Research Article
Enhanced SDIoT Security Framework Models

Seongho Choi¹ and Jin Kwak²

¹ISAA Lab., Department of Computer Engineering, Ajou University, Suwon 16499, Republic of Korea
²Department of Cyber Security, College of Information and Technology, Ajou University, Suwon 16499, Republic of Korea

Correspondence should be addressed to Jin Kwak; jkwak.security@gmail.com

Received 3 March 2016; Accepted 17 April 2016

Academic Editor: Fan Wu

Copyright © 2016 S. Choi and J. Kwak. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Following the recent increase in the interest in IoT (Internet of Things) environment, devices that provide various services are being developed. As the diversity, research is required for efficient management of these devices and enhanced security. Previous network environments tend to be dependent on network devices and have difficulties in active processing of variable traffic. Moreover, it is expected that new or variant attacks will increase as packets of various patterns are generated from numerous devices, and hence immense research effort is required in solving this problem. In order to address such problems, this study aims to investigate strategies for establishing a security framework for the configuration of a software-defined IoT environment and efficient provision of security services. Moreover, the service to reduce the overhead involved in security service provision is configured, and a simple test is conducted to verify the feasibility of the proposed model.

1. Introduction

Recently, interest in the Internet of Things (IoT) as a new technology has been increasing, and studies on various devices and solutions related to it have been accelerated [1].

In an IoT environment, different types of packets and variable traffic can be generated from numerous devices including sensors. Accordingly, the issue of proactively controlling such information in the existing network environment is gradually coming to the fore. In addition, concerning security, as an IoT environment has limitations in performance because it is comprised of sensor devices with inferior specifications, there exists a problem that the existing device security technologies are difficult to apply [1–3]. However, if appropriate security actions are not taken, there are risks of information leakage, data forgery/falsification, and large-scale denial of service attack. Accordingly, studies are required on a security frame for safety in an IoT environment [4, 5].

In this paper, a security framework in an IoT environment utilizing a software-defined network (SDN) is studied. SDN is a technology that has been attracting attention as a means to enhance the efficiency of an existing network environment and reduce the difficulty in managing it. The service provided by the existing network control device can be utilized through software, and detailed and proactive network management is possible depending on the software used [6]. In this paper, a security framework is proposed that utilizes the characteristics of such an environment to develop a plan to provide authentication, access control, network and system security, integrity, and confidentiality, and a big-data security analysis system is developed to cope with new attacks.

2. Background

2.1. IoT. IoT can be regarded as a next-generation IT environment that can provide new services by analyzing diverse data collected from the Internet to which all things are connected. IoT is diversely defined by companies and organizations and in general it can be defined as the technology through which information is shared through wire/wireless networks without being limited by both time and space and the entry into a hyperconnected society through interaction with diverse services is realized [3–5].

IoT can be divided into three categories: device (terminal and sensor) domain, network (wire/wireless) domain, and service interface (platform and application) domain. The device domain transmits the data collected and extracted
International Journal of Distributed Sensor Networks

from a specific system using the system’s embedded communication function to other systems. The network domain is the wire/wireless channel through which the data transmitted between the user and a system and between systems are transmitted and received. The service interface domain generates information by processing the data and controls and manages different devices.

2.2. SDN. SDNs are attracting attention as a solution to various problems of the existing network environment. One of the most significant problems is that the different packets generated by various network devices and the variable traffic are difficult to be processed. Expensive network devices are installed in the related domain to process the different packets and variable traffic, causing the network environment to become complex owing to the protocols and services associated with the difference devices, which makes management of the system difficult. In addition, such devices rely on the network vendor who might be unable to have fast response with the user requirements [6].

SDNs involve the concept of separating the control plane and the transfer plane. The Openflow technology of ONF is attracting attention for specific purposes, and diverse technologies are being studied.

The best feature of Openflow is that it separates the control plane and data plane, and the network can be controlled through a controller using the Openflow protocol. This technology can be used to enable automatic communication through setting by reducing the complexity of the environment of the existing network device and defining diverse protocols. If the program for the control plane is modified, free protocols can be generated within the range of four network layers and specific services can be defined [6–12].

Figure 1 shows the SDN framework.

