Computer Networks 104 (2016) 55–65

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Towards a secured network virtualization

Yang Wang [a,*], Phanvu Chau [a], Fuyu Chen [b]

[a] Department of Math and Computer Science, La Salle University, PA, 19141, United States
[b] Department of Electrical Engineering, SUNY, University at Buffalo, NY, 14260, United States

ARTICLE INFO

Article history:
Received 18 September 2015
Revised 10 March 2016
Accepted 29 April 2016
Available online 6 May 2016

Keywords:
SVNE
Security-awareness
Network Virtualization

ABSTRACT

Network virtualization promises to fulfill the demand for an agile Internet that is friendly to technological innovation. In the past, tremendous efforts have been dedicated to studying the fundamental problem in network virtualization, namely Virtual Network Embedding (VNE). However, until recently, very limited work has addressed the security issues and implications of VNE or network virtualization as a whole, despite their importance. On one hand, the literature lacks a systematic overview of security issues in network virtualization (e.g., which can be VNE-relevant or VNE-irrelevant). On the other hand, existing studies on security-aware VNE share common limitations. This paper aims to present a timely study to fill the above needs with the following contributions: First, we present a classified comprehensive overview of security issues that arise in the context of network virtualization based on multiple criteria. Second, based on the review of existing approaches in Security-aware Virtual Network Embedding (SVNE), a novel framework is presented to address VNE-relevant security issues in network virtualization. Third, our extensive evaluation uncovers a few important implications and shows that the proposed framework can address the SVNE problem with reduced time (compared to that of the regular VNE approach).

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Network virtualization introduces a logical layer of abstraction to allow for a flexible and agile deployment of revolutionary technologies, which is considered a promising solution to address the Internet impasse [1–9]. In network virtualization, service provision is decoupled into two phases: logical representation, in which a service is expressed as a virtual network; and physical mapping, which instantiates each virtual network. These two phases are separately maintained by the Service Provider (SP) and the Infrastructure Provider (InP), respectively. Given the freedom of independent technology decisions by both parties, the inherence resistance (of the current Internet) to technological revolutions is removed in network virtualization. This separation, however, calls for a holistic bridging process that projects the logical network onto the physical network, which is referred to as the *Virtual Network Embedding* (VNE) problem.

Fundamentally, network virtualization builds upon node virtualization (e.g., Xen [10]) in combination with link virtualization (e.g., OpenFlow [11]). As a result, the VNE process contains two corresponding modules: node assignment, and link mapping, respectively. The former decides the physical host for each virtual node (by the creation of virtual machine instance), while the latter allocates bandwidth along substrate paths to connect instances of virtual nodes. Given the NP-Completeness of the VNE problem [12], existing approaches can be classified into three categories: *(i)* Optimal solutions based on Integer Linear Programming (ILP) formulations (e.g., link-based ILP model in [13] and path-based model in [6]); *(ii)* Relaxation approaches based on the LP-relaxation of the ILP formulations (e.g., relaxation and rounding in [13], and decomposition approach in [14]); and *(iii)* Heuristic or meta-heuristic algorithms (e.g., [15]).

Given the essence of *virtualization*, outsourcing computation, storage, content, and network to the third party (i.e., InPs) gives rise to inherent Confidentiality, Integrity and Availability (C.I.A) vulnerabilities [16,17]. As a result, security plays a critical rule in network virtualization. Despite the extensive studies in network virtualization (particulary in VNE), only limited work, however, has addressed the resulting security issues and implications [7–9,18,19]. On one hand, the existing literature lacks a systematic overview of security issues in network virtualization. Note that although it is commonly believed that security factors should be integrated into the VNE process to ensure a robust mapping [1], not all of them should or can be addressed in the VNE process. On the other hand, as to be further discussed, existing studies on security-aware VNE share a few limitations. First, they only address particular aspects of security, e.g., support for data integrity with encryption in [7]. Second, the security requirements (e.g., the

* Corresponding author.
E-mail address: wang@lasalle.edu (Y. Wang).

http://dx.doi.org/10.1016/j.comnet.2016.04.023
1389-1286/© 2016 Elsevier B.V. All rights reserved.

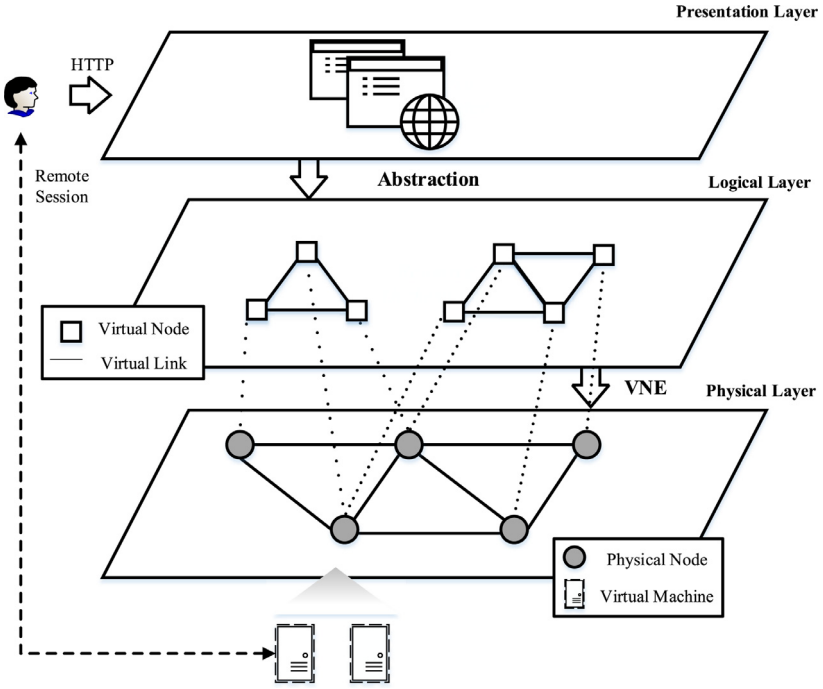


Fig. 1. Three-layer architecture of network virtualization.

security level [8]) are not explicitly defined in an *Open Design* [20] format that can be interpreted by SPs/end users and implemented by the InPs. In this work, we aim to present a timely study to fill the above gaps. We present a classified overview of security issues that arise in the context of network virtualization according to various criteria and hierarchies. Based on the discussion of existing approaches in Security-aware Virtual Network Embedding (SVNE), we propose a new SVNE approach that employs flexible and fine-granular security plans.

The remainder of this work is organized as follows. In Section 2, we overview security issues in network virtualization, which are classified according to multiple criteria such as their VNE-relevance. In Section 3, we define the SVNE problem, and present security plans that can be incorporated in the SVNE problem. In Section 4, we present a security framework that addresses the VNE-relevant security issues by resolving the SVNE problem. Section 5 presents the evaluation of the proposed framework. We discuss other security issues in Section 6, and conclude this work in Section 7.

2. A classified overview of security issues in network virtualization

In this section, we present a classified overview of security issues in network virtualization depending on various criteria including: holistic view from the layer perspective, end users's view, InPs' view, and VNE-relevance.

