A Web-Based Monitoring System of Network Security Functions in Blockchain-Based Cloud Security Systems

1st Jeonghyeon (Joshua) Kim
Dept. of Computer Science and Engineering
Sungkyunkwan University,
Suwon, South Korea
jeonghyeon12@skku.edu

2nd Patrick Lingga
Dept. of Electrical and Computer Engineering
Sungkyunkwan University,
Suwon, South Korea
patricklink@skku.edu

3rd Jaehoon (Paul) Jeong
Dept. of Computer Science and Engineering
Sungkyunkwan University,
Suwon, South Korea
pauljeong@skku.edu

4th Yunchul Choi
Dept.Intelligence and Standards Research
ETRI,
Daejeon, South Korea
cyc79@etri.re.kr

5th JungSoo Park
Dept.Intelligence and Standards Research
ETRI,
Daejeon, South Korea
pjs@etri.re.kr

2022 International Conference on Information Networking (ICOIN) | 978-1-6654-1332-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICOIN53446.2022.9687177

***Abstract*—To prevent security attacks targeting enterprises and services, security managers use Network Security Functions (NSFs) from various manufacturers. For the unified management of these used NSFs, the Interface to Network Security Functions (I2NSF) Working Group of Internet Engineering Task Force (IETF) provides a framework with standard interfaces and data models to easily configure the NSFs. The user can use this framework to easily manipulate and manage NSFs configuration without any security expertise. One of the proposed interfaces is the Monitoring Interface that can be used to monitor the performance and status of the NSFs. We implemented a blockchain network using Hyperledger Fabric to ensure the integrity of the monitoring data of the NSFs delivered through the monitoring interface. In addition, we implemented a web application that can visualize the monitoring data of NSFs in realtime by implementing REST API using the JavaScript-based Hyperledger SDK.**

***Index Terms*—Network Security, Blockchain, Distributed Database, I2NSF, Network Monitoring**

I. Introduction

With the development of 5G, computer network technologies have become faster and more sophisticated, and various services such as self-driving cars and cloud services use advanced communication technologies. However, with the development of these technologies, security attacks and incidents on computer network systems are also increasing. Hackers are conducting many security attacks to hijack sensitive information of companies or to paralyze services. The issue is addressed with the deployment of Network Security Functions (NSFs) as security attack defensers in Network Function Virtualization(NFV) environments.

Various security vendors produce NSFs to provide protection for internet users with different ways of configuration. A network that uses several different vendors to protect their network systems will be difficult to control and manage as a security engineer needs to learn and understand the NSFs from various vendors and environments. Also to provide better and more effective services, the NSFs might have to be integrated with each other. These problems will be difficult without any common solution for controlling and managing the NSFs in a unified and standardized way.

To solve these problems, the Interface to Network Security Functions (I2NSF) Working Group in Internet Engineering Task Force (IETF) proposes standard interfaces and the Corresponding YANG data models to configure and manipulate the NSFs. The I2NSF Working Group introduces the framework for I2NSF in RFC 8329 [1]. The updated framework is proposed in [2] for security management automation with the components and interfaces, as shown in Fig. 1 where the components in I2NSF are:

- **I2NSF User:** The user of I2NSF that configures and manage the NSFs. The user creates the high-level policy to configure the NSFs and observes the status and performance of the NSFs.
- **Security Controller:** The instance that controls the NSFs with the input of high-level policy from the I2NSF User. It translates the high-level policy into low-level policy to be applied as a security service by the NSF.
- **Developer's Management System (DMS):** The vendor's sysytem that provides the security services through the NSFs.
- **I2NSF Analyzer:** The instance that analyzes monitoring data received from the NSFs. It gives feedback informa-

