Hindawi
Security and Communication Networks
Volume 2019, Article ID 5640134, 11 pages
https://doi.org/10.1155/2019/5640134

Research Article

Virtual Network Embedding Based on Security Level with VNF Placement

Dhanu Dwiardhika [1,2] **and Takuji Tachibana** [1]

[1] *Graduate School of Engineering, University of Fukui, Fukui, Japan*
[2] *Center for Informatics and Nuclear Strategic Zone Utilization, National Nuclear Energy Agency of Indonesia, Jakarta, Indonesia*

Correspondence should be addressed to Dhanu Dwiardhika; dhanud@u-fukui.ac.jp

Received 13 October 2018; Revised 17 December 2018; Accepted 2 January 2019; Published 3 February 2019

Guest Editor: Zbigniew Kotulski

In this paper, in order to embed virtual networks by considering network security, we propose a virtual network embedding based on security level with VNF placement. In this method, virtual networks are embedded in a substrate network by considering security and some security VNFs are placed in order to increase the security level of substrate networks. By using our proposed method, many virtual networks can be embedded by considering security level. As a result, the reward can be increased and the cost of placing VNFs is not increased so much. We evaluate the performance of our proposed method with simulation. The performance of this method is compared with the performance of a method that places VNFs randomly and the performance of a method without placing VNFs. From numerical examples, we investigate the effectiveness of this method. In numerical examples, we show that the proposed method is effective in embedding virtual networks by considering network security.

1. Introduction

In order to provide many services on the Internet [1], the service providers use various devices like servers, router, and switches in a data center environment. In general, software for a specific function is implemented in each device. With such implemented software, each device provides the specific function [2, 3]. Therefore, a larger number of devices are required when a larger number of services have to be provided on the Internet. As a result, the increase of physical devices will lead to an increase of the cost such as CAPEX (Capital Expenditure) and OPEX (Operating Expense) for operating the services. However, the usage of network resources is highly unpredictable. Therefore, most devices have not always been fully utilized [1].

Currently, service providers can use virtualization technology to reduce the hardware cost [4]. By using the virtualization technology, multiple virtualized servers can be implemented in a physical server [5, 6]. The utilization of virtualized servers can decrease CAPEX and OPEX significantly. On the other hand, the utilization of virtualization technology increases security risk [4].

Increasing of Internet users not only brings innovation to human life but also has side effects such as the steal of confidential information or the disablement of infrastructure servers. For example, malware attacks like Stuxnet, Havex, and Dragonfly lead to huge financial loss and reputation damage [7].

Server virtualization also raised new security issues apart from traditional vulnerabilities. The first issue came from the fact that the virtual machine is controlled by the host machine. If the physical machine was under attack or controlled by adversaries, then the virtual machines inside the host could be attacked easily by the attacker [8–11]. The second issue came from the resource sharing for a physical machine. If a virtual machine is attacked, the attacker may attack another virtual machine on the same host and also attacks the physical machine [3, 11]. Some virtual servers may have low-security protection. Therefore, this virtual server can be attacked easily. Users do not want to share resources with other users that have low-security protection [12]. The third security issue in virtualization is that some attacks such as DoS attack may disable a link between two physical

machines [3, 10, 11]. Hence, the security management of the virtualized server is important.

In the virtual network environment, server and network function are also virtualized, and virtualized servers and network functions are encapsulated in a virtual machine. Multiple virtual servers and virtual network functions (VNFs) can be placed in one physical server. Moreover, a link between two virtual nodes can also be virtualized [2]. Virtual nodes and virtual links have to be placed in physical networks efficiently by considering storages availability, bandwidth availability, and security protection [9, 11, 13–15].

In a physical network, security network functions such as firewall, deep packet inspection, and intrusion detection have to be utilized in order to improve the protection for the substrate networks [16–19]. In virtual networks, such security network functions are required to improve the protection. Security network functions especially are expected to be virtualized for virtual networks so as to utilize networks safely. The usage of security VNF can improve the protection of virtual networks [19, 20]. Here, the security of virtual networks has been studied in [10, 11, 15, 19] and some methods have been proposed. With these methods, the virtual network can be embedded in substrate networks by considering the security. However, [10, 11, 15, 19] have not considered the utilization of security VNFs in order to improve the protection of virtual networks. As a result, some virtual networks may not be embedded due to a low security. This results in a smaller number of embedded virtual networks and a low reward by increasing the rejection rate of the virtual network.

In this paper, we propose a virtual network embedding based on security level with VNF placement. In this method, virtual networks are embedded in a substrate network by considering security and some security VNFs are placed in order to increase the security level of substrate networks. Here, virtual networks are embedded by placing security VNFs appropriately with an optimization problem. By using our proposed method, many virtual networks can be embedded by considering security level. As a result, the reward can be increased and the cost of placing VNFs is not increased so much. We evaluate the performance of our proposed method with simulation. The performance of this method is compared with the performance of a method that places VNFs randomly and the performance of a method without placing VNFs. From numerical examples, we investigate the effectiveness of this method. Note that we have considered the virtual network embedding by considering security in [21, 22]. However, in [21, 22], we have considered only the cost of virtual network embedding and do not consider the revenue. Therefore, our proposed method in this paper extends the method in [21, 22]. Moreover, we have not investigated the performance of the previous method in more detail due to page limitation in [21, 22].

The rest of this paper is organized as follows. Section 2 introduces related works about the virtual network embedding with security consideration. Section 3 explains our system model. Section 4 explains our proposed VNF placement with an optimization problem. Section 5 shows numerical examples, and, finally, Section 6 denotes the conclusion.

2. Related Works

Recently, network graph has been used to model network system. Reference [23] has used a network graph for modeling network slice in 5G network. Reference [24] has used a network graph for modeling virtual network over an optical network. Moreover, [25] has used a network graph for modeling a virtual network over flexible grid networks.

For virtual network with security consideration, network graph is also used. In [9], a network graph is used for modeling virtual network with trust value and security protection level. Here, trust value is a metric of physical node that indicates whether the physical node can be trusted. For example, when the trust value of a physical node where virtual nodes are embedded is small, the virtual nodes are attacked by the physical node with high probability. Moreover, the security protection level indicates its capability to defend itself against attacks from other physical nodes. For embedding virtual networks, the trust value and security protection level of the physical nodes have to be equal to or higher than those that are required by the virtual networks. On the other hand, [9] cannot consider the security of virtual nodes because the physical nodes have no requirement for the security of virtual nodes.

Reference [10] has modeled a virtual network as an undirected graph $G^V(N^V, L^V)$ in which N^V and L^V represent the set of virtual nodes and the set of virtual links, respectively. In this model, each virtual node and link requests a certain amount of computing capacity, bandwidth capacity, and a security requirement for the virtual node and virtual link. Here, the security requirement has three levels: high, medium, and none. In the high level, the virtual node cannot be embedded in a physical node that is used by other virtual networks. Moreover, in the medium level, two virtual nodes in the same virtual network cannot be embedded in a physical node. In the none level, the virtual node can be embedded in any physical nodes without security restriction. For the link, the security requirement denotes how encryption is used for embedding virtual links. As is the case with [9], this method also does not consider any requirement for virtual nodes.