2.1. Layer perspective: holistic view

In network virtualization, the traditional ISP is decomposed into two new parties: the Service Provider and the Infrastructure Provider. The former offers end-to-end logical services to the end user, while the latter physically deploys and manages the substrate network infrastructure. From the security perspective, we view network virtualization as a three layer architecture as shown in Fig. 1. The *Presentation Layer* provides interfaces (e.g, Web, RESTful API) to end users in which the service feature can be specified. Each service request is abstracted as a virtual network at the *Logical Layer* that consists of virtual nodes and virtual links. Finally, each virtual network is instantiated at the substrate network of the *Physical Layer* via the VNE process.

Depending on the layer where the attack is originated, we can classify the attacks in network virtualization as in Table 1. At the physical layer, when virtual instances co-reside at the same physical host, the shared hypervisor or hardware can be exploited to construct cross-virtual-machine attacks [17,21]. Likewise, classic physical attacks and Denial of Service attacks can also lead to service disruption of the hosted instances [23–25]. At the logical layer, attackers can masquerade as a SP to probe the topology and determine the locations as well as attributes of possible victim instances in the physical network [18]. At the presentation layer, a web-based interface can be exploited with classic SQL injection, and Cross-Site Scripting attacks [17,22]. It is worth noting that the work flow (e.g., VNE process), control flow (e.g., remote access session), and data flow (e.g., computation results) among layers could be hijacked by attacks such as *man-in-the-middle*. In addition, InPs may not deliver the agreed computing/bandwidth resources to the subscriber, which could be hard to verify in scenarios such as Big Data Computation [26].

2.2. Service goals: end users' view

We next look at the security issues from the end users' viewpoint by answering an important question: what do general security goals specifically entail in network virtualization? This leads to a classification shown in Table 2 based on the goals of C.I.A and Assurance. Different from classic network environment, network virtualization implies the outsourcing of data/computation to the third-party away from the end users. The *Confidentiality/Integrity* of user data/computation should prevent access/modification from un-authorized parties including the InPs, and SPs. The *Availability* and *Assurance* goals require the InPs/SPs to deliver the service in an uninterrupted manner, and with the agreed resources, respectively. Given the hosting vantage of SPs, it is challenging to achieve

Table 1
Layer-based classification.

Layer	Physical	Logical	Presentation	Cross Layers
Representative attacks	*a*. Co-resident exploit [17,21] *b*. Physical attack (e.g., optical fiber tapping/jamming) [23,24] *c*. Denial of Service (DOS) [25]	*d*. Topology discovery [18,21,25]	*e*. User interface exploit (e.g., SQL injection, cross-site scripting) [17,22]	*f*. Man-in-the-middle (e.g., impersonation, eavesdropping) [22] *g*. Assurance violation (e.g., computational integrity [26])

Table 2
Goal-based classification.

Goals	Confidentiality	Integrity	Availability	Assurance
Requirements	No un-authorized access (including the SPs and InPs) to user data/computation.	No un-authorized modification to user data/computation.	Uninterrupted service instance.	InPs/SPs commit the agreed service to the end user.
Examples of compromise	Attacks *a,e,f* (from Table 1)	Attacks *a,e,f* (from Table 1)	Attacks *b,c* (from Table 1)	Attack *g* (from Table 1)

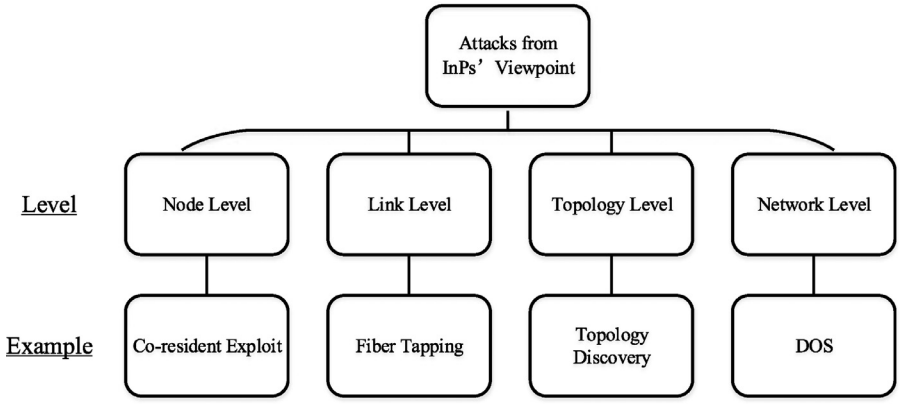


Fig. 2. Granularity-based classification.

the above goals in network virtualization as it is hard to verify that respective goal is not violated.

2.3. Granularity perspective: InPs' view

We further classify the security issues from the InPs' viewpoint depending on the granularity of the origins or targets of the attack, as summarized in Fig. 2.

For the ease of management, InPs generally maintain regularities and patterns in data center topologies (as practiced in many cloud solutions such as *Amazon EC2* [21]). For instance, co-resident virtual machines (VMs) typically share numerically close internal IP addresses, and small packet round-trip times. With sufficient external and internal network probing, the attacker can discover the locations as well as attributes of potential target hosts and/or virtual machines. This type of topology leakage is classified as an attack at the **Topology Level** in Fig. 2, which can serve as step stones for other attacks at the node and network levels. In the most simplified form, the attacker can probe the topology by iteratively sending virtual network requests. These requests act as means to obtain information about the physical network as the InP is expected to respond to each request with a binary answer that indicates whether the request can be accommodated based on available physical resources. Eventually, the attacker can deduce the topology of the InP at no cost after a given number of properly crafted requests [18].

With the leaked topology information, attackers can throw sufficient requests until nodes of their virtual network (i.e., VMs) are placed in the same network or even on the same physical host with the nodes of the targeted virtual network. The attacker can then proceed with attacks at the node or network levels as shown in Fig. 2.

At the node level, the attackers can employ *co-resident exploits* in two forms (Fig. 3). In the first form, the attacker can take advantage of vulnerabilities existing in the hypervisor or VM management software, which leads to penetration of the physical node and in turn the manipulation of any other co-resident targeted VMs. The recent exploit named **Venom** [27], for instance, can affect a wide spectrum of virtualization platforms including Xen [10], KVM [28], and the native QEMU client software [29]. Alternatively, the attacker can even exploit the target VM without penetrating the host via *cross-VM attacks*. The co-residency generally indicates the sharing of physical hardware, which can also serve as a common medium or side-channel between the VMs. Activity unique to the victim VM can be "listened to" and analyzed by the vicious VM via a side-channel in the form of a shared cache or memory bus. For instance, the vicious VM can observe the victim's load-based computational period via the memory access pattern, and uncover sensitive information of the target after sufficient recording.

At the link level, the attacker can employ classic approaches to sniff the traffic from/to victim VMs (e.g., in an Ethernet-based substrate network). The recent trend of employing optical transmission technology in data center networks has also opened doors to attacks on optical fibers. For instance, in *tapping* attacks, attackers can bend fibers to cause data alteration. A *jamming* attack allows the intruder to inject malicious signals, noise, or delays to the legitimate optical signals [23,24].