2.3. SDIoT. A software-defined based Internet of Things (SDIoT) framework is an environment used to efficiently manage an IoT environment by combining the infrastructure of the IoT environment with SDN technology. Recently, studies on an SDIoT environment have been carried out and various studies on software-defined controllers are in progress. Jararweh et al. published a paper on the framework of an IoT environment based on different software-defined controller models in which IoT device protocols and network functions can be dealt with. Their model is such that the network is basically managed by software-defined technology and security is provided through a software-defined security (SDSec) [13].

Table 1 shows the problems that may occur in the main IoT environment.

| Category             | Description                                                                                       |
|----------------------|---------------------------------------------------------------------------------------------------|
| Smart home           | (i) Personal information can be leaked by acquiring data from smart home devices  
(ii) Services can be denied and smart home services can be congested owing to data falsification |
| Smart healthcare     | (i) Sensitive data of patients can be leaked by accessing medical devices  
(ii) Patients’ lives can be threatened owing to data falsification |
| Smart car            | (i) Privacy can be breached, for example, by obtaining location information  
(ii) Traffic congestion and accidents owing to data falsification |
| Smart control system | (i) Important information for the different systems controlled by a smart control system can be leaked  
(ii) Large damage such as interruption of electric power control or stoppage of a hydropower system can occur owing to data falsification |

Table 1 shows the problems that may occur in the main IoT environment.

3.2. Increasing DDoS Attack. According to the data analysis report on the DDoS attacks in the third quarter of 2014, which was recently announced by Akamai, traffic volumes increased significantly from the third quarter of 2013 and the new type of DDoS attack has been renewed. Overall, attacks using the UDP or SYN protocol-based layer are increasing, and the occurrence rate is ∼72 million packets per second [14]. In addition, a simple service discovery protocol (SSDP) amplification attack has been recently identified and is known to affect smart TVs, smart cameras, and so forth. Hence, the threat of a large-scale DDoS attack on an IoT environment is expected to increase [15].
Table 2: The rates of increase in DDoS attacks and the bandwidth.

| Category                  | Compared to Q3 2013 | Compared to Q2 2014 |
|---------------------------|---------------------|---------------------|
| DDoS attacks              | 22% increase        | 2% increase         |
| Average attack bandwidth  | 289% increase       | 80% increase        |
| Average peak packets per second | 366% increase     | 10% increase        |
| Application layer attacks | 44% decrease        | 2% increase         |
| Infrastructure layer attacks | 43% decrease     | 2% increase         |
| Average attack duration   | 5% increase         | 29% increase        |
| Multivector attacks       | 9% increase         | 11% increase        |
| High bandwidth (100+ Gbps) attacks | —                 | 183% increase       |

Table 2 shows the rates of increase in DDoS attacks and the bandwidth in the third quarter of 2013 and the second quarter of 2014 [14, 15].

3.3. False Positive and False Negative. Different packets and variable traffic are a result of an increase in the number of devices in the network. It is difficult for existing security systems to accurately detect such new/variant attacks. To deal with such attacks, existing network security systems analyze the patterns of attacks, update the relevant patterns, and cut off the attacks that correspond to the updated patterns. However, such a process is associated with false positive and false negative detections [16].

4. Proposed Security Framework Models

In this study, an SDIoT environment is developed and methods of building a security framework suited to the SDIoT environment are studied. Basically, the controller system is expanded to provide centralized network management and security in order to simplify the environment of the infrastructure. As the control over the network may be lost if there is an overhead in the function used to control the network, a distributed controller environment is developed and the influence of each sector of the distributed environment on other sectors is minimized. Security functions are comprised of authentication and access control, network security, and big-data security analysis. In addition, a simulation test is carried out for the established framework to determine whether the security framework is applicable to the environment.

4.1. SDIoT Network Models. In this study, a network model created by applying the IoT infrastructure to a basic SDN environment is presented. The flow equipment for network connection is operated through the Openflow-Gateway of the SDN and control is achieved using a software-defined controller. An IoT service system is connected to one Openflow-Gateway and is configured in the state where basic access is denied. The sensor devices and the user devices are connected to each other after being authenticated by the controller. In this process, a lightweight authentication process is carried out, and services related to security are set to be carried out by the controller and the Openflow-Gateway. Different security services are configured in the controller.

