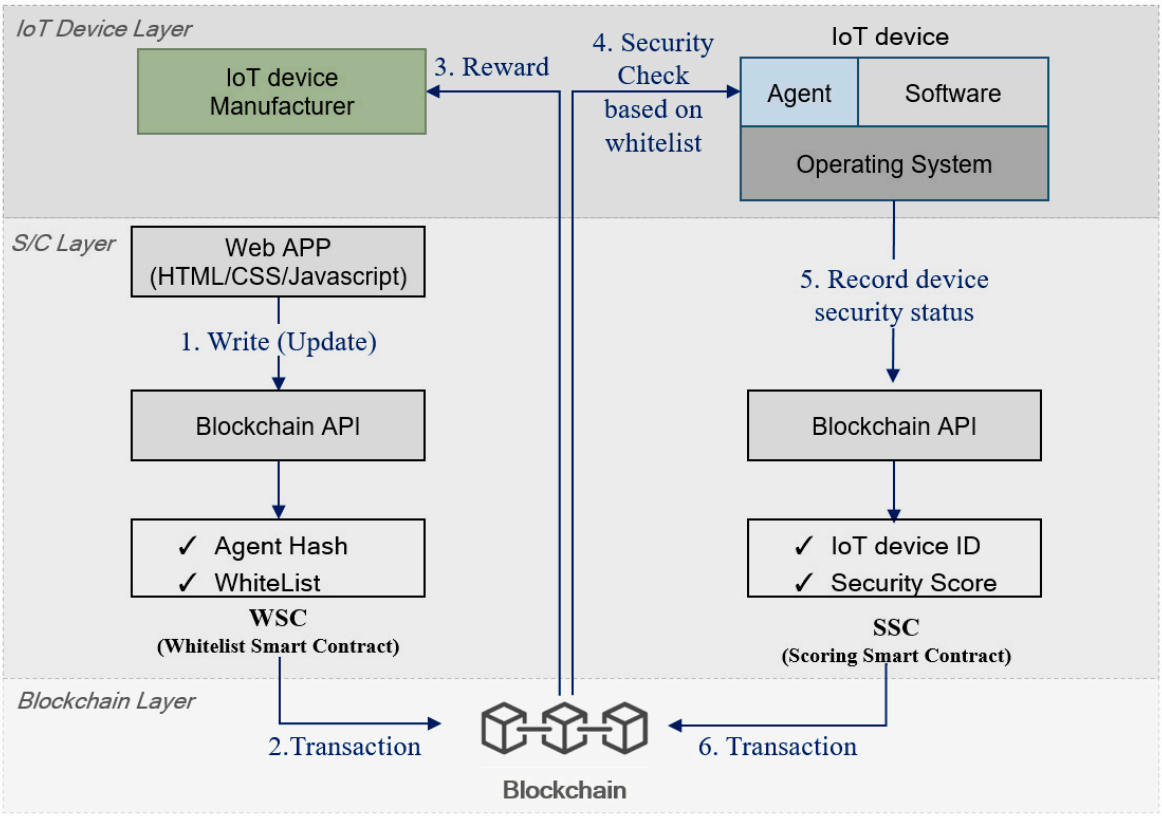

Figure 1. Security level evaluation mechanism for extending the Internet of things (IoT) networks based on a blockchain.

Step 3: Manufacturers are rewarded with tokens whenever they create and update the WSC.

Step 4: Based on the agent information of the device and the whitelist recorded in the blockchain through the WSC, the agent of the IoT device checks whether unverified programs are installed. At this time, the smart contract approaches the most up-to-date state, so IoT device will see the most recent White List.

Step 5: The security status of the IoT device evaluated by the agent is transmitted to the scoring smart contract (SSC) along with the unique identification information of the device and the security level of the device is set through the internal function. When the SSC is used, the agent hash value of the device is checked to verify the integrity.

Step 6: The unique identification information and security level of the device are recorded into the blockchain through the SSC and the security level of the device recorded in the blockchain can be inquired when it is connected to other devices.

The light and secure authentication is an essential and important for IoTs, especially when IoTs are dealing with an immense number of devices. One of the state-of-the-art solutions for IoTs is a Physical Unclonable Function (PUF)-based authentication [37,38]. PUFs are digital fingerprints of IoTs which utilize the physical disorder of random nanoscale phenomena. A PUF is unique to each IoT which cannot be reverse engineered. Therefore, PUF can provide unique identifiers and keys to help secure the massive number of devices found in IoT networks. Thus, the PUF can effectively support IoT authentication. In this research, we assume that an initial state of IoT devices is a 'pure' state which is not contaminated from malicious software because they operate on the verified softwares made by manufacturers. The 'pure' IoT devices can securely connect to others by an authentication technology. However, if a malicious software contaminates an IoT device, the 'contaminated' IoT

device may affect to neighbor IoTs in the network because the conventional authentication cannot detect the contaminated state and prevent from propagating it to others.

Based on this technical background, blockchain can serve as a data store for the hashes of public key certificates of IoT devices. The private keys are stored on the devices themselves assuming that the private keys are PUFs or the keys are stored in the secure zone that are resilient against physical tampering because IoT devices are often used in public locations. In this paper, based on the aforementioned authentication, the proposed scheme provides additional criteria for secure connection and scalability. That is the whitelist/scoring smart contract for device integrity check.