At the **Network** level, attackers can employ *Denial of Service (DOS)* attacks to overwhelm the target virtual network. One such exploitation relies on the under-provisioned nature of data centers

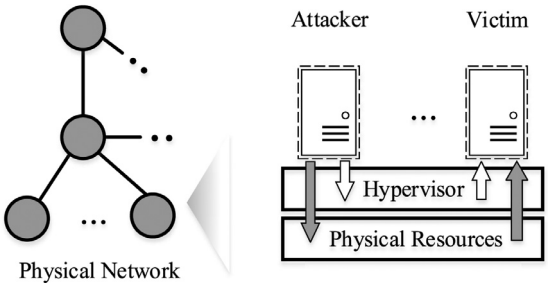


Fig. 3. Cross-VM attacks.

Table 3
Classification based on VNE-relevance.

Classification	Layer(s) Related	Examples
VNE-relevant	Virtual, physical	Attacks *a, c, d, f*
VNE-irrelevant	Presentation, physical	Attacks *b, e, g*

Table 4
Summary of recent work on SVNE.

Work	Main ideas	Limitations
[7]	Data encryption; network isolation	Non-practical due to coarse granularity of network isolation
[8]	Assign trust value to nodes, mapping based on the trust value	How to assign trust value is unknown
[9]	Assign security level to virtual/physical nodes/links, respects the level in mapping	Security level is abstracted without details
[30]	Assign security level to virtual and substrate nodes, respects the level in mapping; No co-residency with virtual nodes of lower security level	Security level is abstracted without details

[25]. For instance, when a group of vicious VMs purposely transmits at the maximum rate, e.g. 1 Gbps, then an under-provisioned up-link with a capacity of 1 Gbps can easily be saturated, thus leading to a denial of service or QoS degradation for other virtual networks that share the same up-link. Note that the source of the DOS attack resides in the same network with the target, thus it is hard to prevent (with firewalls or Intrusion Detection Systems).

2.4. Classification based on VNE relevance

The above classification has covered a variety of possible attacks in network virtualization. As the key enabler of network virtualization, it is commonly believed that security factors should be integrated into the VNE process [1]. However, we note that not all of the security issues can be addressed solely through the VNE process, which leads to the classification based on the VNE-relevance as shown in Table 3. From the layer viewpoint, it is obvious that security-aware VNE cannot address issues that are associated with the presentation layer. Also, given the limited knowledge and control of the physical network, the VNE process cannot address all the attacks at the physical layer such as optical fiber tapping/jamming [23,24].

A few recent work [7–9,30] has studied the security-aware VNE problem, which are summarized in Table 4. The studies in [9,30] both rely on the concept of *security level* where the mapping ensures that a virtual node is mapped to a substrate node with the same of higher security level. However, the concept of security level stays at the abstraction level and it is unknown how to determine in reality. The study in [8] adopts a similar concept called *trust value* to assit the mapping decision, which shares the same limitation. The study in [7] incorporates security into network virtualization by introducing the cryptography of the data communication, and the isolation of virtual networks. The latter implies the exclusive usage of the link and node resources for each virtual network. This isolation, however, is non-practical due to its coarse-granular (i.e., network level). It is worth noting that security schemes, different from other network designs, demand for two major features. First, it requests a comprehensive coverage of all security issues to avoid a *weak chain link*. Second, security requirements (e.g., the service level abstraction for VNE) should be explicitly defined in an *Open Design* [20] manner that can be interpreted by all the parties (e.g., end users, SPs, and InPs). Different from above aforementioned work, in Section 3, we systematically address above VNE-relevant security issues with fine-granular and clearly-defined security plans. We will also discuss the defense strategies for VNE-irrelevant issues in Section 4.

3. Security-aware virtual network embedding: addressing VNE-relevant security issues

Based on the overview in the last section, we present in this section our design to address the VNE-relevant security issues, which is incorporated into the security-aware virtual network embedding problem.

3.1. Security plan design

Before presenting the detailed plan, it is worth mentioning a few design principles that we follow. First, the security requirement should adopt an *Open Design* that can be clearly interpreted by all parties (end users, InPs and SPs). Second, the design should be friendly for end-user customization, and for InPs/SPs pricing. Third, while being *Open*, the design should keep a minimum exposure of information of the physical network.

Table 5 presents our design of the security plans to address the VNE-relevant security issues that were discussed in Section 2, which include three plans that customize the expected security measure of a virtual network request (VNR) at network, node, and link levels, respectively. Each level includes three policies that imply varied security measures.

The *Network* plan achieves the isolation of virtual node instances of VNRs at the network level of the physical network, which can be customized per virtual node of a VNR. When “High” security plan is selected, node instances of a VNR will reside in dedicated network as network co-residency is the major source of vulnerabilities for attacks such as DOS [25]. The “Medium” plan relaxes this restriction by allowing the network co-residency of nodes from VNRs of the same tenancy. The network plan can also mitigate sniffing after the compromise of other host within the same network of a VNR.

The *Node* plan achieves the isolation of virtual node instances of VNRs at the host level of the physical network, which is customized per virtual node of a VNR. The “High” security plan achieves physical host isolation among any VNRs, while the “Medium” security plan implements physical host isolation among tenancies. This node level isolation can effectively address co-residency exploit at the node level [17,21].

The *Link* plan achieves semantic security of the communication channel among virtual instances of a VNR, which can be customized per virtual link. End-to-End (E2E) policy achieves the semantic security via cryptography between two end hosting nodes of a virtual link in a *transport* mode, and requires the hosting nodes possess such cryptography capability. Point-to-Point (P2P) policy achieves *semantic security* in a hop by hop mode (similar to the IPSec *tunnel* mode [7]), which requires all the nodes along

Table 5
Security plans.

Plan name Granularity	Network Per-virtual-node			Node Per-virtual-node			Link Per-virtual-link		
Security policies	*High*	*Medium*	*None*	*High*	*Medium*	*None*	*P2P*	*E2E*	*None*
Policy details	No network sharing among any VNRs	No network sharing with other tenancies	No restrictions	No host co-residency among any VNRs	No host co-residency with other tenancies	No restriction	Cryptography at end *and* intermediate physical nodes	Cryptography at end physical nodes	No cryptography

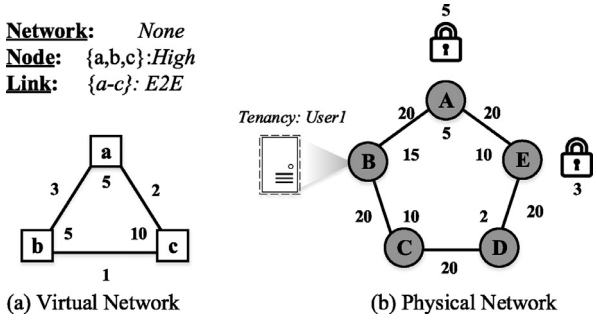


Fig. 4. Network models.

the mapped path of a virtual link to possess cryptology capability. The Link plan aims to mitigate the effects of *man-in-the-middle* attacks as encrypted data is rendered non-readable. Given the high volume of traffic transmission, note that such cryptography may demand purpose-built acceleration hardware.