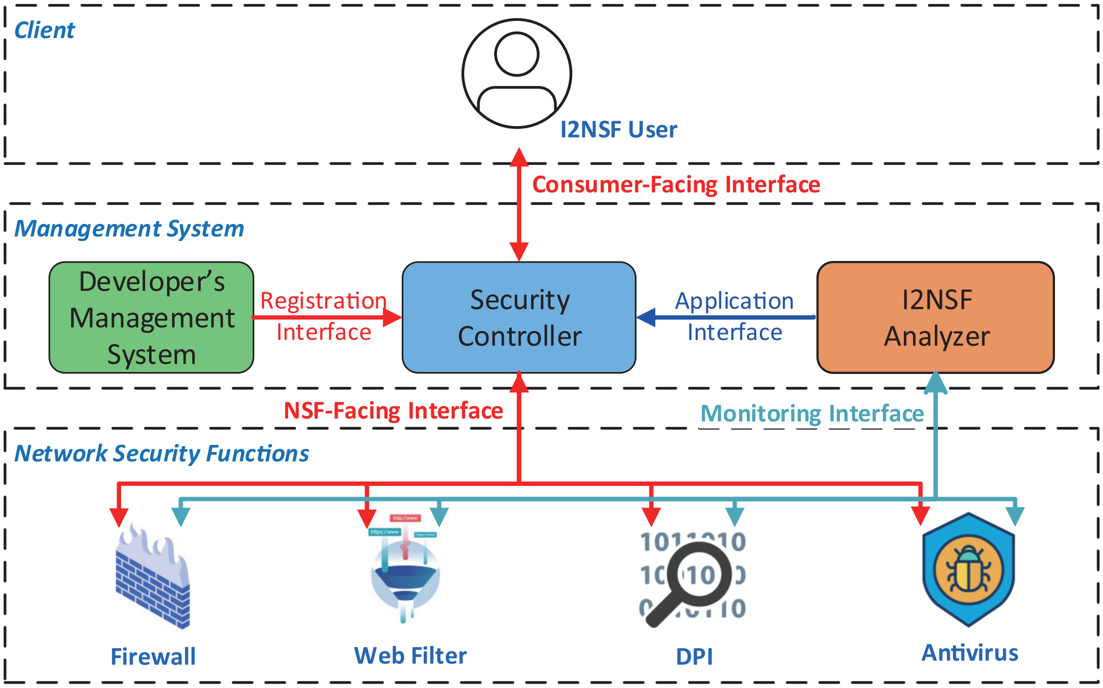


Fig. 1. I2NSF Framework

tion or policy reconfiguration to the Security Controller.

- **Network Security Functions:** The security services that provide protection for the network.

The interfaces in I2NSF are:

- **Consumer-Facing Interface:** To deliver the high-level policy given by the I2NSF User to the Security Controller.
- **Registration Interface:** To deliver the capabilities of an NSF that are provided by a DMS to Security controller.
- **NSF-Facing Interface:** To deliver the low-level policy that has been translated by the Security Controller to a torget NSF
- **Monitoring Interface:** To deliver the data to monitor the status and performance of the NSFs and the network to I2NSF Analyzer.
- **Application Interface:** To deliver the feedback information and policy reconfiguration from the I2NSF Analyzer to the Security Controller.

The previous I2NSF system uses a centralized database that is provided in the Security Controller [3]. As a result, this database system has a security vulnerability, where an attack or problem in the Security Controller may cause security issues in the whole network. For example, if someone is able to attack the database, they can change the NSF's information and status in the database. This will cause a hole in the defense system of the network and may cause easy access to the protected network.

Hence, we propose a blockchain-based system as a distributed database for I2NSF Framework using Hyperledger Fabric. We also tested the system by creating a real-time visualization of the NSF monitoring data. Note that this paper is an extension of our early work for the web-based visualization in the I2NSF system [3]. The contributions of this paper are as follows:

- **The migration of the centralized database to a blockchain-based system for I2NSF Framework.** The proposed solution uses a blockchain-based system as a distributed database to provide an enhanced security in the framework compared to the centralized database.
- **Real-time visualization of NSF monitoring data.** This paper provides the Web-based real-time visualization to observe the performance and status of the NSFs.
- **Evaluation of blockchain-based system for I2NSF Framework.** The system is evaluated to discuss the performance of the blockchain-based system in the I2NSF Framework compared to the centralized database system.