Reference [15] has classified the security requirements for virtual network into three requirements. Those three requirements are node requirements, link requirements, and topology requirements. Moreover, they have proposed a method in order to embed virtual networks securely. Reference [19] has proposed a method to improve the security of networks by maximizing utilization of existing security devices. This method requires a monitoring technique for all network flows.

In [11], security-aware virtual network embedding has been formulated as an optimization problem to maximize the long-term profit of virtual network embedding operation. Moreover, the authors have introduced the numerical concept of security level to indicate the standard of protection provided by the network operators. When a security level for the substrate node is high, the high-level protection mechanism is available in the substrate node. The security level of a substrate network has to be equal to or higher than the security level that is required by virtual networks. In

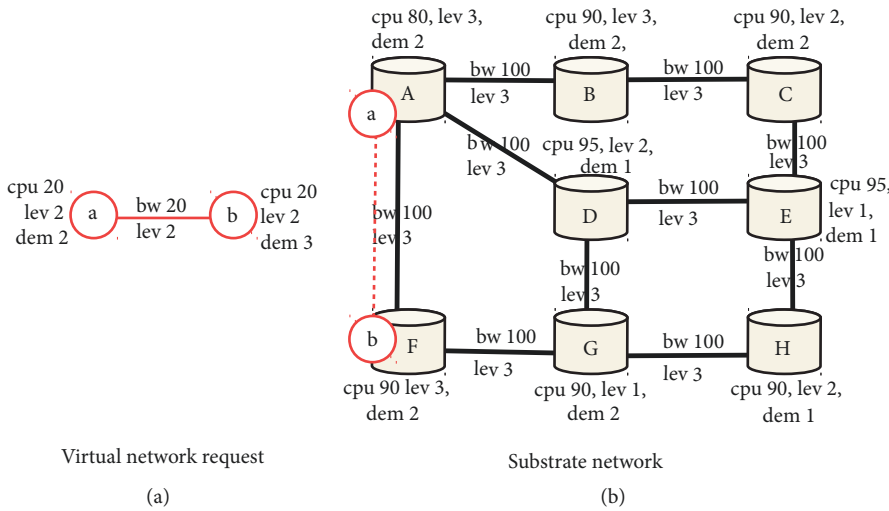

FIGURE 1: Virtual network request and substrate network. (a) Virtual network request a-b. (b) Virtual network a-b is embedded without VNF.

addition, in [11], four security constraints have been provided in the following:

(1) A substrate node should have a security level that is equal to or higher than the demand of every virtual node in that node.

(2) A virtual node should have a security level that is equal to or higher than the demand of its substrate node.

(3) Each virtual node should provide all other virtual nodes of the same host with an adequate security level.

(4) A virtual link with a certain security demand should be embedded in the substrate link with the equal or higher security level.

Those constraints assure that virtual networks are embedded in the substrate network when adequate security protection is available.

Here, if the security level of node or link cannot satisfy the above conditions, a virtual network cannot be embedded and the request for virtual network embedding is rejected. Therefore, in order to embed many virtual networks, the security level is expected to be improved by placing security VNFs. However, in [9–11, 15], such security VNFs are not utilized.

3. System Model

In this section, we model a substrate network as an undirected weighted graph $G^S = (N^S, L^S, A_N^S, A_L^S)$, as shown in Figure 1. In this model, N^S is the set of all substrate nodes and L^S is the set of all substrate links. Each substrate node and substrate link has its own attributes, and we denote attributes A_N^S of the substrate nodes and attributes A_L^S of the substrate links in the following:

$$A_N^S = \left\{\left\{cpu\left(n^s\right), lev\left(n^s\right), dem\left(n^s\right)\right\} \mid n^s \in N^S\right\},$$
$$A_L^S = \left\{\left\{bw\left(l^s\right), lev\left(l^s\right)\right\} \mid l^s \in L^S\right\}. \quad (1)$$

Here, $cpu(n^s)$ is the capacity of CPU of the substrate node n^s and $lev(n^s)$ is the security level, which denotes the standard of protection, of substrate node n^s. Moreover, $dem(n^s)$ is the demand of security level for virtual networks that utilize substrate node n^s. On the other hand, $bw(l^s)$ is the available bandwidth of the substrate link l^s. Moreover, $lev(l^s)$ denotes the security level of l^s. In addition to the attributes, the set of all paths in the substrate network is denoted by P^S.

Moreover, the jth security VNF F_j is denoted with capacity σ_j and security level ζ_j as follows:

$$F_j = \left\{\sigma_j, \zeta_j\right\}. \quad (2)$$

By placing F_j in the substrate node n^s, the security level $lev(n^s)$ of substrate node n^s increases by ζ_j. However, the amount of available CPU decreases by σ_j. Here, by utilizing security VNF F_j in the substrate node n^s we can improve security for virtual nodes in the substrate node n^s.

On such a substrate network, virtual networks are constructed according to user's requests. Here, the ith virtual network requests are denoted as $G_i^V = (N_i^V, L_i^V, Dur_i^V, \mathbf{C}_{i,N}^V, \mathbf{C}_{i,L}^V)$, where N_i^V is the set of virtual nodes, L_i^V is the set of virtual links, and Dur_i^V is the duration for which the virtual network is used. In addition,

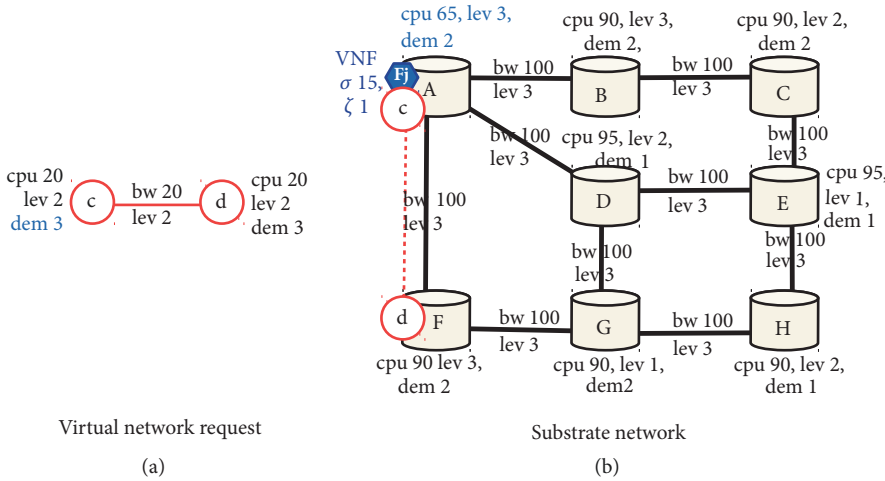

FIGURE 2: Virtual network request and substrate network. (a) Virtual network request c-d. (b) Virtual network c-d is embedded by placing VNF Fj in node A.