With above security plans, one can support both tenancy level isolation, and network and host level isolation. In the next subsections, we present the network model and definition of the SVNE problem that incorporates above security plans.

3.2. Network model

We model a virtual network as an undirected weighted graph $G^V(N^V, L^V)$ in which N^V represents the set of virtual nodes, and L^V represents the set of virtual links. Each virtual node requests a security policy at the network and node plan, and each virtual link requests a security policy for the link plan. Also, each virtual node and link requests a certain amount of computing and bandwidth capacity, respectively. Fig. 4(a) shows an example of a virtual network request consisting of three virtual nodes, and three virtual links. The numbers adjacent to the nodes and links represent their computing capacity and bandwidth demand, respectively. For the virtual network request shown in Fig. 4(a), the security plan is customized as "None" at the network granular, "High" at the node granular (for all the virtual nodes), and "E2E" for virtual link *a–c* at the link granular. Note that since "E2E" plan is selected, the hosting node of *a* and *c* needs to possess the cryptography capability. If a "P2P" plan is selected for *a–c*, all the traversing nodes on the mapping path of *a–c* need to possess the cryptography capability.

The physical substrate network is modelled as an undirected weighted graph $G^S(N^S, L^S)$ in which N^S represents the set of physical nodes, and L^S represents the set of physical links. Each physical node has the following attributes: the available computing capacity, the cryptographic capability, network association [1], and the ownership of currently hosted virtual nodes. The cryptographic capability of a physical node is expressed as the total number of instances of cryptography that this node can support, where one instance corresponds to one passing physical path of a mapped virtual link. Each virtual link has a certain amount of available bandwidth resources. An example of a physical network can be found in Fig. 4(b). It contains five physical nodes connected by five physical links. The numbers adjacent to the physical nodes, and links represent the available computing capacity and bandwidth capacity, respectively. The lock and the associated number adjacent to a physical node indicates the deployment of cryptographic capability, and the maximum number of cryptography instances that can be supported by the node, respectively.

3.3. Problem definition and complexity

The SVNE problem is formally defined as follows:

Definition. SVNE is a decision problem that answers whether a virtual network request (VNR) can be mapped to the physical network while satisfying the following requirements: *(i) node mapping requirement:* one virtual node is mapped onto one physical node, with no two virtual nodes (from the same VNR) sharing the same physical node; *(ii) link mapping requirement:* one virtual link is mapped to one or more physical paths between two physical nodes which host the two incident virtual nodes of that virtual link; *(iii) capacity constraint:* for each virtual node and link, it is mapped to the corresponding physical resource that has a sufficient availability of computing capacity and bandwidth, respectively; *(iv) security plan constraint:* the VNR with the demanded security plan must be accordingly embedded as selected by the accompanying embedding policy.

Note that the SVNE problem is an NP-Complete problem as shown in Theorem 1. This can be proved by showing that a regular VNE problem can be reduced to the SVNE problem that has no specified security plan constraint in above definition. The detailed proof is skipped here.

Theorem 1. *The **SVNE** problem is NP-Complete.*

4. A framework for security-aware virtual network embedding

Based on the definition from last section, we present a framework to address SVNE with the goal of minimizing the overall cost. The framework consists of two major components: pre-processing, and SVNE mathematical modeling.

4.1. Pre-processing

The pre-processing aims to reduce the number of candidate physical nodes for each virtual node according to four filtering operations as follows. We refer to the set of candidate physical nodes of a virtual node I as C_I.

[1] We do not restrict the definition of the concept "network" to allow the implementation variances among SPs. One example is to treat "network" equivalently as a "subnet". A subnet level isolation generally can effectively mitigate the DOS originating from the same broadcast domain, and prevent packet sniffing due to the compromise of nodes from the same subnet.

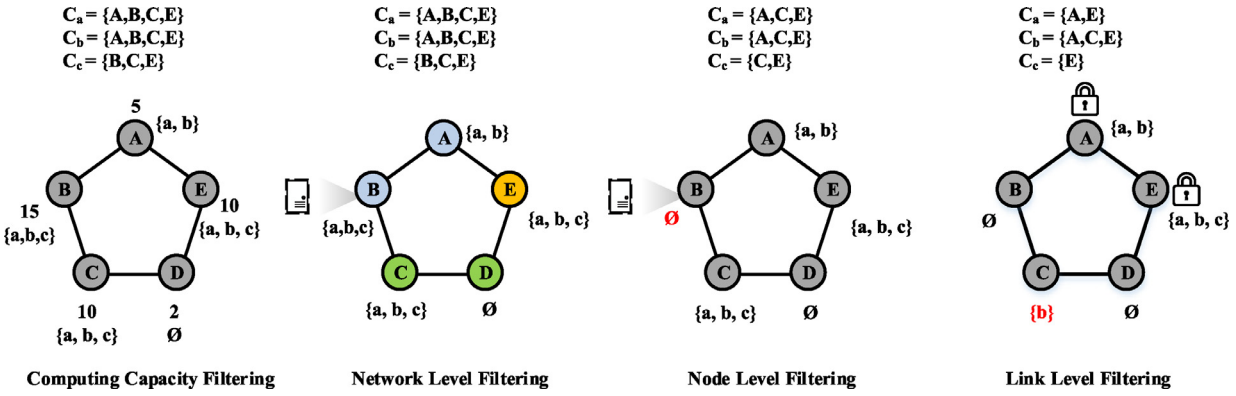


Fig. 5. Pre-processing.

(1) *Computing capacity filtering*. For each virtual node i, identify the candidate physical node set C_i by selecting the physical nodes that satisfy the computing capacity.
(2) *Network level filtering*. For each virtual node i, remove candidate nodes that are incompatible with the network security plan from C_i. For instance, when the "High" plan is selected, candidate nodes located in a network with virtual instances from different tenancies are removed.
(3) *Node level filtering*. For each virtual node i, remove the candidate nodes that are incompatible with the node security plan from C_i. For instance, when "High" plan is selected, candidate nodes that host virtual instances from different tenancies are removed.
(4) *Link level filtering*. For each virtual node i, remove the candidate nodes that are incompatible with the link security plan from C_i. Specifically, if a virtual node has any incident link with semantic security requirement, the candidate node(s) must have such capability.

The final purpose of the pre-processing is to construct an auxiliary graph as follows: starting from the physical network, augment it by connecting each virtual node to each physical node in C_i via an auxiliary link. We use an example to illustrate above process in Fig. 5 based on the virtual network and physical network in Fig. 4. In Fig. 5, the set C_a, C_b, C_c as well as the possible hosted nodes for each physical node are listed for each step. After the computing capacity filtering, for instance, physical node B (with capacity of 15) can accommodate any one of the three virtual nodes. As the network level policy is "None" (for all virtual nodes), the previous filtering results are maintained in network level filtering where the network association is color-coded in Fig. 5. In node level filtering, as we disallow the sharing of physical nodes (i.e., with the node level's "High" plan), physical node B is removed as a candidate for all the virtual nodes. In link level filtering, given the semantic security requirement of virtual link a–c, we remove physical node C from the candidates of a and c. Finally, we construct the auxiliary graph by connecting each virtual node to respective remaining candidate physical nodes as shown in Fig. 6. Note that with the auxiliary graph, the original SVNE problem is reduced to a *multi-commodity flow* problem with one commodity per virtual link [31].