The proposed security evaluation system checks the security of an IoT device twice by first verifying the integrity of the agent installed in the device and then verifying the integrity of the software one more time through the verified agent. The security status verified by the agent is graded as security levels through the smart contract, which is recorded in the blockchain. Based on the recording in the blockchain, the IoT device can be extended safely and quickly when connected to other devices. If device users do not prepare the whitelist or do not update periodically, they will not be rewarded, and the scalability of devices will be reduced. Through this concept of penalty, the security of devices can be steadily maintained, and manufacturers and users can continue to maintain the security of devices to ensure a secure IoT ecosystem.

Figure 2 shows the overall flow of the proposed model. First, an IoT device checks whether whitelisting information on the hash value of software for which security is verified through the WSC and the hash value of the agent were recorded. If there is no record, the IoT device records them in the blockchain by requesting them from the manufacturer. The manufacturer records the hash value of the initial IoT device software as a sha256 function. If they are recorded in the blockchain through the WSC, the manufacturer is rewarded. If the record of the WSC exists, it checks whether there is a record scoring the security of the IoT device through the SSC. If the information recorded in the blockchain through the SSC exists, it attempts to connect to other devices based on this information. A connection is not made if the security level required by the device is not satisfied. If there is no information recorded through the SSC, an integrity verification process is performed in advance to record the security level of the IoT device. If the agent is detected to be forged or falsified, it is alerted to the manufacturer with a request for an update. If the integrity of the agent is verified, the security of the device is evaluated and recorded in the blockchain through the SSC.

Figure 3 shows the structure of the agent. The agent embedded in the IoT device serves to evaluate the security of the device. Therefore, the integrity verification for the agent should be performed in advance for the evaluation of device security. Agent Integrity Check manages the DAHV and compares it with the hash value recorded in the blockchain. This ensures the integrity of the agent. The blockchain API of the agent serves to record information in the blockchain or retrieve the recorded information using the smart contract. If the security of the IoT device is evaluated and then recorded in the blockchain, everyone can directly have access to it. However, the initial IAHV of the device and the whitelist recorded in the block are only accessible through the smart contract. Security Evaluation is responsible for comparing the software installed on the device through Device S/W Monitoring with the verified software recorded in the whitelist, and it calculates the security level by transmitting the verified and unverified software information to the SSC. The process of accessing the blockchain through the smart contract using the blockchain API inside the agent is as follows.

1. The WSC checks the information of the corresponding device and the DAHV verified through the agent integrity check.
2. The WSC checks whether the DAHV matches the IAHV of the device recorded in the blockchain and transmits the whitelist to the agent.
3. The lists of the verified and unverified software through Security Evaluation are transmitted to the SSC along with the device identification value.