$\boldsymbol{C}_{i,N}^{V}$ and $\boldsymbol{C}_{i,L}^{V}$ are the requirements for virtual nodes and virtual links of virtual networks, and those are given by

$$\begin{aligned}\boldsymbol{C}_{i,N}^{V} &= \left\{\left\{cpu\left(n^{v}\right), lev\left(n^{v}\right), dem\left(n^{v}\right)\right\} \mid n^{v} \in N_{i}^{V}\right\},\\ \boldsymbol{C}_{i,L}^{V} &= \left\{\left\{bw\left(l^{v}\right), lev\left(l^{v}\right)\right\} \mid l^{v} \in L_{i}^{V}\right\}.\end{aligned} \tag{3}$$

Note that node n^{v} and link l^{v} denote virtual node and virtual link and do not denote the substrate node and substrate link, respectively. In fact, n^{v} and l^{v} denote only the name of the virtual node and the virtual link. Note that users do not specify the length of the virtual link. Moreover, $len(l^{s})$ denotes the maximum number of substrate links that can be included in l^{s}.

Here, as explained in Section 2, the security level $lev(n^{s})$ of the substrate node n^{s} must be larger than or equal to the security demand $dem(n^{v})$ of the virtual node if n^{v} is mapped to n^{s}. Moreover, the security level $lev(n^{v})$ of the virtual node must be larger than or equal to security demand $dem(n^{s})$ for the virtual node. On the other hand, $lev(l^{s})$ must be larger than or equal to the security level $lev_{i}(l^{v})$ of the virtual link l^{v} if l^{v} is mapped to l^{s}. In terms of bandwidth, $bw(l^{s})$ has to be equal to or larger than $bw_{i}(l^{v})$ if l^{v} is mapped to l^{s}.

In this model, the cost for embedding virtual networks is defined as the amount of resources used by the virtual network in the substrate network. Here, $M_{i} = (\boldsymbol{M}_{i,N}(n^{v}, n^{s}), \boldsymbol{M}_{i,L}(l^{v}, l^{s}))$ denotes the mapping between the ith virtual network embedding request G_{i}^{V} and the substrate network. $\boldsymbol{M}_{i,N}(n^{v}, n^{s})$, especially, shows that n^{v} is mapped to n^{s} and $\boldsymbol{M}_{i,L}(l^{v}, l^{s})$ shows that l^{v} is mapped to l^{s}. Moreover, $\boldsymbol{M}_{\boldsymbol{F}}(F_{j}, n^{s})$ denotes that VNF F_{j} is placed in n^{s}. In this case, the cost is given by

$$\begin{aligned}Cost\left(M_{i}\right) = Dur_{i}^{V}\Bigg[&\sum_{(n^{s},n^{v})\in \boldsymbol{M}_{i,N}} lev\left(n^{s}\right) cpu\left(n^{v}\right)\\ &+ \sum_{(l^{s},l^{v})\in \boldsymbol{M}_{i,L}} lev\left(l^{s}\right) len\left(l^{s}\right) bw\left(l^{v}\right) + \sum_{(F_{j},n^{s})\in \boldsymbol{M}_{F}} \sigma_{j}\zeta_{j}\Bigg].\end{aligned} \tag{4}$$

In addition to the cost, the revenue for G_{i}^{V} is calculated from the resources demanded by the VNE requests with the following equation.

$$\begin{aligned}Rev\left(M_{i}\right) = \alpha \cdot Dur_{i}^{V}\Bigg[&\sum_{(n^{s},n^{v})\in \boldsymbol{M}_{i,N}} dem\left(n^{v}\right) cpu\left(n^{v}\right)\\ &+ \sum_{(l^{s},l^{v})\in \boldsymbol{M}_{i,L}} dem\left(l^{v}\right) bw\left(l^{v}\right)\Bigg],\end{aligned} \tag{5}$$

As shown in (4) and (5), the cost and the revenue depend on the duration Dur_{i}^{V}. In (4), the first term denotes the cost for embedding virtual nodes n^{v} in substrate nodes n^{s} and the second term denotes the cost for embedding virtual links l^{v} in substrate links l^{s}. Moreover, the last term denotes the cost of placing VNF F_{j} in the substrate node n^{s}. In (5), on the other hand, the first term denotes the revenue for embedding virtual nodes n^{v} to n^{s} and the second term denotes the revenue for embedding virtual links l^{v} to l^{s}. Here, α is a ratio of price for a virtual network to cost for the virtual network, i.e., price/cost for a virtual network. The value of α is decided by the provider so that he can obtain a benefit. In (4), the addition of VNFs increases the cost, and hence the provider determines α carefully. If α is equal to or less than 1, the revenue will be less than the cost.

Figure 1 shows how virtual networks are embedded in a substrate network. In this figure, the requirements of virtual node a (cpu 20, lev 2, and dem 2) can be satisfied in substrate node A (cpu 80, lev 2, and dem 2). Moreover, requirements of virtual node b can be satisfied in substrate node F. Substrate link between node A and node F also satisfies the link requirement of virtual network a-b (bw 20, lev 2). Therefore, the virtual network a-b can be embedded in that substrate network as shown in Figure 1(b). On the other hand, the security demand of virtual node c (dem 3) in Figure 2(a) cannot be satisfied in the security level of node A (lev 2). Therefore, in order to satisfy the requirement of virtual node

c, VNF F_j (σ 15, ζ 1) is placed to the substrate node A to increase the security level of substrate node A. However, the placement of VNF F_j decreases CPU resources in node A as shown in Figure 2(b).

For this system model, our proposed virtual network embedding based on security level with VNF placement can be utilized.

4. Virtual Network Embedding Based on Security Level with VNF Placement

In this section, we explain our proposed virtual network embedding based on security level with VNF placement. Our proposed method is used to embed virtual networks so as to decrease the cost (4) and increase revenue (5). In the method, security VNF can be used to embed a lot of virtual networks by increasing the security level of substrate networks. Here, security VNF is a virtual machine that can provide a network security function and the security VNF can be used in a node where it has been installed. By utilizing security VNF in the substrate node, security can be improved for the virtual node. For example, the security VNFs can inspect the network traffic before entering the virtual node so that any malicious traffic can be blocked.