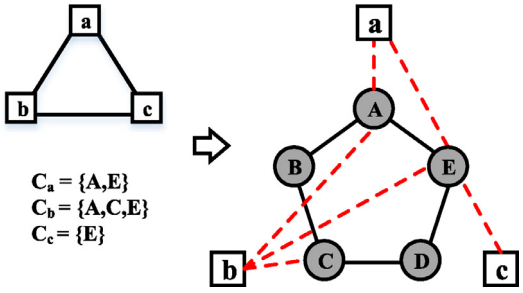


Fig. 6. An example of auxiliary graph.

Table 6
Notations.

ct_p:	The unit cost of flow on path p, $ct_p \geq 0$;
H_p:	The hop number of path p, $H_p \geq 1$;
cn_s:	The unit cost of computing resources at physical node s, $cn_s \geq 0$;
ψ	The set of all the paths for the model;
ψ^c	The set of paths for virtual link c;
ψ_s^c	The set of paths for virtual link c that consists of physical nodes all with cryptography capability;
cr_V	The computing capacity requirement of each virtual node V;
βr_c	The bandwidth requirement of virtual link c;
βc_x:	The bandwidth capacity of physical link x;
nc_i:	The number of cryptography instances that physical node i can support;
$\alpha_{p,(V,s)}$	1 if auxiliary link (V, s) is on the path p, and 0 otherwise;
$\theta_{p,i}$	1 if node i is contained in path p, and 0 otherwise;
ct_L:	The cost of selected link security plan;
ct_L^c:	The cost of selected link security plan for virtual link c;
ct_{ED}:	The cost per cryptography instance;
AL:	The set of auxiliary links;
D_V:	The node degree of virtual node V in the virtual network;
T_V:	The summation of the bandwidth requirements of all incident virtual links of virtual node V.

Table 7
Variables.

$E_{V,s}$:	1 if virtual node V is embedded to physical node s, and 0 otherwise;
sf_p:	The size of the flow on path p, $sf_p \geq 0$;
x_p:	1 if there is a non-zero flow on path p, 0 otherwise.

4.2. Mathematical models for SVNE

Based on the auxiliary graph constructed after pre-processing, we present path-based mathematical models to address the SVNE problem for two cases: the case that path splitting (SVNE-PS) is supported, and the case that no path splitting (SVNE-NPS) is allowed. These two cases adopt the same objective and node assignment constraints while differ in other constraints. The notations and variables of ILP models are listed in Tables 6, and 7, respectively.

Objective: The objective in Eq. (1) aims to minimize the overall cost of the physical node resources and link resources, and the cost of the link level security plan. Note that Eq. (1) does not include the cost of network or node level security plans as they are decided by the unit cost of the respective selected policy and the number of virtual nodes of a VNR that has employed the respective

policy. The ct_L depends on the security policy that is employed for each virtual link i.e. $\sum_{c\in L^V} ct_L^c$. When *E2E* plan is in place for a virtual link, say c, then $ct_L^c = \sum_{p\in\psi_s} 2 * x_p * ct_{ED}$ as only the two nodes that host the two respective incident virtual nodes of the link need the cryptography instances. When *P2P* plan is in place for a virtual link, say c, then $ct_L^c = \sum_{p\in\psi_s^c} (H_p + 1) * x_p * ct_{ED}$ as all the substrate nodes along the mapped path (p) require cryptography instances:

$$Min \sum_{p\in\psi} sf_p * ct_p + \sum_{(V,s)\in AL} cn_s * E_{V,s} * cr_V + ct_L \quad (1)$$

Node assignment: Eqs. (2) and (3) implement the node mapping. Eq. (2) guarantees that no more than one virtual node is mapped to the same physical node, while Eq. (3) ensures that each virtual node is embedded to exactly one physical node:

$$\sum_{V:(V,s)\in AL} E_{V,s} \le 1, \quad \forall s \in N^S \quad (2)$$

$$\sum_{s:(V,s)\in AL} E_{V,s} = 1, \quad \forall V \in N^V \quad (3)$$

Demand and Capacity (for SVNE-PS): Eqs. (4) to (8) are associated with SVNE that allows path splitting. Eq. (4) ensures that the amount of resources allocated at a physical link does not exceed its bandwidth capacity. Eq. (5) ensures that the aggregated flow over all of the paths of the given virtual link is equal to the demand size of the virtual link. Note that for the case of Point-to-Point encryption (i.e., "P2P" policy for link level) for a virtual link c, we use Eq. (6) instead of Eq. (5):

$$\sum_{p:x\in p} sf_p \le \beta c_x, \quad \forall x \in L^S \quad (4)$$

$$\sum_{p\in\psi^c} sf_p = \beta r_c, \quad \forall c \in L^V \quad (5)$$

$$\sum_{p\in\psi_s^c} sf_p = \beta r_c, \quad \forall c \in L^V \quad (6)$$

In Eq. (7), we ensure that only when an auxiliary link *(V, s)* is active, i.e., when $E_{V,s} = 1$, a non-zero flow can pass paths that include this auxiliary link. If V is not mapped to s, i.e., $E_{V,s} = 0$, the right term produces a 0 and thus no path can be established via that auxiliary link. Eq. (8) ensures the binary variable x_p equal to 1 when a non-zero flow presents at path p:

$$\sum_{p\in\psi} \alpha_{p,(V,s)} * sf_p \le E_{V,s} * T_V, \quad \forall (V,s) \in AL, V \in N^V, s \in N_S \quad (7)$$

$$sf_p \le x_p * \beta r_c \quad \forall p \in \psi_s^c, c \in L^V \quad (8)$$

Demand and capacity (for SVNE-NPS): Eqs. (9) to (12) are associated with SVNE that disallows path splitting. Eq. (9) ensures that the amount of resources allocated at a physical link does not exceed its bandwidth capacity. Eq. (10) ensures that exact one path is selected per virtual link. Similarly, for the case of Point-to-Point encryption for a virtual link c, we use Eq. (11) instead of Eq. (10):

$$\sum_{p:x\in p,\, p\in\psi^c} x_p * \beta r_c \le \beta c_x, \quad \forall x \in L^S \quad (9)$$

$$\sum_{p\in\psi^c} x_p = 1 \quad \forall c \in L^V \quad (10)$$

$$\sum_{p\in\psi_s^c} x_p = 1 \quad \forall c \in L^V \quad (11)$$

Eq. (12) ensures that a path can be selected only when included auxiliary links are activated, i.e., when $E_{V,s} = 1$:

$$\sum_{p\in\psi} \alpha_{p,(V,s)} * x_p \le E_{V,s} * D_V \quad \forall (V,s) \in AL, V \in N^V, s \in N^S \quad (12)$$

Finally, both the SVNE-NPS and SVNE-PS models need to employ Eq. (13) to ensure that the number of cryptography instances on a physical node is bounded:

$$\sum_{p\in\psi} x_p * \theta_{p,i} \le nc_i \quad \forall i \in N^S \quad (13)$$

5. Evaluation and analysis

In this section, we present our performance study of the proposed framework. The mathematical models are implemented in CPLEX [32] with the path set limited to the k-shortest paths of all commodities (i.e., virtual links) of the auxiliary graph. The simulation setup is as follows.