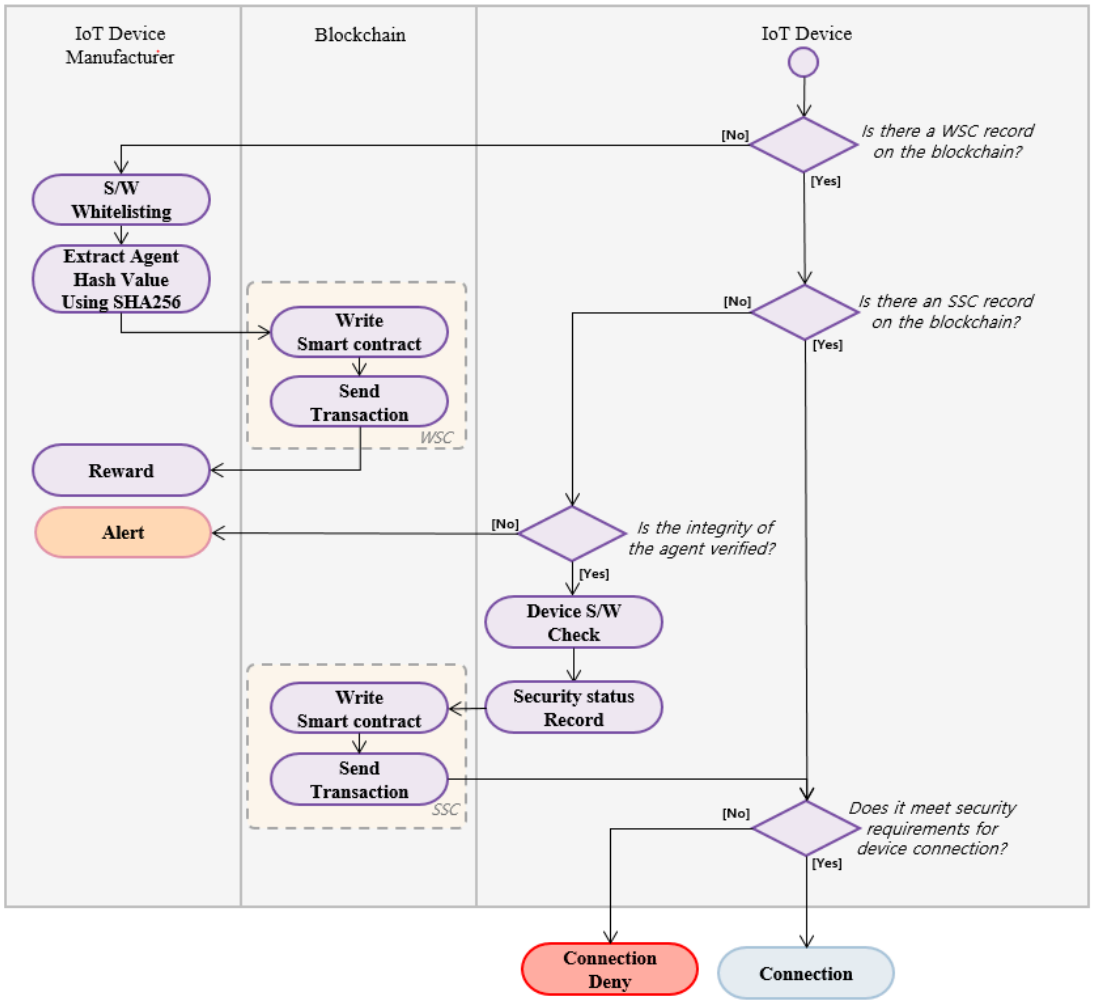

Figure 2. Flowchart for the proposed secure IoT extension.

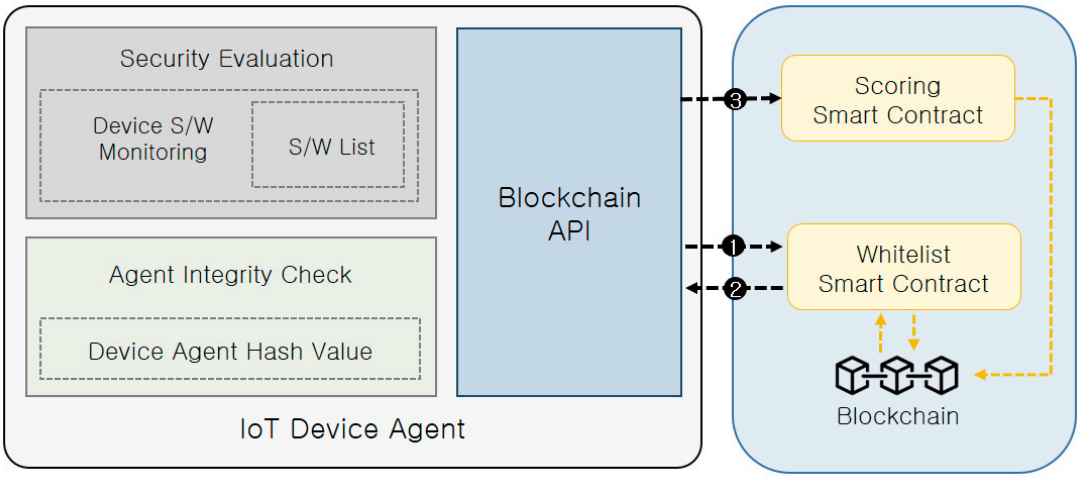

Figure 3. IoT device agent.

Figure 4 shows the WSC. The WSC receives the whitelist including the information of each device and the IAHV from the manufacturer and records them in the blockchain. If data recorded in the block through the WSC from the manufacturer exist, the WSC can be used through the agent of the IoT device. First, the WSC verifies the integrity by comparing the IAHV of the IoT device with the current hash value of the device. If the IAHV does not match the hash value recorded in the block,

it sends an alert message to the device and the manufacturer. If the integrity of the agent is verified, the list for software whitelist of the IoT device recorded in the blockchain is transmitted to the IoT device. The agent makes a list of verified and unverified software through Security Evaluation using the transmitted information and transmits it to the SSC.

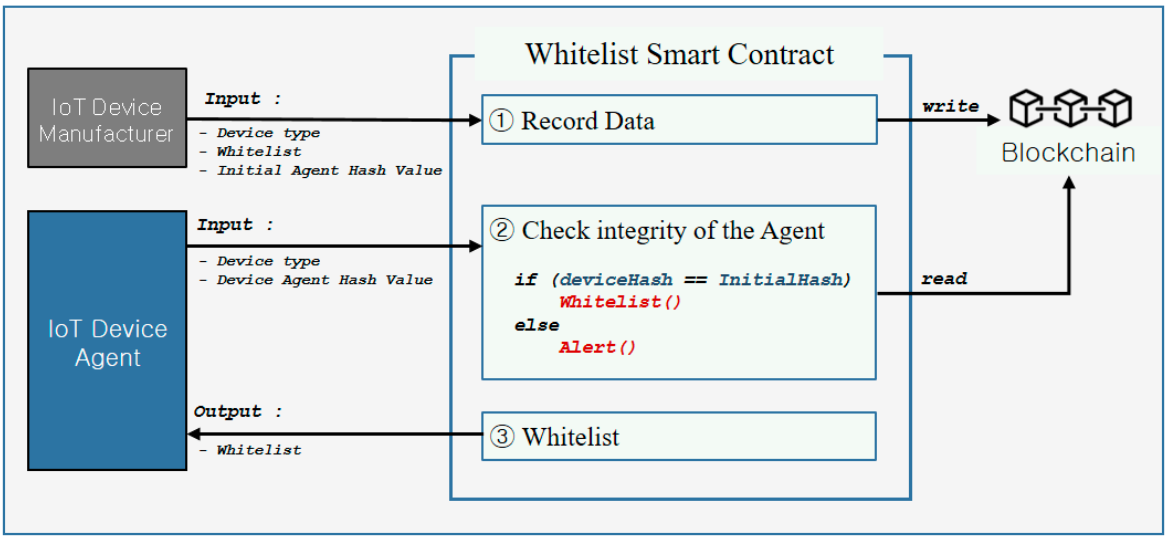

Figure 4. Whitelist smart contract.