In our proposed method, a lot of virtual networks can be expected to be embedded by using security VNFs so that the difference in revenue (5) minus cost (4) is maximized. For achieving this end, we formulate an optimization problem for embedding virtual networks and placing VNFs as follows:

$$\max_{\boldsymbol{M}_{i,N},\boldsymbol{M}_{i,L},\boldsymbol{M}_F} \sum_{i=1}^{|M|} \left[Rev\left(M_i\right) - Cost\left(M_i\right) \right], \tag{6}$$

subject to

$$\sum_{i=1}^{|M|} \sum_{(n^s,n^v)\in \boldsymbol{M}_{i,N}} cpu_i\left(n^v\right) + \sum_{(F_j,n^s)\in \boldsymbol{M}_F} \sigma_j \le cpu\left(n^s\right), \quad \forall n^v \in N_i^V,\ n^s \in N^S, \tag{7}$$

$$\sum_{(n^s,n^v)\in \boldsymbol{M}_{i,N}} dem\left(n^v\right) \le lev\left(n^s\right) + \sum_{(F_j,n^s)\in \boldsymbol{M}_F} \zeta_j, \quad \forall n^v \in N_i^V,\ n^s \in N^S, \tag{8}$$

$$dem\left(n^s\right) \le \sum_{(n^s,n^v)\in \boldsymbol{M}_{i,N}} lev\left(n^v\right), \quad \forall n^v \in N_i^V,\ n^s \in N^S, \tag{9}$$

$$\sum_{i=1}^{|M|} \sum_{(l^s,l^v)\in \boldsymbol{M}_{i,L}} bw\left(l^v\right) \le bw\left(l^s\right), \quad \forall l^v \in L_i^V,\ l^s \in P^S, \tag{10}$$

$$\sum_{(l^s,l^v)\in \boldsymbol{M}_{i,L}} dem\left(l^v\right) \le lev\left(l^s\right), \quad \forall l^v \in L_i^V,\ l^s \in P^S, \tag{11}$$

Constraint (7) assures the availability of the CPU resources in the substrate nodes. From (7), VNF and virtual nodes cannot be placed and embedded in the substrate nodes when CPU resource of the substrate nodes is less than CPU capacity that is demanded by VNF and virtual nodes. Constraint (8) is used to assure that the security level demanded by the virtual nodes have to be satisfied in substrate nodes. Moreover, constraint (9) assures that the virtual nodes have an adequate security level that is required by the substrate nodes. Constraint (10) is used to assure that the substrate links provide enough bandwidth for embedding virtual links. Moreover, constraint (11) assures that the substrate links have an adequate security level that is required by the virtual links. Thus, the constraint conditions assure that virtual networks can be embedded with sufficient security protection in the substrate networks. In this paper, we solve this optimization problem with a genetic algorithm.

5. Performance Evaluation

In this section, we evaluate the performance of the proposed method in substrate networks shown in Figures 3 and 4. For both topologies, the security level $lev(n^s)$ and demand $dem(n^s)$ for each substrate node n^s are selected at random among [1, 3], respectively. Moreover, CPU of n^s is set at random among [80, 95]. On the other hand, the security level $lev(l^s)$ for each substrate link l^s is also selected at random among [1, 3]. Here, we assume that the addition of a security VNF increases the security level of the substrate node by 1. In our simulation, we denote the security level as integer value due to simplicity although our proposed method is available when the security level is denoted as real number. Note that the effectiveness of our proposed method is affected so much by the difference between the integer value and real number. The bandwidth $bw(l^s)$ is 1000. In terms of VNFs, the maximum number of VNFs that can be placed is not limited. Moreover, $\sigma_j = 5.0$ and $\zeta_j = 1.0$.

For the user's request for virtual network embedding, we assume that the request of virtual network arrives at the substrate network according to Poisson process with rate λ. Moreover, in each request, the number of virtual nodes is selected at random among [2, 7]. The security levels are selected at random among [1, 3], respectively. In each request, $cpu(n^v)$ is decided randomly from 5 to 100 and bandwidth $bw(l^v)$ is selected randomly among [5, 30]. The utilization time of virtual network follows an exponential distribution with rate 10. After the utilization of a virtual network finishes, the virtual networks are removed soon and the used resources are returned to the substrate network.

At each request, the genetic algorithm is used to solve the optimization in (6). By genetic algorithm, we find the position for placing the requested virtual nodes, virtual links, and VNFs so that the value of Rev-Cost is maximized. For Subs8 topology, the genetic algorithm uses chromosome with 32-digit binary strings. The 32-digit binary strings are divided into 8 parts that represent substrate nodes. Each part consists of 5-digit binary strings that represent virtual node and VNF to be embedded in that substrate node. For the FullMesh5, the genetic algorithm uses chromosome with 20-digit binary strings. The 20-digit binary strings are divided into 5 parts that represent substrate nodes. Each part consists of 4-digit binary strings that represent virtual node and VNF to be embedded in that substrate node. We run the genetic

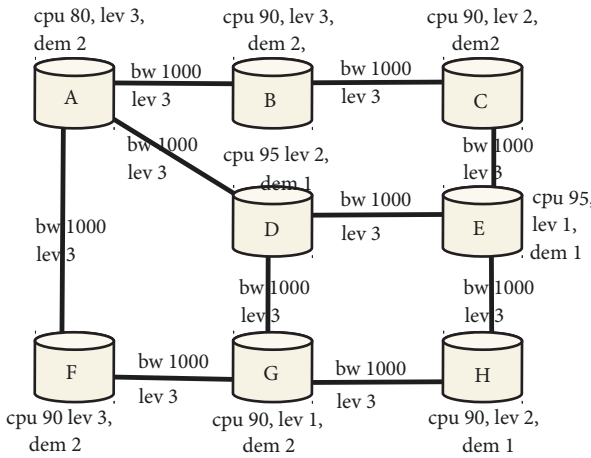

Figure 3: Substrate network with 8 nodes (Subs8 topology).

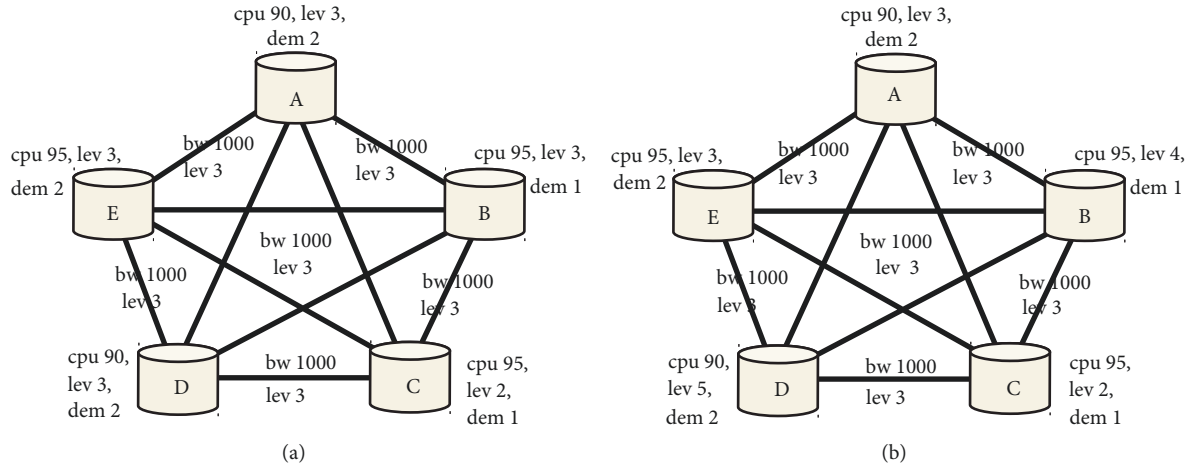

Figure 4: Substrate network with 5-node topology (FullMesh5 Topology). (a) Security level [1, 3]. (b) Security level [1, 5].

algorithm with 100 population size and 5000 generations. We believe that the optimal position was already obtained before 5000th generations by checking the convergence.