Virtual network: Virtual network is randomly generated with node number within the range of [2, 10], and a connectivity of 0.5. Each virtual network request randomly selects security plan at the network, node, and link levels, respectively, with the choice uniformly distributed over the three policies. The link and computing capacity is randomly generated within (0, 10]. Each VNR has an associated tenancy attribute, represented by an integer value uniformly selected from [1, 5]. VNR arrives in a Poisson process with an average rate of 4 VNRs per 100 time units, and each one has an exponentially distributed lifetime with an average of 1000 time units.

Physical network: The physical network is randomly generated with node number within the range of [10, 50], and a connectivity of 0.5. We use r to denote the ratio of physical nodes that are deployed the cryptography capability, each selected physical node can support a number of instances uniformly distributed within [10, 50] unless otherwise specified. The computing capacity and link capacity are uniformly distributed within (0, 50]. Each physical node is associated with a network attribute, represented by an integer value uniformly selected from [1, 10].

5.1. Does the security component bring extra computational overhead?

The first question that we want to look at is the computational time of the proposed framework comparing to a classic VNE ILP model without security-awareness. In this case, we evaluate the averaged time complexity of running a *single* instance of the given problem multiple times. Note that the proposed framework, compared to a regular VNE solution, contains two extra overheads: first, we introduce a pre-processing process for filtering; second, the mathematical model for SVNE is more complex than that of a regular VNE. Table 8 compares the number of variables and constraints for the proposed two models with a regular VNE model where V_{vne} and C_{vne} denotes the number of variables and constraints in a classic VNE without security-awareness [6], respectively. From Table 8, one can see SVNE ILP models have more variables and constraints than that of the VNE ILP model. Generally, we may expect that the added complexity (i.e., security) to the VNE problem will lead to increased computational time (in the SVNE problem).

Fig. 7 demonstrates the running time (in seconds) comparison among various approaches where the x-axis is the size of the auxiliary graph in terms of the number of nodes. The *PS* approach refers to the proposed model with path-splitting and the *NPS* refers to the proposed model without path-splitting. *VNE* refers to the ILP model without security-awareness [6]. The parameter k refers to

Table 8
Complexity comparison of ILP models of VNE and SVNE.

ILP	VNE	SVNE-PS	SVNE-NPS
Number of variables	$V_{vne} = \|N^V\| * \|N^S\| + \|P\|$	$V_{vne} + \|P\|$	V_{vne}
Number of constraints	$C_{vne} = \|N^V\| + \|N^S\| + \|L^V\| + \|L^S\| + \|N^V\| * \|N^S\|$	$C_{vne} + \|N^S\| + \|P\|$	$C_{vne} + \|N^S\|$

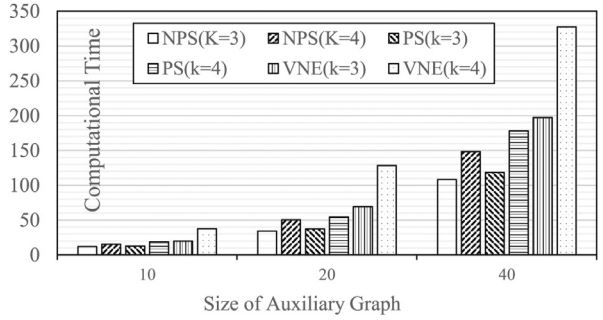


Fig. 7. Comparison of computational time.

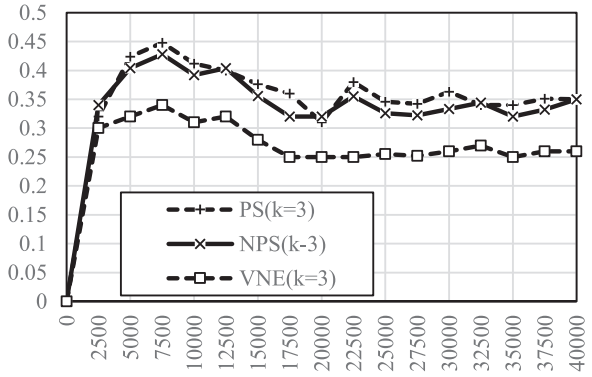


Fig. 8. Service blocking when $k = 3$.

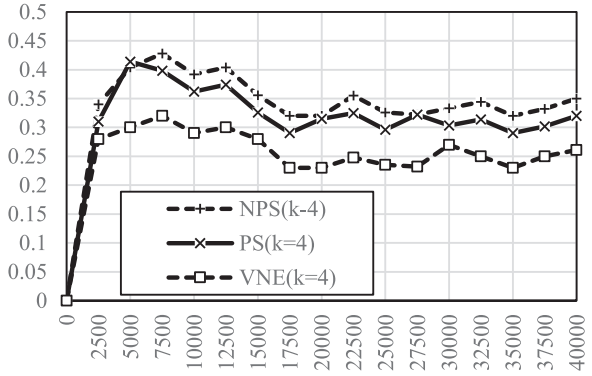


Fig. 9. Service blocking when $k = 4$.

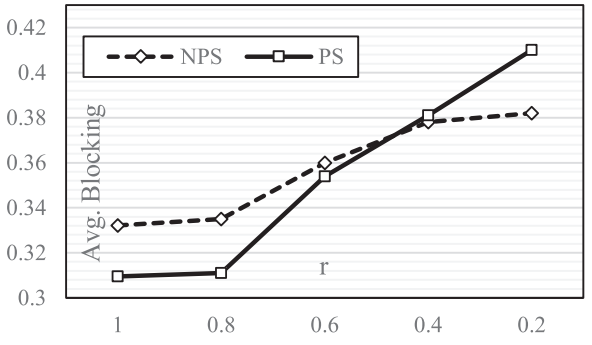


Fig. 10. The impact of parameter r.

the fact that we limit the path set to be the k-shortest paths for each commodity (i.e., virtual link) over the auxiliary graph. With the proposed framework, surprisingly, SVNE schemes consume less time than that of general VNE schemes. This time decrease is made possible by the pre-processing process. With a small polynomial time computation overhead in the pre-processing, the solution space (i.e., node assignment and path space) can be greatly reduced, which leads to a reduced problem size and hence the decreased computational time of the mathematical model. Moreover, the *PS* consumes slightly more time than the *NPS* scheme since the former has more variables and constraints than the latter. In practice, the computation overhead can be mitigated in two ways. First, the actual computation can be outsourced to high performance servers in the cloud. Second, the virtual network request can arrive in an advance reservation manner, thus leaving sufficient window for computation.

5.2. How does security requirement impact the service blocking?

The other important question to the InP is: with the added security components, what is the impact to the service blocking probability? Compared to a classic VNE mapping, note that the security awareness can lead to two new types of blocking: first, the isolation at network or node granular blocks the VNR (even when sufficient computation, and bandwidth resources are available). Second, the lack of cryptography capability may block otherwise admissible VNRs.