Figure 5 illustrates the SSC. The SSC sets the security score of a device based on the software installed on the device and the list of verified software through the agent and sets the security level based on the security score. The security level set through the SSC is recorded in the blockchain along with the device information. The security level is recorded in the IoT device, which can be used to connect to other devices. The integrity can be verified by comparing it with the security level recorded in the blockchain. The security level can be set to a device, which can be used for IoT environments and applications at the time of authenticating and extending the device. The lower the security level of a device is, the more it is restricted to extension. It can be designed to extend proportionally with the security level. For medical IoT device for which security is critical, connections to devices other than Level A can be restricted. If the security score is lowered and the security level of the device changes, a connection is restricted, and the user is notified through the smart contract.

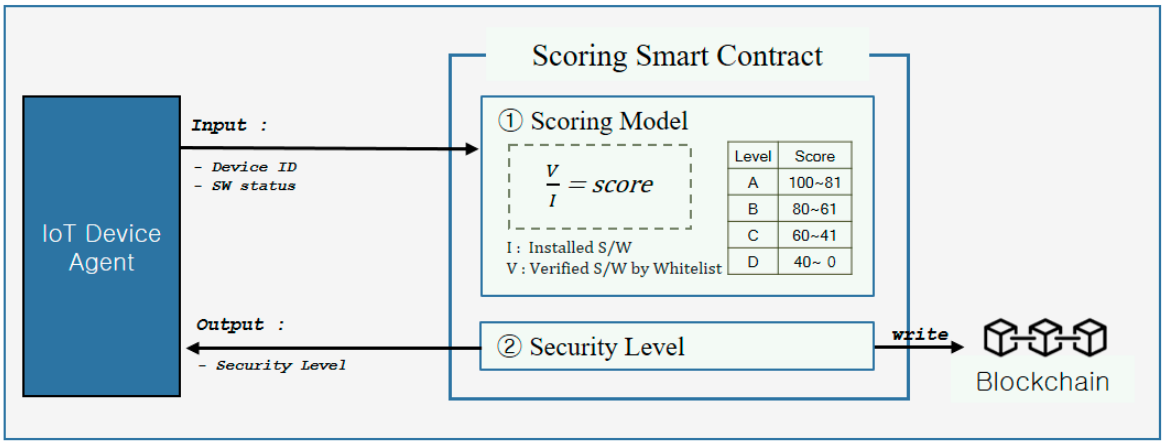

Figure 5. Scoring smart contract.

5. Performance Evaluation

Most authentication methods of traditional IoT devices check whether devices are legitimate by testing the existence of a key. However, this authentication method does not consider security when extending the network and requires user intervention. Therefore, we designed a reliable security evaluation system by automatically verifying security using the whitelist and the smart contract and then recording it in the blockchain.

As IoT devices are used in various fields, security and scalability required by each field are also varied. Therefore, the proposed model ensures maximum scalability when security is high, and restricts scalability when security is low. In this study, we compare the proposed method with the conventional authentication method [39,40] using a network simulation model.

5.1. Evaluation Method

We implemented an abstraction model to evaluate the performance of the proposed model. For network simulation, 200 IoT device nodes were randomly arranged and placed on a 30 × 30 two-dimensional planes. After setting the location information of each IoT device, the devices were allowed to perform connection with only devices capable of physically communicating. The security score is assumed to be recorded in the blockchain through the agent and the smart contract. The security score was set to all 200 nodes and the security level was properly assigned to them as the level A, B, C, and D. Four malicious nodes infected with viruses were also set (two Cs and two Ds). In addition, in order to evaluate security, for the level A node in which the installed software is thoroughly verified, the integrity of all software is verified so that it is set not to be infected by viruses even though it is connected to malicious devices. If it is installed with unverified software, it is set to be infected with viruses even for a simple connection. In addition, when IoT devices were in use, an unproven malicious software would have a significant impact on security. Therefore, based on the assumption that unverified software, i.e., new software that does not exist in the whitelist, is installed after a certain period of time when IoT devices are in use, the security score is set to randomly drop from 10 to 19 points with a probability of 1/50 when 200 nodes make a single connection within the range. The proposed model sets the security score and level according to the number of unverified software. The lower the level is, the more it is restricted to extension. Table 2 shows security scores and levels according to the number of unverified software. The proposed model offers different connection ranges depending on the level and it is possible to connect only to devices of the same level among all the devices within the connection range. In contrast, the comparative model (i.e., conventional model) is set to extend IoT devices to all the devices within the connection range regardless of the security of devices.

Table 2. Evaluation method.

No. of Unverified SW (Score)	Security Level	Connection Range of the Comparative Model	Connection Range of the Proposed Model
0 (100–81)	A	4 hop	4 hop
1 (80–61)	B	4 hop	3 hop
2 (60–41)	C	4 hop	2 hop
over 3 (41–0)	D	4 hop	1 hop

The proposed model checks whether new software has been installed or unverified software is present through the WSC agent, and the security level is given through the SSC. It is extended depending on the security level. If other IoT devices are within the connection range but connection to them fails owing to low security, it requests for the whitelist update.