For the performance comparison, we evaluate the performance of Random VNF method where VNFs are placed at random in the substrate network. Here, in the Random VNF method, the number of placed VNFs is the same as the number of requested virtual nodes. We also evaluate the performance of No VNF method where VNFs cannot be placed.

We compare the performance of each method in terms of the average value of Rev-Cost, the rejection rate, and the average number of used VNFs. Here, the average value of Rev-Cost is given by the sum of Rev-Cost for the accepted request divided by the number of accepted requests as shown in (12). In (12), $Accept(M_i)$ is one (zero) if request M_i is accepted (not accepted). Therefore, $\sum_{i=1}^{|M|} Accept(M_i)$ is the number of accepted requests. The rejection rate is given by the number of rejected requests divided by the number of arrived requests as shown in (13). In this equation, $|M|$ means the total number of requests. The average number of used VNFs is equal to the number of VNFs that had been used in all accepted requests divided by the number of accepted requests as shown in (14).

$$Rev - Cost = \frac{\sum_{i=1}^{|M|} \left[Rev\left(M_i\right) - Cost\left(M_i\right)\right]}{\sum_{i=1}^{|M|} Accept\left(M_i\right)}, \tag{12}$$

$$Reject\ rate = \frac{|M| - \sum_{i=1}^{|M|} Accept\left(M_i\right)}{|M|}, \tag{13}$$

$$Avg\ VNF = \frac{\sum_{i=1}^{|M|} F_j\left(M_i\right)}{\sum_{i=1}^{|M|} Accept\left(M_i\right)}. \tag{14}$$

5.1. Impact of Arrival Rate. In this section, we investigate the impact of the arrival rate of user's requests on the performance of each method in Subs8 topology.

Figure 5 shows the value of the objective function (6) against the arrival rate λ. In the following, we denote the value of the objective function as Rev-Cost. In this figure, Rev-Cost of the proposed method is higher than those of other

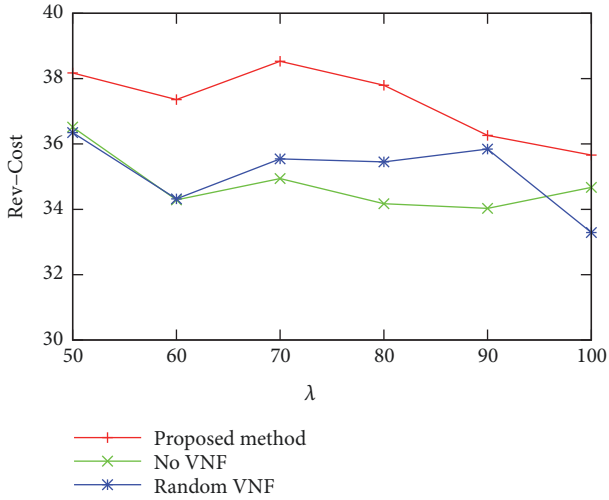

Figure 5: Revenue-cost against arrival rate in Subs8 topology.

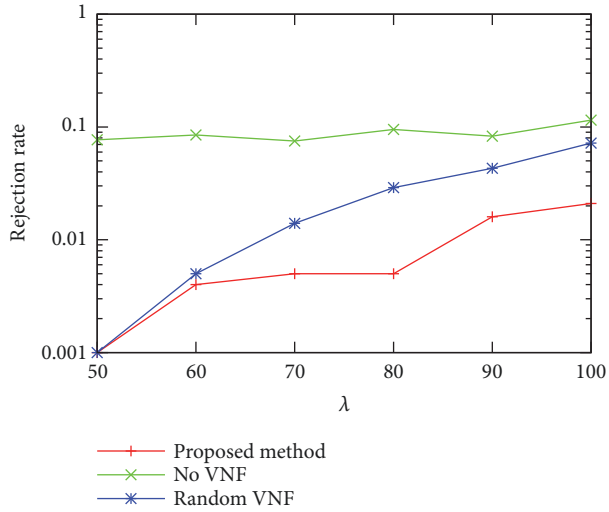

Figure 6: Rejection rate against arrival rate in Subs8 topology.

methods regardless of λ. Highest Rev-Cost can be achieved with efficient placement of virtual network in the substrate network. Therefore, by comparing with the other methods, the embedding for virtual networks using the proposed method is the most efficient.

Moreover, Figure 6 shows that the rejection rate of the proposed method is lower than those of other methods regardless of λ. From this figure, we can find that the rejection rate for the proposed method is smaller than those methods. Therefore, with our proposed method, more virtual networks can be embedded in the substrate network.

Figure 7 shows the average number of VNFs used in the substrate nodes. From this figure, we find that the average number of VNFs that was placed in the substrate network with the proposed method is smaller than those of another method regardless of λ. Note that there is no result for No VNF method because VNFs are never placed in the method. Here, with the proposed method, VNFs are placed in the

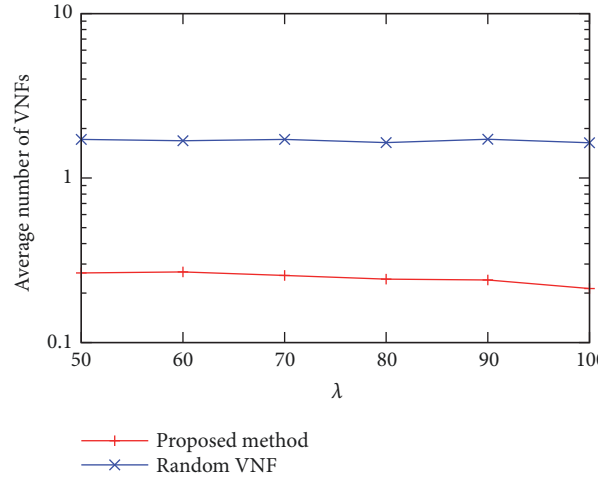

Figure 7: Average VNF used against arrival rate in Subs8 topology.

virtual network so as to satisfy the request. Therefore, the proposed method can embed a lot of virtual networks by using VNFs in the substrate network.

5.2. Impact of Security Level. In this section, we investigate the performance of each method against increment of security level in the substrate node in Subs8 topology. In this evaluation, the security level of each substrate node is multiplied by x. For example, when x is equal to two, $lev(l^s)$ becomes $2lev(l^s)$. This represents that the security is improved as x becomes large.

Figure 8 shows the Rev-Cost of the proposed method is higher than those of other methods even if the security level of the substrate node and link increases. On the other hand, the results of the Random VNF method can be improved when x is large.