The remainder of this paper is composed as follows. Section II introduces the related work of security policy translation. Section III shows the design and implementation of our proposed scheme. In Section IV, the performance evaluation of the blockchain-based system in I2NSF Framework is discussed. Section V provides the conclusion and the future work of the paper.

II. Related Work

A. Intent-Based Cloud Services for Security Applications

J. Kim et al. [4] propose an Intent-Based Cloud Service for security applications (IBCS). To easily manage various NSFs, IBCS provides an automated and virtualized cloud-based security service. Using the proposed IBCS, a network manager can create and deploy security policies using high-level languages without specific security knowledge for each

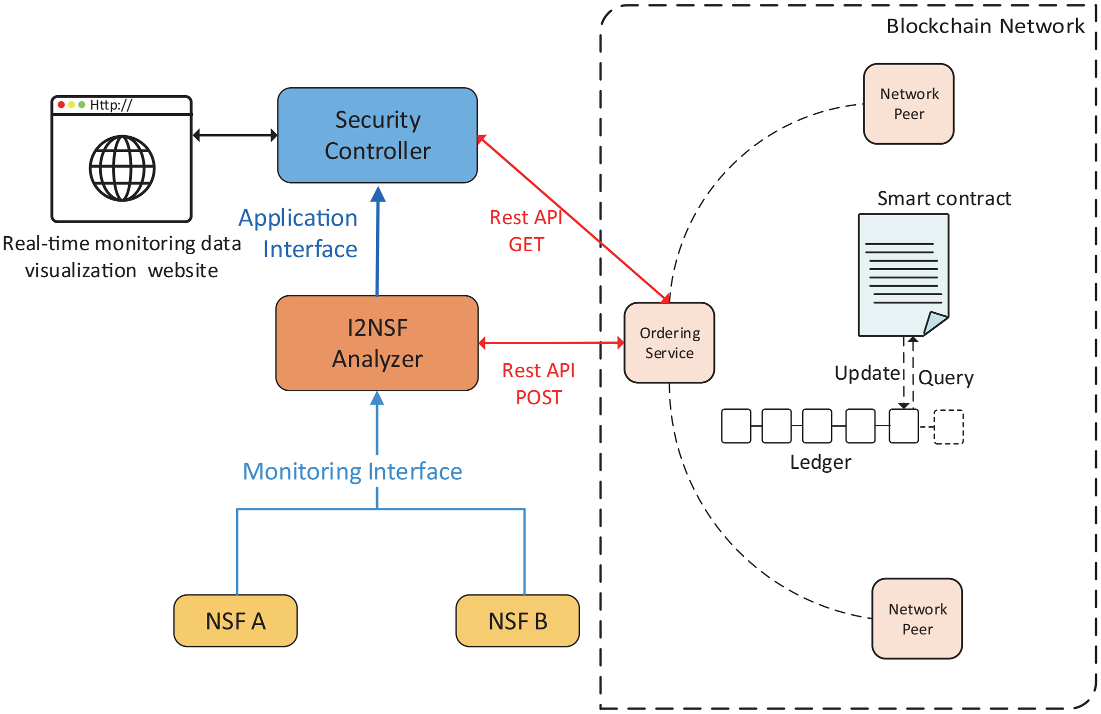


Fig. 2. Blockhain Network-based Architecture for NSFs Monitoring with Web Application

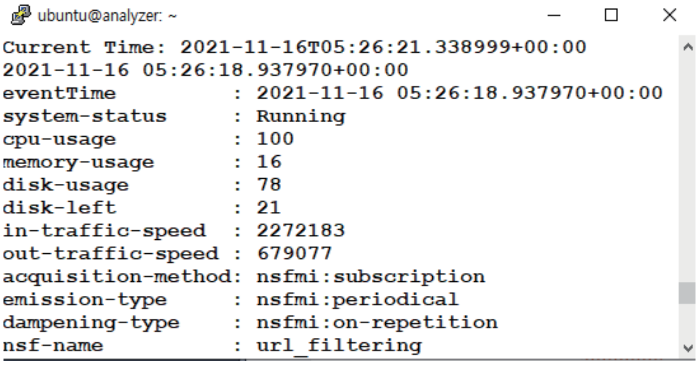
NSF. A security policy written in a high-level language by the user is translated into a low-level policy by a security policy translation in the security controller. The translated low-level policy is distributed to the NSFs through the NSF-facing interface. A blockchain-based real-time visualization of NSFs' monitoring data as a web application is implemented by utilizing the proposed IBCS architecture.

