Future Generation Computer Systems 67 (2017) 315–324

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Security challenges with network functions virtualization

Mahdi Daghmehchi Firoozjaei [a], Jaehoon (Paul) Jeong [b], Hoon Ko [a], Hyoungshick Kim [a,*]

[a] Department of Electrical and Computer Engineering, College of Information & Communication Engineering, Sungkyunkwan University, Republic of Korea
[b] Department of Interaction Science, College of Information & Communication Engineering, Sungkyunkwan University, Republic of Korea

HIGHLIGHTS

- We analyze potential security threats in NFV and categorize them.
- We suggest reasonable countermeasures to cope with those security threats.
- We present a secure virtualization environment to mitigate those threats in NFV.

ARTICLE INFO

Article history:
Received 29 February 2016
Received in revised form
14 June 2016
Accepted 14 July 2016
Available online 29 July 2016

Keywords:
Network functions virtualization
Network security
Virtualized network function
Security threats

ABSTRACT

The advent of network functions virtualization (NFV) has revolutionized numerous network-based applications due to its several benefits such as flexibility, manageability, scalability, and security. By the software-based virtualization of network functions on a single infrastructure, NFV provides users with a framework that dynamically provisions various network services in a flexible manner. However, NFV faces several security challenges (e.g., multi-tenancy and live migration) which make it vulnerable to some cybersecurity attacks (e.g., side-channel attacks and shared resource misuse attacks). In this paper, we provide an overview of NFV, discuss potentially serious security threats on NFV and introduce effective countermeasures to mitigate those threats. Finally, we suggest some practical solutions to provide a trustworthy platform for NFV.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Nowadays, the impressive effects of network functions virtualization (NFV) are evident in the wide range of applications from IP node implementations (e.g., future Internet architecture) to mobile core networks [1]. NFV allows network functions (e.g., packet forwarding and dropping) to be performed in virtual machines (VMs) in a cloud infrastructure rather than in dedicated devices [2]. NFV as an agile and automated network is desirable for network operators due to the ability of easily developing new services and the capabilities of self-management and network programmability via software-defined networking (SDN) [3]. Furthermore, co-existence with current networks and services leads to improve customer experience, and reduces the complexity, capital expenditure (CAPEX), and operational expenditure (OPEX).

In theory, virtualization broadly describes the separation of resources or requests for a service from the underlying physical delivery of that service [4]. In this view, NFV involves the implementation of network functions in software that can run on a range of hardware, which can be moved without the need for installation of new equipment. Therefore, all low-level physical network details are hidden and the users are provided with the dynamic configuration of network tasks [5].

Despite many advantages, NFV introduces new security challenges. Since all software-based virtual functions in NFV can be configured or controlled by an external entity (e.g., third-party provider or user), the whole network could be potentially compromised or destroyed. For example, in order to properly reduce hosts' heavy workloads, a hypervisor in NFV can dynamically try to achieve the load-balance of assigned loads for multiple VMs through a flexible and programmable networking layer which is known as virtual switch; however, if the hypervisor is compromised, all network functions can be disabled completely.

Also, NFV's attack surface is considerably increased, compared with traditional network systems. Besides network resources (e.g., routers, switches, etc.) in the traditional networks, virtualization environments, live migration, and multi-tenant common infrastructure could also be attacked in NFV. For example, an attacker can snare a dedicated virtualized network function (VNF) and then spread out its bots in a victim's whole network using

* Corresponding author.
E-mail address: hyoung@skku.edu (H. Kim).

http://dx.doi.org/10.1016/j.future.2016.07.002
0167-739X/© 2016 Elsevier B.V. All rights reserved.

the migration and multicast ability of NFV [6]. To make matters worse, the access to a common infrastructure for a multi-tenant network based on NFV inherently allows for other security risks due to the shared resources between VMs. For example, in a data center network (DCN), side-channels (e.g., cache-based side channel) attacks and/or operational interference could be introduced unless the shared resources between VMs is securely controlled with proper security policies. In practice, it is not easy to provide a complete isolation of VNFs in DCNs [7].

Although NFV is in the initial development stage, several previous studies have introduced NFV and its challenging issues. For example, Han et al. [8] discussed its challenges and opportunities in terms of the performance requirement and architecture. Veitch et al. [9] investigated the practical challenges of real-world NFV development for both performance optimization and diagnostic purposes. The relationship of NFV with SDN and cloud computing was presented in [10,11] which categorized the SDN hypervisors and compared their features of network attribute abstraction and isolation. Furthermore, the state-of-the-art in NFV and its standardization efforts were discussed in [12]. To assure the reliability of the common infrastructure of NFV, Cotroneo et al. discussed the possible reliability challenges of infrastructure in [13].

In this paper, we identify security risks in NFV and analyze their potential impacts from a variety of different angles. The main contributions of this paper are summarized as follows:

- We explore potential security threats in NFV and give the detailed list of the identified threats corresponding to our threats categories. Our categorization will be helpful for network operators and security engineers who look forward to deploying NFV-based services.
- We suggest reasonable countermeasures to cope with those security threats. Based on the recent technologies for trustworthy platform, we particularly present a secure virtualization environment to mitigate the security threats in NFV.