Moreover, Figure 9 shows that the rejection rate of the proposed method is lower than those of other methods regardless of the security level of the substrate node and link. This is because our proposed method can place VNFs effectively. Therefore, with our proposed method, more virtual networks can be embedded in the substrate network regardless of the security level of substrate networks.

Figure 10 shows the average VNFs that were placed in the substrate nodes with proposed method are lower than the random method. From this figure, we find that the average number of VNFs used in the network with the proposed method is smaller than those of other methods in a low- or high-security level of the substrate network. Here, with the proposed method, VNFs are placed in the virtual network so as to satisfy the user's request. Therefore, the proposed method can embed a lot of virtual networks by using resources in the substrate network. From these figures, we find that the proposed method has better performance than other methods even if substrate networks have not been designed without the consideration of security.

5.3. Impact of Network Topology. In this section, we use FullMesh5 topology (a) and FullMesh5 topology (b) as shown

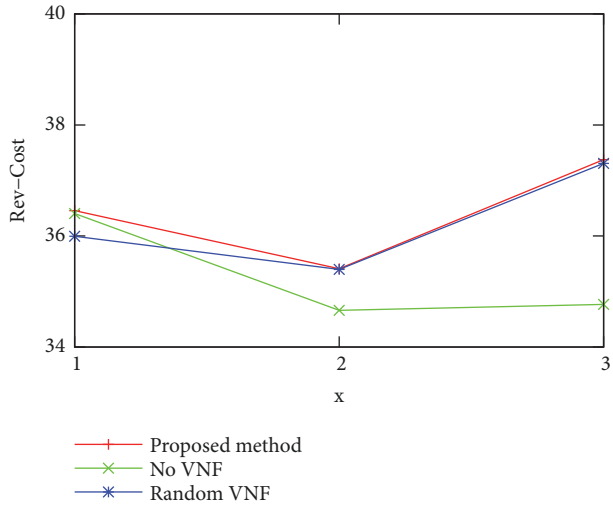

Figure 8: Revenue-cost against security level.

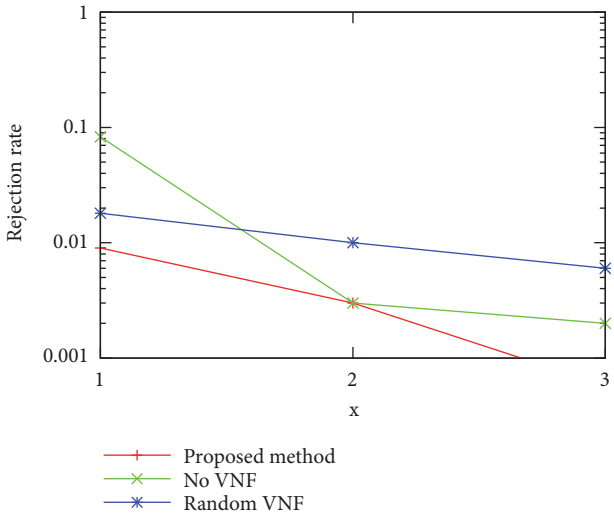

Figure 9: Rejection rate against security level.

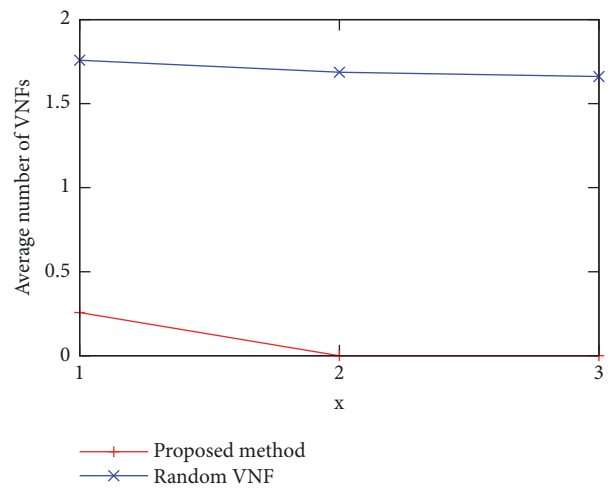

Figure 10: Average number of VNF used against security level.

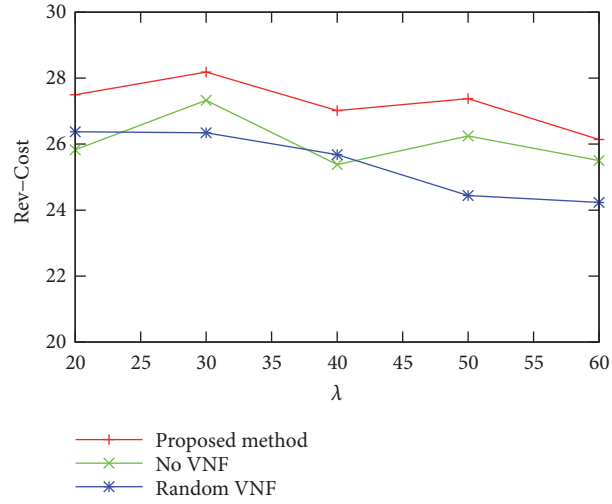

Figure 11: Revenue-cost against arrival rate in FullMesh5 topology (a).

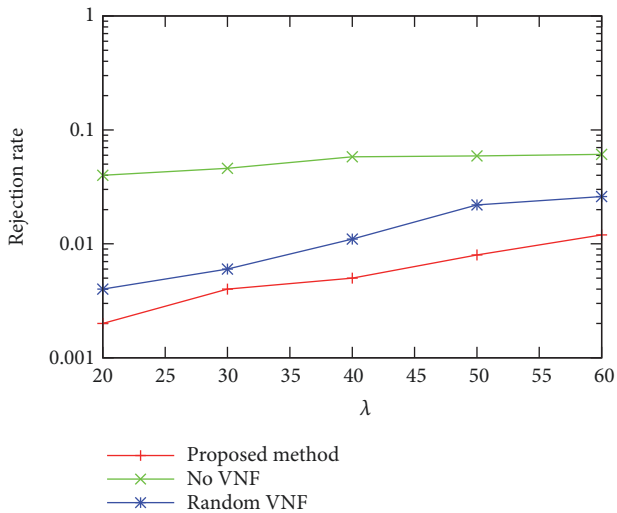

Figure 12: Rejection rate against arrival rate in FullMesh5 topology (a).

in Figure 4(a) and Figure 4(b) to evaluate the performance of each method. For FullMesh5 topology (b) in Figure 4(b), security level is determined at random in [1, 5], a higher range than security level of FullMesh5 topology (a) in Figure 4(a).

Figure 11 shows the Rev-Cost of the proposed method in FullMesh5 topology (a) is higher than those of other methods regardless of λ. This result is similar to the result for Subs8 topology as shown in Section 5.1. Compared with the other methods, the embedding of virtual networks using the proposed method is the most efficient.