B. Hyperledger fabric: a distributed operating system for permissioned blockchains

This paper proposes an open-source system called Hyperledger Fabric [5], which is a type of private blockchain network. Hyperledger Fabric is a scalable blockchain system for first-time applications. Hyperledger is a decentralized application. It supports a modular consensus protocol. Hyperledger does not rely on cryptocurrency, and unlike existing public blockchains, only authorized people can participate in the Hyperledger Fabric network. Additionally, the shared ledger of the Hyperledger network is shared by all peers participating in the network. When a new transaction is added to the ledger, it must be approved by a certain number of peers. A blockchain network is implemented as a distributed database by using the proposed Hyperledger Fabric to store the NSFs' monitoring data to ensure data integrity.

C. Blockchain-based Database to ensure Data Integrity in Cloud Computing Environments

In [6], a blockchain system is used to solve the data integrity problem in a cloud environment. The authors in [6] have a demonstrated the integrity of the blockchain ledger and have given some examples of how integrity is guaranteed even in cloud environment. However, unlike this paper, our paper implemented and tested a web-based monitoring application using Hyperledger Fabric using blockchain on the I2NSF framework at an IETF standard.



```
Current Time: 2021-11-16T05:26:21.338999+00:00
2021-11-16 05:26:18.937970+00:00
eventTime           : 2021-11-16 05:26:18.937970+00:00
system-status       : Running
cpu-usage           : 100
memory-usage        : 16
disk-usage          : 78
disk-left           : 21
in-traffic-speed    : 2272183
out-traffic-speed   : 679077
acquisition-method: nsfmi:subscription
emission-type       : nsfmi:periodical
dampening-type      : nsfmi:on-repetition
nsf-name            : url_filtering
```

Fig. 3. Example of monitoring data being pushed to a blockchain network

III. Design and Implementation

In this implementation, I2NSF Monitoring Interface YANG Data model [7] is used to provide the monitoring data. To retrieve the monitoring data from the NSFs, the I2NSF analyzer uses the NETCONF subscription [8]. The NSFs will provide the monitoring data periodically or when an event is detected to the I2NSF Analyzer. After receiving the monitoring data, the I2NSF Analyzer will push the data to be stored at the Blockchain Network. Fig. 2 shows a Blockchain Network-based I2NSF Framework for NSF Monitoring.