Figure 2 shows the SDIoT network models.

4.2. Proposed Security Framework Models. The SDIoT security framework model is divided into an infrastructure layer, control layer, and application layer following the form of the existing SDN framework.

The infrastructure layer is where the network devices are located. It is comprised of various sensor devices and user devices and the systems of the IoT service infrastructure. The network devices for data forwarding are controlled by Openflow-based devices.

The control layer provides services for network control. The additional services it provides include authentication and access control, network and system security, confidentiality and integrity verification, and big-data security analysis. Such services are provided by the software developed for the controller, not by the existing network security devices and a server in a separate location.

The application layer comprises service applications for users and administrator applications for system management by the administrator. The service application is used by users to attain IoT services, and the administrator application is used to control the SDIoT environment.

Figure 3 shows the SDIoT security framework model.

4.2.1. Infra. Layer

(A) Openflow-Gateway. Openflow-Gateway is a basic network device that comprises an SDN and is used to forward data. It receives control commands from the controller and is comprised of a flow table for network topology, routing, and traffic permit. The flow table is used by the controller to route devices and connect them with devices capable of communication. Based on this characteristic, it is configured to enable the controller to perform different types of detailed data control such as collection of detailed traffic data and traffic detouring by updating the flow table set in the Openflow-Gateway of the relevant layer when an abnormal event related to security occurs.

(B) Sensor Device. Sensor devices are subdevices basically used to provide IoT services. They mostly collect data and transmit it to the system for IoT services. The types of information collected for the sensor devices are classified depending on the IoT environment, and sensitive data may be collected depending on the environment. Examples of sensitive data that can be collected include specific information...
Figure 2: Proposed SDIoT network models.

Figure 3: Proposed SDIoT security framework models.
that can be used to identify individuals such as medical information. Such information should be safely managed, as failure to do so may cause infringement of privacy.

(C) User Device. User devices are the terminals for IoT service users. They represent the diverse smart devices used to attain services such as general smartphones, wearable devices, personal computers, and notebook computers. They may include specific sensor devices depending on the service and all smart devices equipped with general sensors. Such devices require authentication and identification technologies.

(D) IoT Service Infrastructure. The IoT service infrastructure provides diverse IoT services in the environment where the sensor devices, data collection databases, and servers are combined and makes them available to users. The relevant infrastructure can be built differently depending on the type of the IoT environment and should be controlled and protected by the controller.

4.2.2. Control Layer

(A) Authentication and Access Control Model. The authentication and access control module authenticates the device that attempts to access the network in the IoT environment and provides the verdict for whether or not the service should be provided. In this paper, the security password authentication (SPA) technique of the SDP, one of the SDN technologies, is used to determine whether or not to authenticate a device and allow it to network access.

As SDP technology has a blind effect of providing a server-concealing function for the internal network controlled by the controller, it is difficult for an unauthenticated device to access the network and even more difficult to collect information about the internal environment. When network resources are desired to be used in the relevant environment, an SPA should be carried out, and as an authentication function is provided through a single packet, lightweight services can be provided with low-specification sensor devices.

(i) Authentication and Access Control for Sensor Device. Sensor devices and user devices are given secret SPA information in advance through a safe communication channel in the process of admission or device registration. Each device transmits the SPA information to the gateway before it is connected to the network, which is checked by the controller, and waits for a mutual transport layer security (TLS) connection. In this process, information about the SPA result is not transmitted to the device. If the SPA information is not transmitted, the TLS connection is not executed even if it is requested by the device.

Figure 4 shows the outline of the sensor device authentication and data transmission.

The authentication process is as follows.

Step 1. The sensor device carries out the initial registration process and transmits SPA information to the connected Openflow-Gateway.

Step 2. If the SPA information is authenticated, the sensor device updates the system information that can be transmitted to the flow table of the Openflow-Gateway.

Step 3. The sensor device requests for a TLS connection after transmitting the SPA information.

Step 4. A TLS connection is established between the sensor device and the SDIoT service system.

Step 5. Sensing data is transmitted.