If the failure of connection to devices within the connection range is continuously accumulated, the whitelist is updated to verify software and the security score is set to increase from 10 to 19 points randomly. In addition, when viruses are removed by the whitelist update, it is set to improve

scalability by increasing the security score. However, there is a case where software has not been verified despite the whitelist update and there are other vulnerabilities although a single vulnerability is improved. To consider these realistic factors, we set different score increase criteria by each security level. The basic update period of the whitelist is determined depending on the cumulative number of connection failures and the update period for the level A device is the same as the basic update cycle. Table 3 shows the criteria for score increase by each level according to the whitelist update cycle. As the level decreases, it has 2, 3, or 4 times of the basic update period as the score increase criteria.

Table 3. Score increase criteria by each level according to the whitelist update period.

Weight Factor for Update Period	0.5	1.0	1.5
Level A	5	10	15
Level B	10	20	30
Level C	15	30	45
Level D	20	40	60

As scalability and security results can vary for the proposed model according to the whitelist update period, a value with the highest network throughput was set as the basic value of the update period to have an appropriate value. Figure 6 shows the throughput simulation results according to the update period. Throughput is the result of the successfully transmitted data size divided by the data transmission time as described in Equation (1).

$$Throughput = \frac{Successfullytransmitteddatasize}{Datatransmissiontime} \quad (1)$$

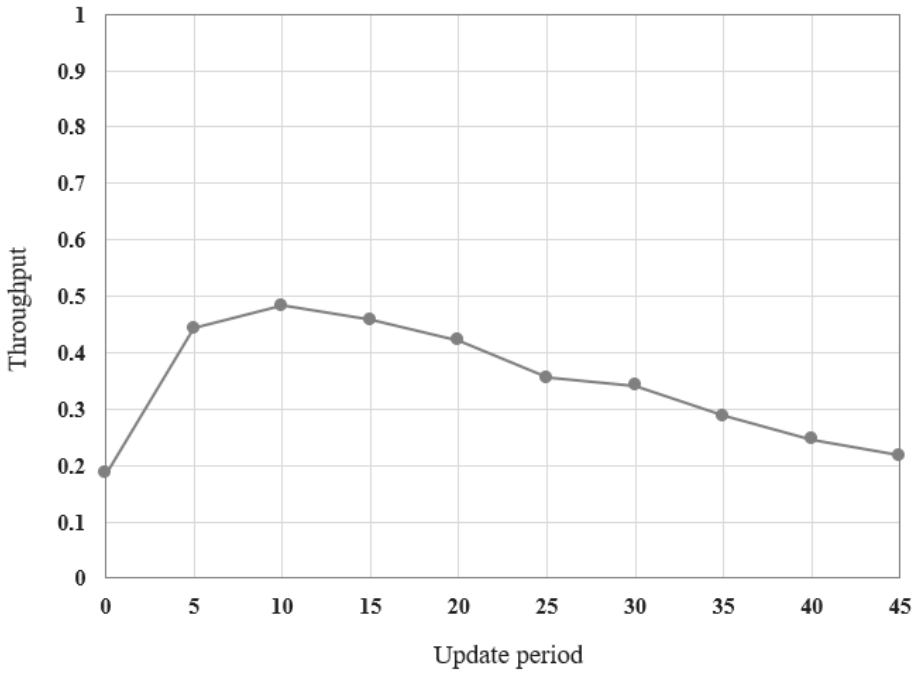

Figure 6. Throughput values.

As the overhead required for sending data increases, throughput is reduced owing to the increased transmission time. In contrast, throughput increases as overhead decreases. IoT devices with excellent scalability can send data to destinations promptly and efficiently, resulting in improved throughput. The restricted connection will reduce throughput. Simulation results demonstrate that the packet overhead for updating is reduced as the update period becomes longer; thus, the network throughput

is improved. However, throughput is improved only when the update period increases to a certain level. Network scalability is restricted when the update period is set to be excessively long, resulting in the decrease in throughput. Therefore, based on the simulation results, the basic update period was set to 10 times as the highest throughput was achieved when 10 connection failures were accumulated and then the whitelist was updated. In this study, as the provided updating period is different depending on the applications and requirements of IoT devices, we also evaluated 0.5 and 1.5 times of the basic period along with the comparative model. Table 3 shows the criteria for score increase by each level according to the whitelist update period. The value of the update period evaluated in the proposed model is divided into 5, 10, and 15, which is the basic period and 0.5 and 1.5 times of the basic period, respectively.

5.2. Evaluation Results and Analysis

In this study, each node is allowed to connect to devices automatically within the connection range in order to evaluate the scalability of the proposed model and the comparative model. The comparative model allows all devices to be connected to each other within the connection range whereas the proposed model allows them to be connected to only the devices of the same level within the connection range according to the security level defined by the proposed method. The conventional (i.e., comparative) model is denoted by C, and the proposed model with 5, 10, and 15 of the update period denoted by P1, P2, and P3, respectively.