The rest paper is organized as follows. In Section 2, we briefly present an overview of NFV and its structure. In Section 3, we present the main security threats of NFV into two categories and discuss their corresponding solutions. In Section 4, we provide our recommendations to build a secure virtualization environment. Finally, we conclude the paper along with future work in Section 5.

2. Overview of NFV

In general, for deploying a new network service or platform, it is required to add some variety of hardware appliances which necessarily increase the cost of purchasing new network resources and hiring new engineers to manage those resources. However, rapid changes in technology have led to shorter product life cycles in the network industry [14]. NFV is a key enabling technology for avoiding substantial changes in the actual physical components of network systems by providing network functions through pure software implementation rather than hardware resources. In the virtualized environment, hardware can be emulated, and multiple virtual functions can share available resources and run simultaneously on an infrastructure through virtualization [15].

In NFV, traditional network appliances, which are mainly deployed as physical hardware components, can be virtualized and run on a common infrastructure [1,8], as shown in Fig. 1. That is, virtual appliances (e.g., firewall, WAN acceleration, VPN, router, content delivery network (CDN)) can be moved to, or instantiated in various locations in the network on demand, without the installation of new pieces of equipment.

Based on the framework introduced by the European Telecommunications Standards Institute (ETSI) [16], NFV is built on three main domains: (1) VNF, (2) NFV infrastructure, and (3) NFV management and orchestration (MANO). VNF can be considered as a container of network services provisioned by software, very similar to a VM operational model. The infrastructure part of NFV includes all physical resources (e.g., CPU, memory, and I/O) required for storage, computing and networking to prepare the execution of VNFs. The management of all virtualization-specific tasks in NFV framework is performed by NFV management and orchestration domain. For instance, this domain orchestrates and manages the lifecycle of resources and VNFs, and also controls the automatic remote installation of VNFs.

Since hardware and software for NFV are generally developed by different vendors, interoperability remains a major challenge to deploy NFV services. For example, MANO could be effectively implemented only if VNFs and network appliances can be accessed and managed through standard interfaces that hide as much of their heterogeneity in physical resources as possible. To provide open and standard interfaces toward the physical resources, as depicted in Fig. 1, NFV infrastructure includes a virtualization layer. This layer logically partitions physical resources and provides the anchor between VNF and the underlying layer of virtualized infrastructure [16]. Hypervisors are primary tools to implement this layer. A hypervisor provides a host with a virtualization environment that is functionally equivalent to the original machine environment [17]. Practically, the hypervisor monitors VMs' operations and manages the access to resources and provides failure recovery for required QoS [15]. In the view of security, the hypervisor should provide an isolated space for serving VMs and proper access control mechanisms to prevent unauthorized access to the shared resources between VMs. In practice, however, it is not easy secure isolation between them. We discuss this issue in Section 3.2.2.

3. Security threats

Theoretically, NFV is an ideal solution for deploying new network equipment and services because network functions can be dynamically updated via software downloads and updates instead of replacing physical hardware. However, some security and robustness issues still need to be addressed to fully attain the benefits of using NFV. We will particularly face two significant security challenges: (1) Network function-specific security issues and (2) Generic virtualization-related security issues [14], as shown in Fig. 2.

Since NFV is working on a network infrastructure, it is important to achieve the desired performance levels for enabling NFV. Unfortunately, most existing IP based networks are vulnerable to network traffic attacks such as distributed denial of service (DDoS). In theory, NFV controllers are potentially seen as a risk of single point of failure. The chance of such attacks is heavily dependent on the network topology and selection of NFV controllers [14].

Interestingly, virtualization can be an effective tool to prevent malicious network traffic by providing several useful capabilities such as isolation and live migration. For example, a centralized security management service can efficiently manage and control malicious network traffic within a pre-defined set of isolated VNFs; however, on the other hand, attackers can exploit the weakness of such virtualization technologies. In the next sections, we will discuss those issues in more detail.

3.1. Network function-specific threats

Network function-specific threats refer to attacks on network functions and/or resources (e.g., spoofing, sniffing and denial of service). Unsurprisingly, this type of threats is related to the attacker's capabilities and the target network topology.