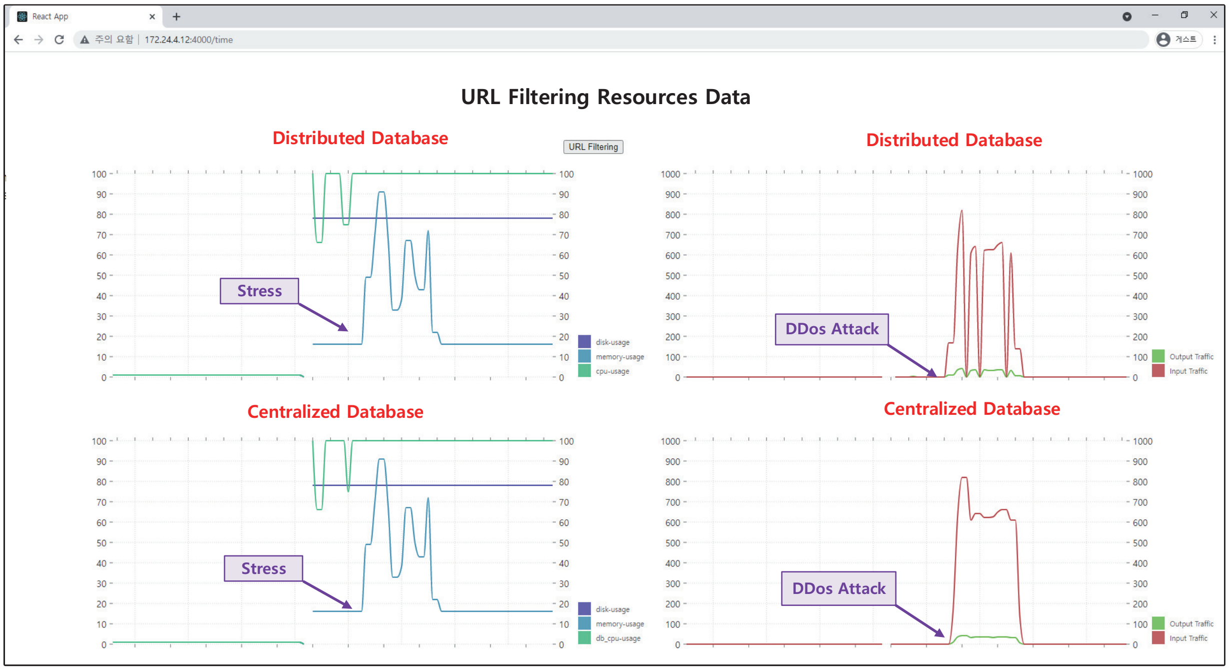


Fig. 4. Real-Time NSF data Monitoring Web Application

The implementation utilizes Hyperledger Fabric version 2.2 with two peers and one orderer in one channel. In a blockchain network, all participating peers share the same ledger through the created channel. The chaincode is implemented using JavaScript to store the NSF monitoring data used in the I2NSF monitoring interface on the ledger of the blockchain network. In addition, REST API was implemented using Hyperledger JavaScript SDK to access the blockchain network. The data can be pushed and pulled from the blockchain network using REST API commands.

In order to push monitoring data from the I2NSF Analyzer to Hyperledger, the data is sent using the POST method in REST API. Fig. 3 shows an example of a monitoring information being pushed to the blockchain network. Data stored in the blockchain network includes event time, system status, CPU usage, memory usage, available disk space, input traffic, and output traffic. The CPU usage, memory usage, and available disk space indicate the hardware resource statuses of the NSFs. Input traffic and output traffic show the network status traffic information that are entering and leaving the NSFs. The data can be stored and managed in the blockchain ledger of Hyperledger Fabric.

Fig. 4 shows the web-based real-time NSF data monitoring through a web application. To display the real-time NSF monitoring data on a web application, we implemented a web front-end server using React in a security controller. This web application for real-time NSF monitoring data visualization visualizes the monitoring data of NSFs in realtime. The upper graph in Fig. 4 visualizes monitoring data of NSFs in realtime using a distributed database based on the blockchain. On the other hand, the graph at the bottom of Fig. 4 shows a graph that visualizes the monitoring data of NSFs using a centralized database (i.e. MySQL). In each graph, when a DDoS attack occurs on the NSF, the in-traffic-speed increases rapidly. Also, when a stress attack occurs on the NSF, we can see that the memory-usage also increases rapidly.

However, in the case of the distributed database using blockchain (i.e. Hyperledger Fabric), saving the monitoring data takes more time, compared to the centralized database. Thus there is a difference in the shape of the real-time graph shown after the DDoS attack occurs.

The data stored in the blockchain network can be queried through REST API's GET method on the React based real-time visualization to be displayed in the web browser of a user. Through the visualized monitoring data of NSFs in real-time, I2NSF users can manage NSFs more efficiently. It is possible to detect abnormal network traffic by security attacks (e.g Distributed Denial of Service (DDoS) attack) and its resource exhaustion through the Real-time NSFs monitoring data visualization website.