Figure 7 shows the results of comparing the security of the proposed model with that of the comparative model when 200 IoT devices in ad-hoc mode attempted to connect to devices within the connection range. Vulnerability is the result of dividing the number of infected devices by the number of all devices as described in Equation (2).

$$Vulnerability = \frac{Maliciousdevices}{Alldevices} \tag{2}$$

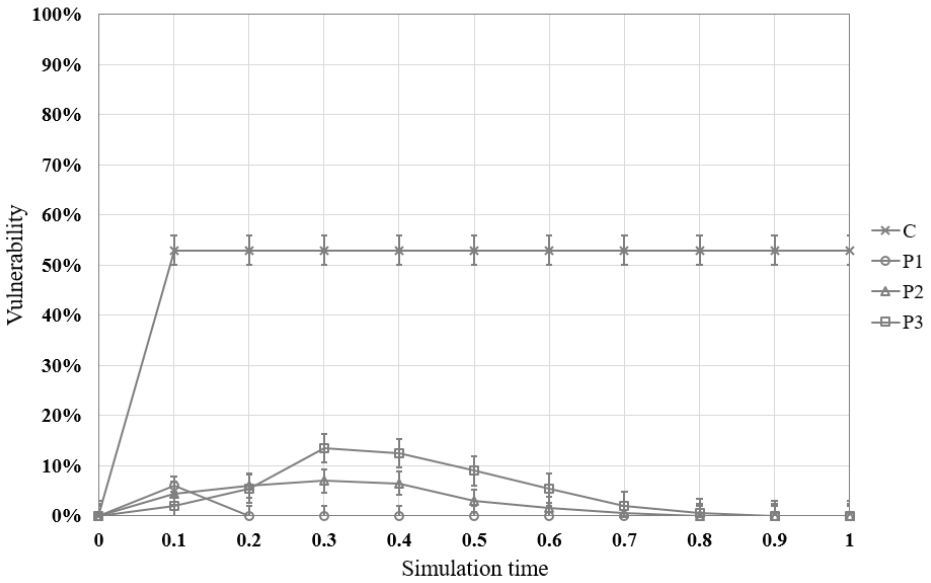

Figure 7. Comparison results of security.

The X-axis represents the number of executions and the Y-axis represents the number of devices infected owing to their connection to malicious devices when 200 devices are connected to the peripheral devices. The error bar represents the standard deviation. In the comparative model, malicious devices were directly connected to all connectable devices and 53% of the devices were infected owing to their connection to malicious devices. However, in the proposed model, the number of devices connected to

malicious devices was reduced by 14% for P3, 8.5% for P2, and 4.5% for P1. The number of devices connected to malicious devices decreased by up to 48.5% for P1. Further, as the proposed model is restricted to connection depending on the security level and continuously verifies software through the agent and the whitelist, the number of malicious devices continues to decrease owing to the whitelist update.

Figure 8 shows the results of comparing the scalability of the proposed model with that of the conventional model when 200 devices attempted to connect to devices within the connection range. Scalability is the result of dividing the number of connected IoTs by the number of physically available connections as described in Equation (3).

$$Scalability = \frac{ConnectedIoTs}{Physicallyavailableconnections} \tag{3}$$

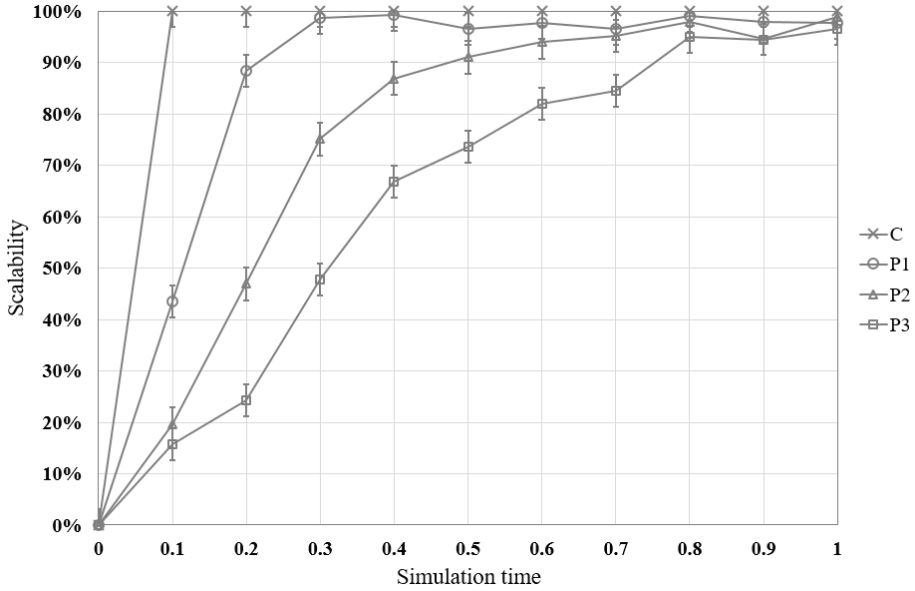

Figure 8. Comparison results of scalability.

The X-axis represents the number of executions, and the Y-axis represents scalability based on the number of connected devices within the connectable range. The error bar represents the standard deviation. If they are connected with all devices within the connection range, they have a value of 100%. In the early connection stage, the proposed model shows very low scalability compared with the comparative model because the devices can be connected with only the devices of the same level and the connection range is also restricted by the level. However, whenever the devices attempt to connect to other devices, devices with low security level are restricted to connection, resulting in the continuous whitelist update. After a certain period of time, most devices will have level A security level, showing similar scalability to the comparative model. As the period to update the whitelist becomes longer, it takes longer to extend the devices by over 90%. As shown in Figure 8, it took 0.23 for P1, 0.5 for P2, and 0.8 for P3 before achieving a scalability of 90% or more.