(ii) Authentication and Access Control for User Device. In the case of a smart device with sufficient resources such as smartphones, notebook computers, or personal computers, safety can be achieved utilizing certificate-based authentication and access control system. The relevant function performs authentication using a certificate issued by a third institute as an option. Figure 5 shows an example of a user device with sufficient resources using the service after being authenticated using a certificate issued by a certificate authority (CA).

The authentication process is as follows.

Step 1. The user device obtains a certificate from a third-party CA through the application.
Step 2. The user device transmits the certificate and authentication information through the Openflow-Gateway using the application.

Step 3. The controller verifies the certificate through the third-party CA.

Step 4. If the certificate is valid, the flow table of the Openflow-Gateway is updated.

Step 5. A TLS connection between the user device and the SDIoT service infrastructure is attempted.

Step 6. The IoT service is requested.

Step 7. The IoT service is provided.

(B) Network Security. The proposed SDIoT security framework model provides a security service function for the network through controller software. In general, an IDS, IPS, and a firewall, which are used for network security, are provided in the network environment as software, not as network devices. As many packets and variable traffic can be generated from different devices in such a process, the overhead for the controller system can increase. As network control can be lost if the overhead of the controller increases, the overhead should be reduced and its influence on the control function should be minimized. Accordingly, a means of organizing a system based on a distributed processing system is proposed to provide a network and system security service. The components for network security include a packet and traffic collector, traffic monitor, security analysis software, and log management system.

Figure 6 shows the outline of the proposed network security model.

(i) Traffic Collector. The traffic collector is the basic collector module required for detection of different attacks from packets and traffic in the network. It receives packets from the Openflow-Gateway using mirroring technology and tap equipment. The module may be overloaded when processing a large amount of data. Accordingly, the traffic collector should be managed for different modules to suit the system environment, as shown in Figure 6.

(ii) Security Analysis. The security analysis system detects attacks from the collected packets and traffic and blocks them. The service is provided by configuring the IDS/IPS, firewall, and web firewall, which are the existing security solutions in the analysis module. In addition, as a delay in network traffic may occur in the process of analysis if the amount of data is large because of the increase in computational load, the analysis should be carried out in a distributed environment by adding module agents to fit the scale of the IoT environment.

(C) Integrity and Confidentiality. When providing an IoT service, security threats of data leakage and falsification may arise in the process of collecting and storing data. For this reason, services should be provided to ensure the integrity and confidentiality of data. However, a problem such as electric power loss or overload on the device may occur when a high-performance computation is carried out in an IoT environment due to the characteristics of the device. Accordingly, lightweight encryption and a hashing algorithm for use in a low electric power and low-specification device should be used in an IoT environment. A sensor device transmits data to a gateway to which access is allowed and the gateway checks the flow table information of the relevant device and transmits it to the relevant database system. The encryption and hashing are carried out directly by the relevant infrastructure system, and the key and policy are allocated by the controller.

Figure 7 shows an example of the encryption and integrity verification process during the data storage process using lightweight codes and the hash function.

(D) Big-Data Security Analysis. The big-data security analysis module analyzes correlations by collecting different types of
log information such as that from security logs, diverse access logs, and system use logs. Different types of packet and traffic may occur in an IoT environment as diverse devices use the network function, and this may result in an increase in new/variant attacks. As it is difficult to detect such attacks through the existing security module and positively cope with them, abnormal signs of new/variant attacks must be detected quickly through the big-data security analysis platform.

The big-data security analysis module collects all abnormal logs (attack event logs, logs related to failure in log-in, and logs of system use at a specific time) from each distributed module and detects abnormal signs through the security indexes based on the collected log information.

In this study, logs are collected through log agents in the software from the network security module, authentication module, and network control module, and as the software can collect information related to the devices and network, the environment can be more simplified than that comprised of complicated devices in the existing network [17–19].

Figure 8 shows an example of the big-data security analysis architecture of the proposed security framework.
(i) Log Agents. A log agent is a module that collects log data from each module. It collects the log information stored in each service module. As the log data generated by each system vary, it may be difficult to define the field format when they are stored in a database. Accordingly, it is better to use the non-SQL database.