In our proposed architecture, I2NSF Analyzer receives resources data (i.e., CPU usage, memory usage, and disk usage) and traffic information (i.e., inbound traffic and outbound traffic) of NSFs every second through the monitoring interface. The transmitted monitoring data is stored in Hyperledger Fabric in the form of a blockchain network through the REST API POST command. For new monitoring data to be added, it must be approved by all nodes participating in the blockchain network.

Due to the endorsement process in the blockchain network between peers and the ledger, the registered transaction can guarantee data integrity. In the centralized database, if one database is corrupted or attacked, the integrity of the NSFs monitoring data cannot be guaranteed. However, our proposed blockchain-based I2NSF framework records all events in the ledgers, so data integrity can be guaranteed. In addition, by visualizing the real-time monitoring data of NSFs stored through the blockchain network, so security administrators and I2NSF users can easily detect a security attack period. Also, we can store all event logs of NSFs in the blockchain network.

IV. Performance Evaluation

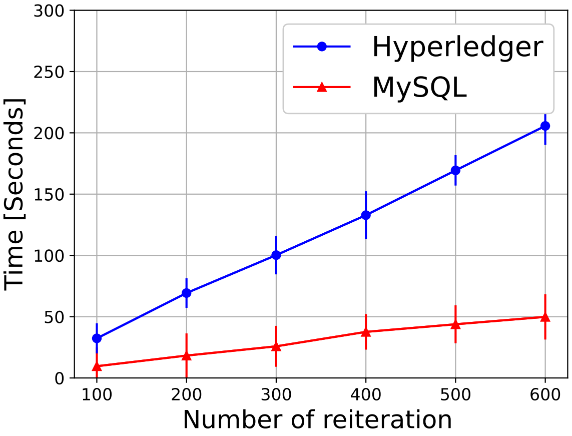


Fig. 5. Data Query Time of Hyperledger to the Number of Reiterations

To evaluate the performance of the proposed blockchain-based I2NSF framework, we conducted an experiment comparing the delay it takes to query data on a blockchain network as a distributed database and MySQL as a centralized database. Fig. 5 show the result of experiment. We tested the response time by increasing the number of queries from 100 to 600. As a result of the experiment, in the case of the blockchain-based I2NSF framework using Hyperledger, it took 0.323 second on average to query one monitoring data. MySQL took about 0.096 second on average to query one monitoring data. Therefore, it was found that the data query speed through MySQL is about 3.3 times faster than the blockchain-based I2NSF framework. The blockchain-based I2NSF framework using Hyperledger guarantees data integrity, but it takes a lot of time in the process of endorsement and ledger propagation between Hyperledger Fabric peers. Therefore, the time it takes to query data takes longer than when using the existing centralized database, but the currently implemented NSF real-time monitoring service operates without any problem because it queries once every second. In order to store and query NSF monitoring data in realtime at faster intervals, it is necessary to improve the performance of transaction processing through the Hyperledger Fabric.

V. Conclusion

We implemented a blockchain network using Hyperledger Fabric instead of the previously developed centralized database to monitor the data of NSFs used in the I2NSF framework in real-time to ensure data integrity. It is implemented so that data can be pushed and pulled using the REST API in realtime. We implemented a service that can detect NSF's performance and status in realtime using a web application. By storing all monitoring data of NSFs in the blockchain network, the integrity of the stored NSF monitoring data can be guaranteed. Due to the nature of Hyperledger Fabric, there is a problem that the delay is longer to push and pull transactions than the centralized database .As future work, we will work to improve the performance of our blockchain-based I2NSF framework.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Ministry of Science and ICT (MSIT), Korea (No. 2020-0-00395, Standard Development of Blockchain based Network Management Automation Technology). This work was supported in part by the MSIT, Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-2017-0-01633) supervised by the IITP. Also, thanks to Moses Gu for assisting in the development of the web front-end with React. Note that Jaehoon (Paul) Jeong is the corresponding author.