Figure 12 also shows that the rejection rate of the proposed method in FullMesh5 topology (a) is lower than those of other methods regardless of λ. Here, the rejection rate is decreased by placing VNFs effective. Therefore, with our proposed method, more virtual networks can be embedded even in the FullMesh5 substrate network.

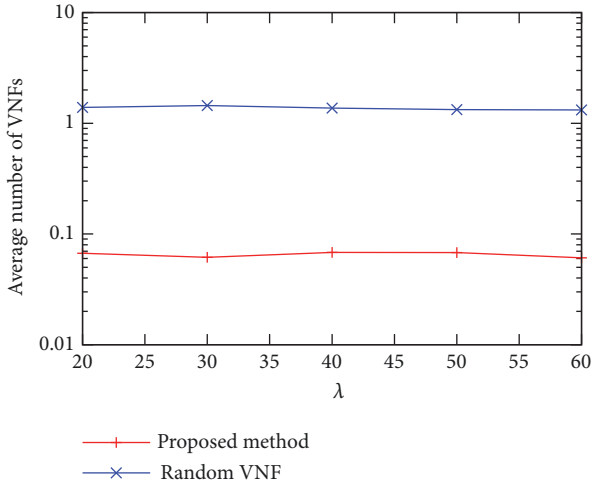

Figure 13: Average number of VNFs used against arrival rate in FullMesh5 topology (a).

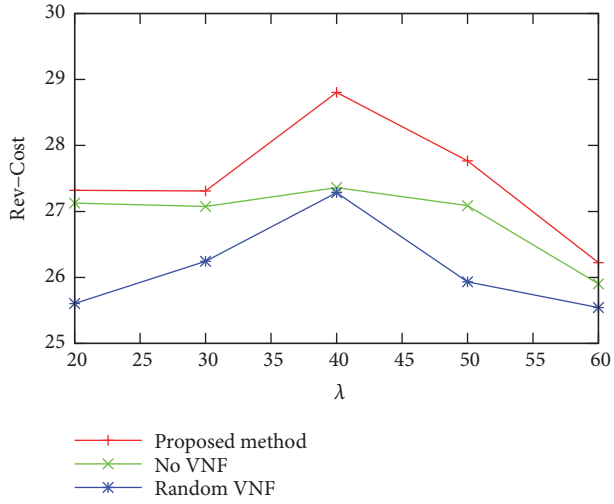

Figure 14: Revenue-cost against arrival rate in FullMesh5 topology (b).

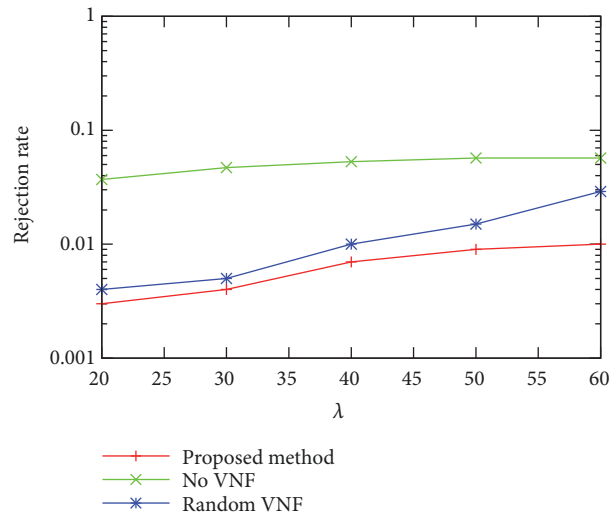

Figure 15: Rejection rate against arrival rate in FullMesh5 topology (b).

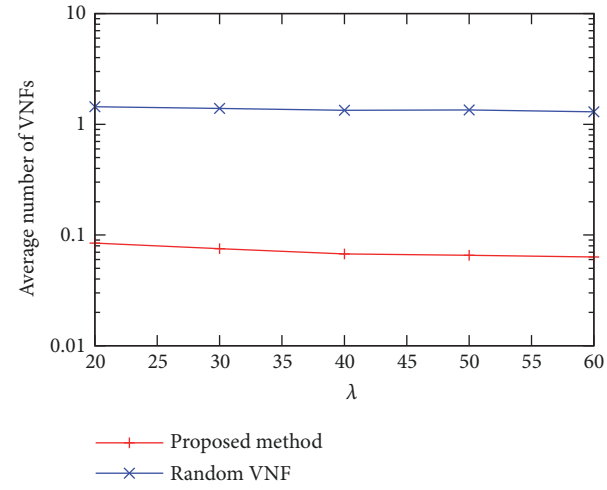

Figure 16: Average number of VNFs used against arrival rate in FullMesh5 topology (b).

Figure 13 shows the average number of VNFs that were placed in the substrate nodes in FullMesh5 topology (a). From this figure, we find that the average number of VNFs for the proposed method is smaller than those for other methods regardless of λ. Here, by using the proposed method, VNFs can be placed in the substrate network to satisfy the request. The proposed method can embed a lot of virtual networks by using resources even in the FullMesh5 substrate network.

Figure 14 shows Rev-Cost of the proposed method in FullMesh5 topology (b) is higher than those of other methods regardless of λ. This result is similar to the result for FullMesh5 topology (a) as shown in Figure 11. Compared with the other methods, the embedding of virtual networks using the proposed method is the most efficient.

Figure 15 shows that the rejection rate of the proposed method in FullMesh5 topology (b) is lower than those of other methods regardless of λ. Here, the rejection rate is decreased by placing VNFs effectively. Therefore, with our proposed method, a larger number of virtual networks can be embedded even in the FullMesh5 substrate network with higher range of security level.

Figure 16 shows the average number of VNFs that have been placed in the substrate nodes in FullMesh5 topology (b). From the comparison with the result in FullMesh5 topology (a), the average number of VNFs shown in Figure 16 is higher than the average number of VNFs in FullMesh5 topology (a) shown in Figure 13. This is because a larger number VNFs are needed to satisfy higher security.

From these results, we find that, with our proposed method, virtual networks can be embedded efficiently even in the different range of security level.

5.4. Calculation Time. Finally, for both topologies, we evaluate the calculation time for solving the optimization problem

Table 1: Calculation time for Subs8 topology and FullMesh5 topology.

Number of virtual nodes in the request	Subs8 Topology	FullMesh5 Topology
2	17 [sec]	13 [sec]
4	23 [sec]	17 [sec]
5	28 [sec]	20 [sec]
8	36 [sec]	-

for one user's request. For Subs8 topology, we change the number of virtual nodes in the user's request to 2, 4, 5, and 8. For FullMesh5 topology, on the other hand, the number of virtual nodes is 2, 4, and 5. Note that we cannot select 8 virtual nodes in the FullMesh5 topology. We calculate the time that is needed from the arrival of request to get the solution of the optimization problem.