We first look at the first type of blocking. To avoid a convoluted effect due to the second type of blocking, we employ $r = 1$ and allow each physical node to support unlimited instances of cryptography. The resulting comparison of the service blocking probability between VNE and SVNE-NPS and SVNE-PS is shown in Fig. 8 for $k = 3$ and Fig. 9 for $k = 4$. In both figures, the X-axis is the running time units of the simulation, and the Y-axis is the total service blocking ratio. One can observe that the general VNEs clearly have less service blocking compared to SVNEs (i.e., $\sim$ 20%) in both the cases of $k = 3$, and $k = 4$. Clearly, this is due to the fact that the added security requirements reduce the possible node assignment and link mapping solution space with network and node level isolation. Furthermore, in most cases, SVNE-PS scheme has a slightly lower blocking than SVNE-NPS scheme. This can be explained by the fact that allow path splitting can prevent the blocking due to the lack of a single path that satisfies resource requirements. Finally, it is obvious that increase k can lead to blocking reduction due to the increased path space for mapping (at the expense of increased computational time).

For the second type of blocking, we look at the impact of the parameter of r (i.e., the ratio of physical nodes that employ cryptography capacity). As shown in Fig. 10, the PS scheme achieves lower average blocking probability than NPS when r is sufficient large (e.g., $r \geq 0.8$). This matches our earlier observation and can be explained by the increased path spaces with path splitting. When r approaches smaller values, surprisingly, NPS can lead to lower blocking in our simulation. This interesting observation can be explained as follows. With path splitting, it does increase the

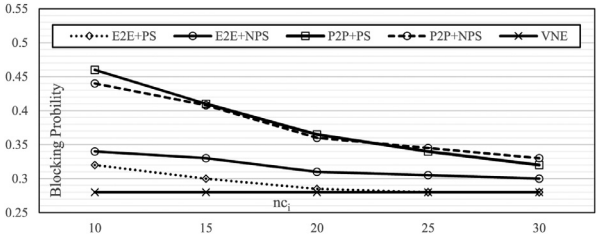


Fig. 11. The impact of parameter nc_i, PS/NPS strategies, and E2E/P2P plan.

chance of accepting requests that could be blocked in NPS due to the limited resource over a single path. However, splitting request over multiple paths, the resulted mapping leads to saturation of cryptography capacities (when P2P link level plan is selected), which in turn blocks more future traffic (especially when such capacity is limited, i.e., small r).

We further study the impact of the cryptography capacity, the mapping strategies (i.e., NPS and PS), and the link level security plan (i.e., P2P and E2E) in Fig. 11. We set r to be one, and support uniform number of cryptography capacity on all the substrate nodes (i.e., nc_i is the same for all $i \in N^S$). Also, we assume that the network level, and node level have no security requirements to accurately assess the impact of the *P2P* and *E2E* link level security plans. In Fig. 11, the $E2E + PS$ scheme refers to the case that all requests adopt E2E security plan and splitable mapping, and $E2E + NPS$ scheme refers to the case that all requests adopt *E2E* security plan and non-splitable mapping. Likewise, the $P2P + PS$ scheme refers to the case that all requests adopt P2P security plan and splitable mapping, and $P2P + NPS$ scheme refers to the case that all requests adopt P2P security plan and non-splitable mapping. There are a few major observations from Fig. 11. First, the major reason that $E2E + PS$ outperforms $E2E + NPS$ is due to the increased path search space for mapping. Note that when E2E plan is adopted, the required number of cryptography instances are the same for these two schemes. Second, comparing $P2P + PS$ with $P2P + NPS$, in contrast, the former outperforms the latter only when the nc_i value is large. This is consistent with the observation in Fig. 10 and can be explained by the same reason. Third, comparing the two schemes that adopt *P2P* plan with the two respective schemes that adopt *E2E* schemes, it is evident *P2P* schemes lead to higher blocking since it requires more cryptography instances. The $E2E + PS$ scheme, when nc_i is large, has a close performance with the regular VNE scheme.

Overall, above observations have a few implications in reality. First, to maintain the same level of service blocking ratio, the InP has to expect a CAPEX investment on security components, which hopefully will pay off in the long term. Second, the InP has to install an intelligent pricing scheme to drive the overall revenue increase.

6. Addressing other security issues in network virtualization

Security of network virtualization relies on securing all the components. In this section, we discuss other security issues based on the layer that may not be able to resolved based on the VNE process alone.

6.1. Addressing presentation layer security issues

The InPs that manage the presentation layer should be mainly responsible for the security of this layer. The SPs should prevent possible exploitations that can occur at the user interface via encryption including application layer secure protocol such as HTTPS.

6.2. Addressing logical layer security issues

The topology discovery [18] attack at the logical layer, while VNE-relevant, is associated with the mapping result instead of the VNE process itself. Recall that topology discovery attack relies on a binary answer of “Yes” or “No” to each virtual network request. Although we may not directly address this issue in the SVNE problem, one can remove this prerequisite by replacing the binary answer with a *price* for the quoted virtual network request instead of a binary answer. Further note that technically, the “No” answer can be removed by considering the cost for upgrading/activating/installing extra resources to accommodate the otherwise blocked request. On the one hand, an InP always maintains an evolving physical network that has to be dimensioned over time. On the other hand, with sufficient obscurity, the discovered topology becomes useless as the physical network will already be changed before the full disclosure.

6.3. Addressing physical layer security issues

First, the physical security of the physical resources should be the sole responsibilities of the InP. For instance, attackers can only tap or jam fiber links once they have obtained access to the physical resources [23].

Second, outsourcing data and computation to SPs implies inherent confidentiality and integrity issues [17]. The violation may be totally oblivious to the tenancy, especially when it is originated from the SPs (e.g., as an *insider attack*). This type of security issues necessitate assurance processes that need to be audited by the third party.

Third, the SVNE process can minimize *Host Exploit* when “High” or “Medium” security plans are in place. However, when such isolation is not selected by the user, to avoid possible exploit, the InP holds the main responsibility to ensure the security of the operating system, hypervisor, network security of the data center as well as the interfaces (e.g., remote control sessions) to other parties. Their efforts should include, for instance, hardening the host via firewall, and installing Intrusion Detection Systems.

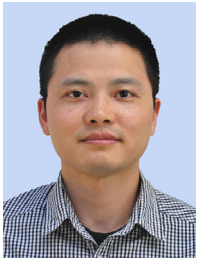
7. Conclusion and future work

This work has presented a classified overview on security issues in network virtualization based on four criteria: layer, security goal, granularity, and VNE-relevance. To address the VNE-relevant security issues, we have designed security plans to be incorporated into the resulting SVNE problem, which is further addressed with a framework employing preprocessing and mathematical models. Our overall solution adopts an open design principle and supports flexible security level customization. Likewise, we have discussed the possible solutions to other security issues that may not be directly addressed in the VNE process. In the future, we plan to develop a pricing model to ensure both the revenue increase of the InPs (to amortize the investment on security components) and obscurity level of the mapping results.