(ii) Security Matrix. A security index is the data required for a certain attack type, which is put in order based on the collected log information, and can be set differently depending on the type of abnormal sign to be detected. Abnormal signs of data leakage and approach by the attacker can be detected through an analysis performed by combining the information about the connection allowed after the abnormal access attempt and the information about the system use and data transmission records; such information can be designated through segmented studies. Such studies have been performed recently. In addition, new/variant attacks can easily be detected using security indexes.

(iii) Integrated Monitoring. The monitoring system is a module that shows comprehensive information about the IoT environment, information about any attempted attacks, and the results of detecting abnormal signs. The administrator can prepare fallbacks against different attacks by using the relevant monitoring tools and can define the security index rules for accuracy of the analyzed result. The monitoring tool should be mutually linked with the administrator application of the application layer.

4.2.3. Application Layer

(A) User Application. The user application is employed by users to use IoT services, and it supports authentication and service use functions. The relevant application enables the controller to execute SPA before a user device sends a request to the IoT service system and to provide IoT services to the authenticated user device.

(B) Administrator Application. The administrator application supports the administrator in using services of the controller and the network device. The relevant software enables the administrator to check the results of the controller security policy update and comprehensive monitoring and should be managed so as not to allow any person other than the administrator to use it. The administrator can receive different network statuses through the relevant software and check the results of the security policy update and big-data security analysis.

5. Security Analysis

5.1. Authentication and Access Control. As data leakage, data falsification, and other malicious attacks can be committed through unauthorized access to devices in an IoT environment, an appropriate authentication and access control technology should be implemented. In particular, as there are many low electric power/low specification IoT devices, an authentication and access control technology suitable for these is required. In this study, a means of providing
authentication and access control is presented by which the SPA technology of the SDP is applied to configure a security framework. The relevant technology allows different devices to access the network only if they transmit an SPA packet at the beginning of the process. If there is no answer to the relevant information, it is difficult for the attacker to determine whether the attack is successful even when the attacker generates and sends an SPA message randomly. In addition, all the packets received from unauthenticated devices are blocked.

When the SPA process is completed, the controller attempts to establish a TLS connection, which updates the flow table by designating the services, servers, and devices that can be connected to the device that has been authenticated. Hence, it is impossible for the attacker to access other systems through a route other than the designed route in an environment where the controller is protected. Even when the attacker is connected to the network environment through this route, access to other resources is difficult, as the attacker cannot get access to the desired route from the controller.

For safety in the situation above, the controller should be physically protected, and the secret information values for SPA should be safely shared in the new-device registration and update processes. In addition, the SPA information, identification information unique to devices, and additional authentication information should be safely stored and additional security technologies for verification should be applied.

5.2. DDoS Attack Decreases. As the devices that use the network service in an IoT environment increase, the scale of DDoS attack also increases. This may lead to problems in different IoT fields by causing damage greater than that of the existing DDoS attacks.

In this study, as access to the network and traffic are permitted using the SPA function for connection to the network, it is difficult for the attacker to launch attacks because all the traffic is denied unless the information about the SPA registered in the controller is acquired. Accordingly, if the information about the SPA is safely stored, the threat of DDoS attack can be reduced. If DDoS attacks are considered normal behavior circumventing such a method, bandwidth distribution can be utilized as a traffic control policy.

5.3. Abnormality Detection for New Attacks. In an IoT environment where the number of devices using the service increases, the threat of new and different attacks increases in comparison to that for the existing network environment. It is difficult to detect such attacks using the pattern detection technology of the existing security service. New/variant attacks can circumvent the pattern detection rules of the existing security service, which can lead to the problem of false negatives and false positives. In addition, it is difficult to cope with such attacks, as it is difficult to detect them before significant damage actually occurs.

In this study, big-data security analysis technology is used to solve such problems so that the system can be built to detect abnormal signs for new/variant attacks. At present, many companies and researchers are studying big-data security analysis platforms, and several solutions have been developed and supplied. In this study, a big-data security analysis platform is developed and managed as software using the characteristics of a software-defined environment. It can be built by integrating a security service module, an authentication and access control module, IoT service infrastructure, and a control module. Regarding the functional aspect, abnormal signs can be detected by utilizing the data collection technologies of the existing big-data security analysis platform, as they can comprehensively collect system error logs, security event logs, and access logs and analyze their correlation. The correlation analysis is performed using the security indexes, with which countermeasures against new/variant attacks can be set by defining the data to be collected in detail in accordance with the type of abnormal sign to be detected.