References

[1] D. Lopez, E. Lopez, L. Dunbar, J. Strassner, and R. Kumar, "Framework for Interface to Network Security Functions," RFC 8329, Feb. 2018. [Online]. Available: https://rfc-editor.org/rfc/rfc8329.txt

[2] J. P. Jeong, P. Lingga, and P. Jung-Soo, "An Extension of I2NSF Framework for Security Management Automation in Cloud-Based Security Services," Internet Engineering Task Force, Internet-Draft draft-jeong-i2nsf-security-management-automation-02, Aug. 2021, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-jeong-i2nsf-security-management-automation-02

[3] L. Patrick, K. Jeonghyeon, G. Mose, and J. Jaehoon (Paul), "A Web-Based Monitoring System for Network Security Functions in Cloud Security Systems," *KICS-2021-Fall*, 2021.

[4] J. Kim, E. Kim, J. Yang, J. Jeong, H. Kim, S. Hyun, H. Yang, J. Oh, Y. Kim, S. Hares, and L. Dunbar, "Ibcs: Intent-based cloud services for security applications," *IEEE Communications Magazine*, vol. 58, no. 4, pp. 45–51, 2020.

[5] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, 2018, pp. 1–15.

[6] E. Gaetani, L. Aniello, F. Lombardi, A. Margheri, and V. Sassone, "Blockchain-based database to ensure data integrity in cloud computing environments," in *Italian Conference on Cybersecurity*, 2017.

[7] J. P. Jeong, P. Lingga, S. Hares, L. Xia, and H. Birkholz, "I2NSF NSF Monitoring Interface YANG Data Model," Internet Engineering Task Force, Internet-Draft draft-ietf-i2nsf-nsf-monitoring-data-model-11, Oct. 2021, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-i2nsf-nsf-monitoring-data-model-11

[8] H. Trevino and S. Chisholm, "NETCONF Event Notifications," RFC 5277, Jul. 2008. [Online]. Available: https://rfc-editor.org/rfc/rfc5277.txt

Jonghyeon (Joshua) Kim is a Ph.D student in the Department of Computer Science and Engineering, Sungkyunkwan University, South Korea since spring 2021. His Ph.D advisor is Professor Jaehoon (Paul) Jeong. He got a BS degree from Pusan National University. His research interests include Software Defined Networking (SDN), Network Functions Virtualization (NFV), Cloud Native Computing, and Indoor localization.

Patrick Lingga is a Ph.D. student in the Department of Electrical and Computer Engineering of Sungkyunkwan University since 2019. His MS and Ph.D. advisor is Professor Jaehoon (Paul) Jeong. He got a BS degree in July 2019 from Bandung Institute of Technology, Indonesia. His major was Telecommunication Engineering in the Department of Electrical Engineering and Informatics. Currently, his research interests include Software Defined Networking (SDN), Network Functions Virtualization (NFV), Network Security, and Cloud Computing.

Yunchul Choi received a B.S. degree in electrical & computer engineering from Chungnam National University, Daejeon, Korea, in 2007 and the M.S. degree in computer network from Chungnam National University, Daejeon, Korea, in 2010. Since 2012, he has been a Researcher with the Protocol Engineering Center (PEC), ETRI, Daejeon, Korea, and his current research interest includes the testbed construction for future internet.

JungSoo Park is a principal researcher at the Electronics and Telecommunications Research Institute in Korea. He received his Ph.D., his M.S. and his B.S. degree in the Department of Electronic Engineering at Kyungpook National University in 2013, 1994 and 1992, respectively. His research areas are Internet of Things, Blockchain, network security, Software-Defined Networking, Network Functions Virtualization and Internet Protocol version six.

Jaehoon (Paul) Jeong is an associate professor in the Department of Computer Science and Engineering at Sungkyunkwan University in Korea. He received his Ph.D. degree in the Department of Computer Science and Engineering at the University of Minnesota in 2009. He received his B.S. degree in the Department of Information Engineering at Sungkyunkwan University and his M.S. degree from the School of Computer Science and Engineering at Seoul National University in Korea in 1999 and 2001, respectively. His research areas are Internet of Things, Software-Defined Networking, Network Functions Virtualization, security, and vehicular networks. Dr. Jeong is a member of ACM, IEEE and the IEEE Computer Society