References

[1] A. Fischer, J.F. Botero, M.T. Beck, H. de Meer, X. Hesselbach, Virtual network embedding: a survey, IEEE Commun. Surv. Tut. 15 (4) (2013) 1888–1903.
[2] N.M.M.K. Chowdhury, R. Boutaba, Network virtualization: state of the art and research challenges, Commun. Mag. IEEE 47 (7) (2009) 20–26.
[3] T. Anderson, L. Peterson, S. Shenker, J. Turner, Overcoming the internet impasse through virtualization, Computer 38 (4) (2005) 34–41.
[4] Y. Wang, Q. Hu, X. Cao, Connectivity as a service: towards optical-based network virtualization, in: International Conference on Computing, Networking and Communciations'14, 2014, pp. 264–268.
[5] N. Feamster, L. Gao, J. Rexford, How to lease the internet in your spare time, ACM SIGCOMM Comput. Commun. Rev. 37 (1) (2007) 61–64.

[6] Q. Hu, Y. Wang, X. Cao, Resolve the virtual network embedding problem: a column generation approach, in: INFOCOM, 2013 Proceedings IEEE, 2013, pp. pp.410–414.
[7] L.R. Bays, R.R. Oliveira, L.S. Buriol, M.P. Barcellos, L.P. Gaspary, Security-aware optimal resource allocation for virtual network embedding, in: Proceedings of the 8th International Conference on Network and Service Management, 2012, pp. 378–384.
[8] C. Xing, J. Lan, Y. Hu, Virtual network with security guarantee embedding algorithms, J. Comput. 8 (11) (2013) 2782–2787.
[9] S. Liu, Z. Cai, H. Xu, M. Xu, Security-aware virtual network embedding, in: IEEE International Conference on Communications, 2014.
[10] Xen, http://www.xenproject.org/,
[11] OpenFlow, https://www.opennetworking.org//,
[12] D.G. Andersen, Theoretical approaches to node assignment, 2002, Unpublished Manuscript.
[13] N.K. Chowdhury, M.R. Rahman, R. Boutaba, Virtual network embedding with coordinated node and link mapping, in: Proceedings of IEEE INFOCOM'09, 2009, pp. 783–791.
[14] Q. Hu, Y. Wang, X. Cao, Virtual network embedding: an optimal decomposition approach, in: Proceedings of IEEE ICCCN, 2014, pp. 1–6.
[15] J. Lischka, H. Karl, A virtual network mapping algorithm based on subgraph isomorphism detection, in: Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures, 2009, pp. 81–88.
[16] C. Wang, Q. Wang, K. Ren, W. Lou, Ensuring data storage security in cloud computing, in: 17th International Workshop on Quality of Service'09, 2009, pp. 1–9.
[17] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, J. Molina, Controlling data in the cloud: outsourcing computation without outsourcing control, in: ACM Workshop on Cloud Computing Security'09, 2009, pp. 85–90.
[18] Y.-A. Pignolet, S. Schmid, G. Tredan, Adversarial vnet embeddings: a threat for ISPS? in: INFOCOM, 2013 Proceedings IEEE, 2013, pp. 415–419.
[19] P. Chau, Y. Wang, Security-awareness in network virtualization: a classified overview, in: Proceedings of 11th IEEE MASS: WSDIF Workshop, 2014.
[20] M.T. Goodrich, R. Tamassia, Introduction to Computer Security, Pearson, Boston, USA, 2011.
[21] T. Ristenpart, E. Tromer, H. Shacham, S. Savage, Hey, you, get off of my cloud! exploring information leakage in third-party compute clouds, in: S. Jha, A. Keromytis (Eds.), Proceedings of CCS 2009, ACM Press, 2009, pp. 199–212.
[22] E.L. Haletky, VMware vSphere and Virtual Infrastructure Security: Securing the Virtual Environment, Pearson Education, Inc., Boston, MA, 2009.
[23] N. Skorin-Kapov, J. Chen, L. Wosinska, A new approach to optical networks security: Attack-aware routing and wavelength assignment, IEEE/ACM Trans. Netw. 18 (3) (2010) 750–760.
[24] M. Furdek, Physical-layer attacks in optical wdm networks and attack-aware network planning
[25] H. Liu, A new form of dos attack in a cloud and its avoidance mechanism, in: Proceedings of the 2010 ACM Workshop on Cloud Computing Security Workshop, 2010, pp. 65–76.
[26] R. Liu, P. Mordohai, W.H. Wang, H. Xiong, Integrity verification of k-means clustering outsourced to infrastructure as a service (IAAS) providers, in: SIAM International Conference on Data Mining'13,, 2013, pp. 632–640.
[27] VENOM, http://arstechnica.com/security/2015/05/extremely-serious-virtual-machine-bug-threatens-cloud-providers-everywhere/,
[28] KVM, http://www.linux-kvm.org/,
[29] QEMU, http://wiki.qemu.org/,
[30] A. Fischer, H.D. Meer, Position paper: secure virtual network embedding, Praxis der Informationsverarbeitung und Kommunikation 34 (4) (2011) 190–193.
[31] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, Englewood Cliffs, NJ, 1993.
[32] CPLEX, http://www-01.ibm.com/software/.



Yang Wang received his M.S, and Ph.D. of computer science from Georgia State University in 2011, 2012, respectively. Prior to this position, he taught at Georgia State University since 2009, and received outstanding graduate teaching award in 2012. He also worked as a Senior Network Research Engineer at Futurewei Technologies from 2011 to 2012, and Senior Cloud/Integration Engineer at Internap Internet Infrastructure Service Corp from 2012 to 2013. His major interests are in the area of optical networking, network virtualization, network security, and pedagogy in CS/IT teaching, where he published over 30 papers in above areas. He is a reviewer for many international journals including *IEEE Transactions on Computers*, *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Journal on Selected Area in Communication*, and *IEEE Communication Letters*, and has been recognized as Exemplary Reviewer for IEEE Communication Letters in 2014. He serves on the Technical Committee for multiple international conferences including IEEE Globecom 2016, ICNC (2014, 2015, and 2016), IEEE Sarnoff Symposium 2015, and International Broadband and Photonics Conference 2015. Recently, he received two summer REU grants (2014, 2016) to study the security issues in network virtualization, and Amazon AWS Education Grant Award (2015-2016) to teach virtualization in the networking courses of La Salle.

Phanvu (Vu) Chau received his B.S. in information technology with distinguished honors from La Salle University. His primary areas of interest include security in network virtualization and cloud technologies. Vu currently holds the position of Developer Support Specialist at Google.

Fuyu Chen received the B.S. and M.S. degrees in mathematics from Nanjing University, Jiangsu, China, in 2003 and 2006, respectively. He received the M.A. degree in mathematics in 2008, and Ph.D. degree in electrical engineering in 2012, both from the State University of New York (SUNY) at Buffalo. His research interests are in the areas of cooperative communications, signal processing and networking, including the optimization and performance analysis of next generation wireless network. He is currently a software engineer at Brocade, working on Fabric Systems Routing Module Development.