6. Conclusions

Interest in technological solutions to the scalability and security issues of IoT technology in dense networks continues to grow, and there are also increasing examples of overcoming them using blockchain technology. In this study, we investigated IoT connection technology that could securely extend IoT devices on the fly based on a blockchain by analyzing the limitations of authentication methods of IoT devices. The proposed model can evaluate security by whitelisting software installed

on IoT devices, recording the information in the blockchain through the smart contract, and then verifying the security of IoT devices when authenticating them. Accordingly, the proposed model can also automatically extend them. Manufacturers record a list of software that can be installed on IoT devices in the blockchain, and both verified and unverified software are automatically checked through the embedded agent and WSC when IoT devices are used. This is again recorded in the blockchain through the SSC; thus, the security of IoT devices can be inquired at any place. The results of evaluating security through the proposed model and automatically extending them in proportion to security demonstrate that infection by malicious devices is reduced by up to 48.5%. The proposed model shows better performance than the traditional IoT authentication methods in terms of security and scalability but is relatively slow in terms of extension. In this research, an abstraction model is applied to simplify the complex network problem, and to perform a proof-of-concept of the proposed scheme for demonstrating its feasibility. In a conventional centralized system, a client and a server should gain access to each other using complicated handshake protocols. It is a major cause to increase communication overhead and latency for connectivity. On the other hand, the proposed scheme can simplify the connectivity procedures by using blockchain-based authentication and automating security verification. Thus, it is expected that the proposed scheme may outperform in terms of latency and throughput compared with conventional systems. As further works, we plan to implement and demonstrate the proposed model in a real testbed and to compare the complexity and overhead of performance and implementation with theoretical analysis.

Author Contributions: Y.-J.C. formulated the research idea and implemented and evaluated the simulation model. H.-J.K. studied and formulated the structure of the smart contract and agent. I.-G.L. conceived the basic idea, established the research methodology, and verified the model.

Funding: This work was supported by the Sungshin Women's University Research Grant of 2019-1-82-010/1.

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

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of things
MAC	Medium Access Control
OTP	One Time Password
API	Application Programming Interface
Dapp	Decentralized Applications
IAHV	Initial agent hash value
DAHV	Device agent hash value
WSC	Whitelist smart contract
SSC	Scoring smart contract
PUF	Physical Unclonable Function

References

1. Novo, O. Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT. *IEEE Internet Things J.* **2018**, *5*, 1184–1195. [CrossRef]
2. Fraga-Lamas, P.; Fernández-Caramés, T.M.; Suárez-Albela, M.; Castedo, L.; González-López, M. A Review on Internet of Things for Defense and Public Safety. *Sensors* **2016**, *16*, 1644. [CrossRef] [PubMed]
3. Shah, T.; Venkatesan, S. Authentication of IoT Device and IoT Server Using Secure Vaults. In Proceedings of the 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/ 12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), New York, NY, USA, 1–3 August 2018; pp. 819–824.
4. Yu, Y.; Li, Y.; Tian, J.; Liu, J. Blockchain-Based Solutions to Security and Privacy Issues in the Internet of Things. *IEEE Wirel. Commun.* **2018**, *25*, 12–18. [CrossRef]