6. Simulation Test Performance Analysis

The simulation evaluates the possibility of the functions of the existing security system to be suitably carried out by the software in the proposed security framework model and determines whether the overhead generated in the process of collecting, analyzing, and transmitting the actual data is reduced.

Table 3 shows the environment in which the simulation test is conducted.

6.1. Implementation. The setup for the simulation test of the functional part of the security framework proposed in this paper is shown in Figure 9. The network environment for Openflow of the SDN was built using Mininet 2.0, and KVM technology was used to develop a Hadoop system, a floodlight controller, and a security analysis module. An IP was set for each VM instance through the network bridge in the control layer. The Openflow-switches in the infrastructure layer are controlled by connecting the network to the floodlight controller. Network control is set to enable communication between hosts by setting the flow-entry as the default value. Each host plays the role of replacing an IoT device and continuously communicates with other hosts. Information about the packets and traffic generated during communication is collected through the traffic collection software separated as a VM instance, and the security analysis software performs attack detection based on the collected information.

6.2. Simulation Test. In the test, different hosts continuously exchange messages in the Openflow environment organized as a mininetwork, and several specific attack hosts are designated to carry out SYN Flooding attacks. The security analysis module detects attacks by analyzing the SYN Flooding attacks while the hosts transmit and receive messages to and from each other. In addition, it measures the overhead of the CPU and memory while the data are transmitted to the Hadoop file system for big-data security analysis. To determine an efficient way to build the security framework, the rate of the increase in the CPU and memory usage when the security service is provided from a single controller system must be considered.
The test is carried out as follows.

**Step 1.** One Openflow-switch is configured and two hosts are generated and connected to the controller using Mininet 2.0.

**Step 2.** Host 1 carries out a SYNFlooding attack on Host 2.

**Step 3.** Four hosts are configured, among which two hosts continuously transmit and receive messages while the other two hosts perform SYNFlooding attacks.

**Step 4.** Eight hosts are configured, among which four hosts continuously transmit and receive messages while the other four hosts perform SYNFlooding attacks.

**Step 5.** Steps 2 to 4 are repeated for each switch host, increasing the number of Openflow-switches to 10.

### 6.3. Analysis of Test Results and Performance

In this study, the security service is provided through software and, at the same time, a big-data security analysis function is added to cope with new/variant attacks. However, the control system overhead may increase in the process of collecting information to detect attacks or abnormal signs or in the process of providing security services. If the overhead of the controller system increases, the controller system may stop and control over the network may be lost.

Accordingly, a means of building the system in an environment where the modules are distributed is proposed to reduce the overhead. By conducting an actual test, it was found that the overhead can be reduced by operating the security modules being separated from each other. The test results are shown in Table 4. Consequently, though the CPU and memory use in a controller system in a distributed module environment shows a high occupancy rate when the number of hosts is small, the increase in the occupancy rate drops as the number of hosts and switches increases. This shows that a distributed system has an advantage in computing data in the process of collecting and analyzing the data.

This result shows that the relevant controller functions can be provided using cloud service in the future, and it is expected that security services can be provided and networks can be managed efficiently by virtualizing modules and increasing the number of virtual instances depending on the scale of each field of the IoT environment.

### 7. Conclusion

A security framework for the IoT environment that uses SDN technology has been studied. SDN has attracted attention as a means to improve the efficiency and solve the limitations of the existing network environment. It uses software to organize the services provided by the existing network device through improvement of the existing security control environment. In addition, detailed and positive management of
the network can be achieved depending on the development and setup of the software. In this study, a security framework that can provide authentication and access control, network and system security, integrity, and confidentiality by utilizing the characteristics of such an environment has been proposed, and a means of building it has been studied. In addition, the feasibility of the system has been evaluated and its safety and efficiency have been studied by conducting a test for determining the possibility of using the security service and for determining whether the distributed module environment can reduce the overhead.