Table 1 shows that the calculation time increases as the number of virtual nodes increases. This is because the optimization problem becomes complex. Moreover, the calculation time for Subs8 topology is larger than that for FullMesh5 topology. This means the calculation time becomes large when the number of nodes and the number of links in substrate network are large. However, even if the number of virtual nodes and the size of the topology increase, our proposed method can calculate a request within 1 min in these cases.

6. Conclusions

In this paper, we proposed virtual network embedding based on security level with VNF placement. In this method, some VNFs are placed in order to increase the security level of virtual networks and a lot of virtual networks can be embedded. We evaluated the performance of our proposed method with numerical examples. We found that a larger number of virtual networks can be embedded based on security level by using the proposed method. Moreover, by using proposed method VNE can be embedded with high revenue and less cost.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of the paper.

Acknowledgments

This research is supported by University of Fukui, National Nuclear Energy Agency of Indonesia (BATAN), and Research and Innovation in Science and Technology Project (RISET-PRO), Ministry of Research, Technology, and Higher Education of Indonesia.

References

[1] M. D. Ananth and R. Sharma, "Cost and Performance Analysis of Network Function Virtualization Based Cloud Systems," in *Proceedings of the 7th IEEE International Advanced Computing Conference, IACC 2017*, pp. 70–74, India, January 2017.

[2] R. Mijumbi, J. Serrat, J. Gorricho et al., "Network function virtualization: state-of-the-art and research challenges," *IEEE Communications Surveys &Tutorials*, vol. 18, no. 1, pp. 236–262, 2016.

[3] A. Aljuhani and T. Alharbi, "Virtualized Network Functions security attacks and vulnerabilities," in *Proceedings of the 7th IEEE Annual Computing and Communication Workshop and Conference, CCWC 2017*, USA, January 2017.

[4] S. H. Li, D. C. Yen, S. C. Chen et al., "Effects of Virtualization on Information Security," *Computer Standards & Interfaces*, vol. 42, pp. 1–8, 2015.

[5] T. Kuo, B. Liou, K. C. Lin, and M. Tsai, "Deploying chains of virtual network functions: On the relation between link and server usage," in *Proceedings of the IEEE INFOCOM 2016 - IEEE Conference on Computer Communications*, pp. 1–9, San Francisco, CA, USA, April 2016.

[6] S. Clayman, E. Maini, A. Galis, A. Manzalini, and N. Mazzocca, "The dynamic placement of virtual network functions," in *Proceedings of the IEEE/IFIP Network Operations and Management Symposium: Management in a Software Defined World, NOMS 2014*, Poland, May 2014.

[7] M. Ehrlich, L. Wisniewski, H. Trsek, D. Mahrenholz, and J. Jasperneite, "Automatic mapping of cyber security requirements to support network slicing in software-defined networks," in *Proceedings of the 22nd IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2017*, pp. 1–4, Cyprus, September 2017.

[8] A. Bousselham and T. Sadiki, "Security of virtual networks in cloud computing for education," in *Proceedings of the 2014 International Conference on Web and Open Access to Learning, ICWOAL 2014*, UAE, November 2014.

[9] C. Xing, J. Lan, and Y. Hu, "Virtual network with security guarantee embedding algorithms," *Journal of Computers (Finland)*, vol. 8, no. 11, pp. 2782–2788, 2013.

[10] Y. Wang, P. Chau, and F. Chen, "Towards a secured network virtualization," *Computer Networks*, vol. 104, pp. 55–65, 2016.

[11] S. Liu, Z. Cai, H. Xu, and M. Xu, "Security-aware virtual network embedding," in *Proceedings of the ICC 2014 - 2014 IEEE International Conference on Communications*, pp. 834–840, Sydney, Australia, June 2014.

[12] C. Beşiktaş, D. Gözüpek, A. Ulaş, and E. Lokman, "Secure virtual network embedding with flexible bandwidth-based revenue maximization," *Computer Networks*, vol. 121, pp. 89–99, 2017.

[13] A. Haider, R. Potter, and A. Nakao, "Challenges in resource allocation in network virtualization," in *Proceedings of the 20th ITC Specialist Seminar*, Vietnam, May 2009.

[14] G. P. Alkmim, D. M. Batista, and N. L. S. da Fonseca, "Mapping virtual networks onto substrate networks," *Journal of Internet Services and Applications*, vol. 4, no. 1, pp. 1–15, 2013.

[15] A. Fischer, R. Kühn, W. Mandarawi, and H. De Meer, "Modeling security requirements for VNE algorithms," in *Proceedings of the 10th EAI International Conference on Performance Evaluation Methodologies and Tools, ValueTools 2016*, pp. 149–154, Italy, October 2016.

[16] M. Sourour, B. Adel, and A. Tarek, "Collaboration between security devices toward improving network defense," in *Proceedings of the 7th IEEE/ACIS International Conference on Computer and Information Science, IEEE/ACIS ICIS 2008*, pp. 13–18, USA, May 2008.

[17] C.-J. Chung, P. Khatkar, T. Xing, J. Lee, and D. Huang, "NICE: Network intrusion detection and countermeasure selection in virtual network systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 4, pp. 198–211, 2013.

[18] V. Wilson and A. G. Krishnan, "Improving security in a virtual network by using attribute based encryption algorithm," in *Proceedings of the 2016 IEEE International Conference on Circuit, Power and Computing Technologies, (ICCPCT '16)*, India, March 2016.

[19] S. Shin, H. Wang, and G. Gu, "A first step toward network security virtualization: from concept to prototype," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2236–2249, 2015.

[20] P. Massonet, L. Deru, A. Achour et al., "Security in Lightweight Network Function Virtualisation for Federated Cloud and IoT," in *Proceedings of the 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*, pp. 148–154, Prague, Czech Republic, August 2017.

[21] D. Dwiardhika and T. Tachibana, "Virtual Network Embedding Based on Security Level with VNF Placement," in *Proceedings of the 2017 Society Conference of Electronic Information Communication*, pp. 7–27, September 2017.

[22] D. Dwiardhika and T. Tachibana, "Cost Efficient VNF Placement with Optimization Problem for Security-Aware Virtual Networks," in *Proceedings of the 2018 IEEE 7th International Conference on Cloud Networking (CloudNet)*, pp. 1–3, Tokyo, Japan, October 2018.

[23] Z. Kotulski, T. W. Nowak, M. Sepczuk, and M. A. Tunia, "Graph-based quantitative description of networks' slices isolation," in *Proceedings of the 2018 Federated Conference on Computer Science and Information Systems*, pp. 369–379, Poland, September 2018.

[24] J. Zhang, Y. Ji, M. Song et al., "Dynamic virtual network embedding over multilayer optical networks," *Journal of Optical Communications and Networking*, vol. 7, no. 9, pp. 918–927, 2015.

[25] A. Gonzalez, E. Barra, A. Beghelli, and A. Leiva, "A sub-graph mapping-based algorithm for virtual network allocation over flexible grid networks," in *Proceedings of the 2015 17th International Conference on Transparent Optical Networks (ICTON)*, pp. 1–4, Budapest, Hungary, July 2015.