5. Rodriguez-Zurrunero, R.; Utrilla, R.; Rozas, A.; Araujo, A. Process Management in IoT Operating Systems: Cross-Influence between Processing and Communication Tasks in End-Devices. *Sensors* **2019**, *19*, 805. [CrossRef] [PubMed]
6. Fernández-Caramés, T.M.; Fraga-Lamas, P. A Review on the Use of Blockchain for the Internet of Things. *IEEE Access* **2018**, *6*, 32979–33001. [CrossRef]
7. Dery, S. Using whitelisting to combat malware attacks at Fannie Mae. *IEEE Secur. Priv.* **2013**, *11*, 90–92. [CrossRef]
8. Powers, J.; Smith, R.; Korkmaz, Z.; Ahmed, H. Whitelist malware defense for embedded control system devices. In Proceedings of the Saudi Arab. Smart Grid (SASG), Jeddah, Saudi Arabia, 7–9 December 2015; pp. 1–6.
9. Stornetta, W.S.; Haber, S. How to Time-Stamp a Digital Document. *J. Cryptol.* **1991**, *3*, 99–111.
10. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 11 March 2019).
11. Pinna, A.; Tonelli, R.; Orrú, M.; Marchesi, M. A petri nets model for blockchain analysis. *Comput. J.* **2018**, *61*, 1374–1388. [CrossRef]
12. Pierro, M.D. What Is the Blockchain? *Comput. Sci. Eng.* **2017**, *19*, 92–95. [CrossRef]
13. Kshetri, N. Can Blockchain Strengthen the Internet of Things? *IT Prof.* **2017**, *19*, 68–72. [CrossRef]
14. Christidis, K.; Devetsikiotis, M. Blockchains and Smart Contracts for the Internet of Things. *IEEE Access* **2016**, *4*, 2292–2303. [CrossRef]
15. Feng, L.; Zhang, H.; Chen, Y.; Lou, L. Scalable Dynamic Multi-Agent Practical Byzantine Fault-Tolerant Consensus in Permissioned Blockchain. *Appl. Sci.* **2018**, *8*, 1919. [CrossRef]
16. Dinh, T.T.A.; Liu, R.; Zhang, M.; Chen, G.; Ooi, B.C.; Wang, J. Untangling Blockchain: A Data Processing View of Blockchain Systems. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 1366–1385. [CrossRef]
17. Meng, W.; Tischhauser, E.W.; Wang, Q.; Wang, Y.; Han, J. When intrusion detection meets blockchain technology: A review. *IEEE Access* **2018**, *6*, 10179–10188. [CrossRef]
18. Gao, F.; Zhu, L.; Shen, M.; Sharif, K.; Wan, Z.; Ren, K. A Blockchain-Based Privacy-Preserving Payment Mechanism for Vehicle-to-Grid Networks. *IEEE Netw.* **2018**, *32*, 184–192. [CrossRef]
19. Kim, S.-K.; Kim, U.-M.; Huh, J.-H. A Study on Improvement of Blockchain Application to Overcome Vulnerability of IoT Multiplatform Security. *Energies* **2019**, *12*, 402. [CrossRef]
20. Casado-Vara, R.; de la Prieta, F.; Prieto, J.; Corchado, J.M. Blockchain framework for IoT data quality via edge computing. In Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems, Shenzhen, China, 4 November 2018; pp. 19–24.
21. Kim, M.-Y.; Jung, D.; Chang, Y.; Choi, D.-H. Intelligent Micro Energy Grid in 5G Era: Platforms, Business Cases, Testbeds, and Next Generation Applications. *Electronics* **2019**, *8*, 468. [CrossRef]
22. Tso, R.; Liu, Z.-Y.; Hsiao, J.-H. Distributed E-Voting and E-Bidding Systems Based on Smart Contract. *Electronics* **2019**, *8*, 422. [CrossRef]
23. Panarello, A.; Tapas, N.; Merlino, G.; Longo, F.; Puliafito, A. Blockchain and IoT Integration: A Systematic Survey. *Sensors* **2018**, *18*, 2575. [CrossRef]
24. IOTA Foundation. Improving the Anonymity of the IOTA Cryptocurrency. Available online: https://iota.org (accessed on 11 March 2019).
25. Novo, O. Scalable Access Management in IoT Using Blockchain: A Performance Evaluation. *IEEE Internet Things J.* **2019**, *6*, 4694–4701. [CrossRef]
26. The Linux Foundation. Hyperledger blockchain for business. Available online: https://www.hyperledger.org (accessed on 11 March 2019).
27. Streamr. Unstoppable Data for Unstoppable Apps: DATAcoin by Streamr. Available online: https://www.streamr.com/ (accessed on 13 April 2019).
28. IoT Chain. IoT Chain: A high-security IoT OS. Available online: https://iotchain.io/ (accessed on 13 April 2019).
29. Waltonchain. Waltonchain White Paper. Available online: https://www.waltonchain.org (accessed on 13 April 2019).
30. Zhou, L.; Wang, L.; Sun, Y.; Lv, P. BeeKeeper: A Blockchain-Based IoT System with Secure Storage and Homomorphic Computation. *IEEE Access* **2018**, *6*, 43472–43488. [CrossRef]

31. Goyal, T.K.; Sahula, V. Lightweight security algorithm for low power IoT devices. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, India, 21–24 September 2016; pp. 1725–1729.
32. El-hajj, M.; Fadlallah, A.; Chamoun, M.; Serhrouchni, A. A Survey of Internet of Things (IoT) Authentication Schemes. *Sensors* **2019**, *19*, 1141. [CrossRef] [PubMed]
33. Mick, T.; Tourani, R.; Misra, S. LASeR: Lightweight Authentication and Secured Routing for NDN IoT in Smart Cities. *IEEE Internet Things J.* **2018**, *5*, 755–764. [CrossRef]
34. Park, N. Mutual authentication scheme in secure internet of things technology for comfortable lifestyle. *Sensors* **2015**, *16*, 1–16. [CrossRef] [PubMed]
35. Jerkins, J.A. Motivating a market or regulatory solution to IoT insecurity with the Mirai botnet code. In Proceedings of the IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 9–11 January 2017; pp. 1–5.
36. Alaba, F.A.; Othman, M.; Hashem, I.A.T.; Alotaibi, F. Internet of Things security: A survey. *J. Netw. Comput. Appl.* **2017**, *88*, 10–28.
37. Bohm, C.; Hofer, M. *Physical unclonable functions in theory and practice*, 1st ed.; Springer: New York, NY, USA, 2012.
38. Aman, M.N.; Chua, K.C.; Sikdar, B. Mutual authentication in IoT systems using physical unclonable functions. *IEEE Internet Things J.* **2017**, *4*, 1327–1340. [CrossRef]
39. Ometov, A.; Petrov, V.; Bezzateev, S.; Andreev, S.; Koucheryavym, Y.; Gerla, M. Challenges of Multi-Factor Authentication for Securing Advanced IoT Applications. *IEEE Netw.* **2019**, *33*, 82–88. [CrossRef]
40. Porambage, P.; Schmitt, C.; Kumar, P.; Gurtov, A.; Ylianttila, M. Two-phase authentication protocol for wireless sensor networks in distributed IoT applications. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Istanbul, Turkey, 6–9 April 2014; pp. 2728–2733.