The test results indicated that security services could be provided over the entire network through the controller software by improving the existing network environment. In addition, to solve the problem of the overhead in the process of providing the security service, it was concluded that it is relatively efficient to process data in a distributed module environment. To apply the proposed security framework model in the future, additional studies are required on the detailed system configuration and means of efficiently operating the system. To achieve efficiency, the overhead of the control layer during collection and processing of data should be reduced, and studies on the environment where extension is easy should be conducted.

In the situation where service devices are developed in diverse fields as IoT technology advances in the future, many studies are required on the efficient management of devices and organization of the security services and security control environment. The result of this study can be utilized as the base for such studies, and if an optimized environment is developed by modifying the proposed framework to fit various environments of IoT through verification of its safety, a safe IoT environment can be developed.

**Competing Interests**

The authors declare that they have no competing interests.

**Acknowledgments**

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (no. NRF-2014R1A2A1A11050818).

**References**

[1] A. Whittmore, A. Agarwal, and L. Da Xu, "The internet of things—a survey of topics and trends," *Information Systems Frontiers*, vol. 17, no. 2, pp. 261–274, 2015.

[2] R. Want, B. N. Schilit, and S. Jenson, "Enabling the internet of things," *Computer*, vol. 48, no. 1, pp. 28–35, 2015.

[3] S. Li, L. D. Xu, and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.

[4] A. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," *Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016.

[5] B.-I. Jang and C.-S. Kim, "A study on the security technology for the internet of things," *Journal of Security Engineering*, vol. 11, no. 5, pp. 429–438, 2014.

[6] J. H. Yu, W. S. Kim, and C. H. Yun, "A technical trend and prospect of software defined network and openflow," *KNOM Review*, vol. 4, pp. 1–22, 2014.

[7] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, ACM, Monterey, Calif, USA, October 2010.

[8] N. Handigol, B. Heller, V. Jeyakumar, D. Mazières, and N. McKeown, "Where is the debugger for my software-defined network?" in *Proceedings of the 1st ACM International Workshop on Hot Topics in Software Defined Networks (HotSDN ’12)*, pp. 55–60, ACM, Helsinki, Finland, August 2012.

[9] ONF, *Software-Defined Networking: The New Norm for Networks*, ONF White Paper, Open Networking Foundation, 2012.

[10] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, 2013.

[11] N. N. Fernando, J. J. Salvatti, E. Cerqueira, and A. J. G. Abelem, "Management of the existing network environment using open flow control plane," in *Proceedings of the IEEE Network Operations and Management Symposium (NOMS ’12)*, pp. 1143–1150, Maui, Hawaii, USA, 2012.

[12] A. Dixit, F. Hao, S. Mukherjee, T. V. Lakshman, and R. Komella, "Towards an elastic distributed SDN controller," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN ’13)*, vol. 43, pp. 7–12, ACM, Hong Kong, 2013.

[13] Y. Jararweh, M. Al-Ayyoub, A. Darabseh, E. Benkhelifa, M. Vouk, and A. Rindos, "SDidT: a software defined based internet of things framework," *Journal of Ambient Intelligence and Humanized Computing*, vol. 6, no. 4, pp. 453–461, 2015.

[14] D. Perakovic, M. Peria, and I. Ćrtić, "Analysis of the IoT impact on volume of DDoS attacks," in *Proceedings of the XXXIII Simpozijum o Novim Tehnologijama u Poštanskom i Telekomunikacionom Saobraćaju (PosTel ’15)*, Belgrade, Serbia, December 2015.

[15] Akamai, "akamai’s [state of the internet]/security; Q3 [2014 Report] Volume I Number, 2014.

[16] KERIS, "Building security monitoring system using big data analysis at the educational (administrative) institution," RM 2014-21, 2014.

[17] G. Wang, T. S. E. Ng, and A. Shaikh, "Programming your network at run-time for big data applications," in *Proceedings of the 1st ACM International Workshop on Hot Topics in Software Defined Networks (HotSDN ’12)*, pp. 103–108, ACM, Helsinki, Finland, August 2012.
[18] P. Qin, B. Dai, B. Huang, and G. Xu, “Bandwidth-aware scheduling with SDN in hadoop: a new trend for big data,” http://arxiv.org/abs/1403.2800.

[19] C. Tankard, “Big data security,” Network Security, vol. 7, pp. 5–8, 2012.