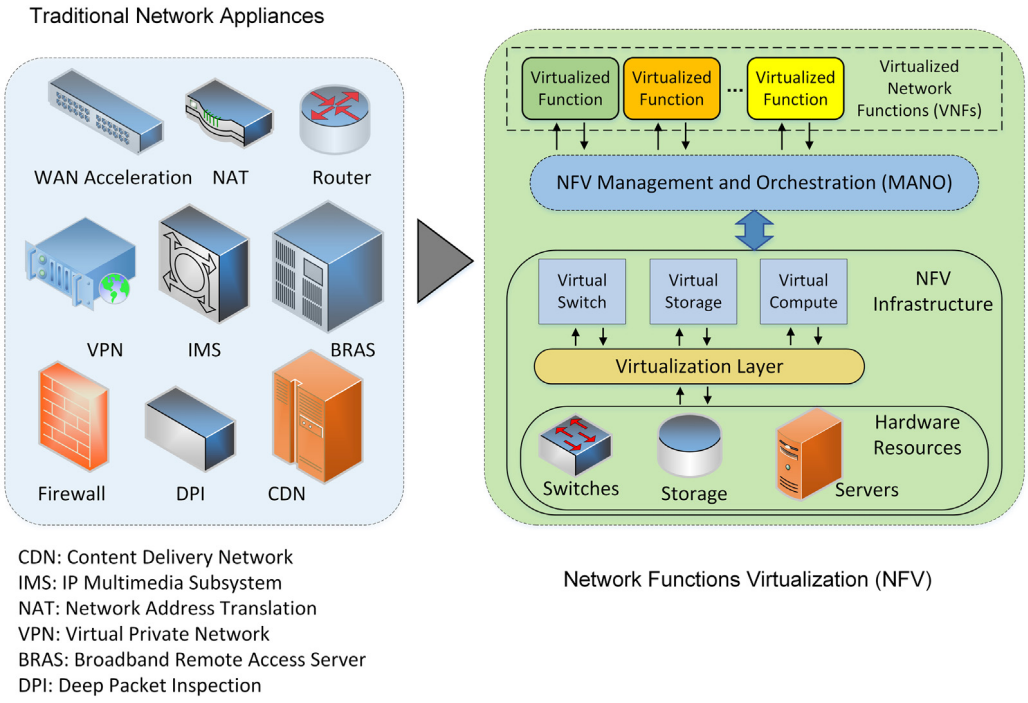

Fig. 1. The main concept of network functions virtualization (NFV).

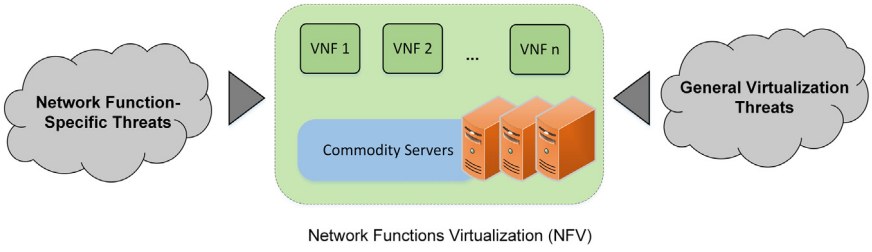

Fig. 2. Two significant security challenges on NFV.

In the private deployment of NFV where any attacker has no remote access to the network using NFV, the main network function-specific threats are only limited to the malicious insiders. In this case, sophisticated security protection solutions (e.g., firewall) focusing on outsider attacks are not effective anymore against an insider (e.g., malicious administrator) who has specific access rights to the infrastructure. In theory, identity and access management mechanisms (e.g., role-based access control) could be properly applied to mitigate the impact of insider attacks. Also, the analysis of access logs can detect suspicious insider attack activities [14]. Note that a configuration error can often expose a network function to the public Internet in practice [14]. Therefore, it is very hard to assume that there is no need to consider external attackers even when NFV is originally deployed in a private manner.

As illustrated in Fig. 3, in the public deployment of NFV, the network can be accessible by remote clients through public networks and/or by the third-party networks to control the specific VNFs. Due to the appearance of VNFs controlled by the entities in the third-party networks, NFV generally faces a wide range of network security challenges. If the remote clients or third-party network entities are malicious, the infrastructure can be disabled or compromised by using network attacks (e.g., DDoS) as well as software attacks that exploit specific vulnerabilities in the deployed virtualization software.

At the first glance, it seems that those network attacks can be simply prevented by using existing security solutions such as firewall and intrusion detection systems, but it is not easy to mitigate those attacks in some situations for NFV. For example, an attacker can create botnets running on the cloud service providing NFV [18]. In this case, since the attacker performs network attacks from inside the network via the botnets, the effectiveness of conventional security solutions may be limited [6]. Therefore, NFV providers should carefully consider mitigating sophisticated DDoS attacks such as Xmas-tree and TCP SYN flood from inside the network [18,19]. Furthermore, intelligent important data allocation [20] may be helpful to avoid unwilling access by allocating the appropriate parts to the decision-making groups at various management levels.

Finally, the applicability of software interfaces or APIs in cloud services, which are utilized by customers to interact with the cloud services, gets involved in NFV. Inherently, the security and availability of services offered by cloud nodes depends upon the security of these interfaces [21]. Based on this, the APIs in cloud services are open to considerable security issues on NFVs. Surely, the threats related to network functions of NFV are not new, but are still more investigated to build a secure NFV environment for service providers.

3.2. General virtualization threats

The foundation of NFV is set on network virtualization. In this NFV environment, a single physical infrastructure is logically shared by multiple VNFs. For these VNFs, providing a shared,

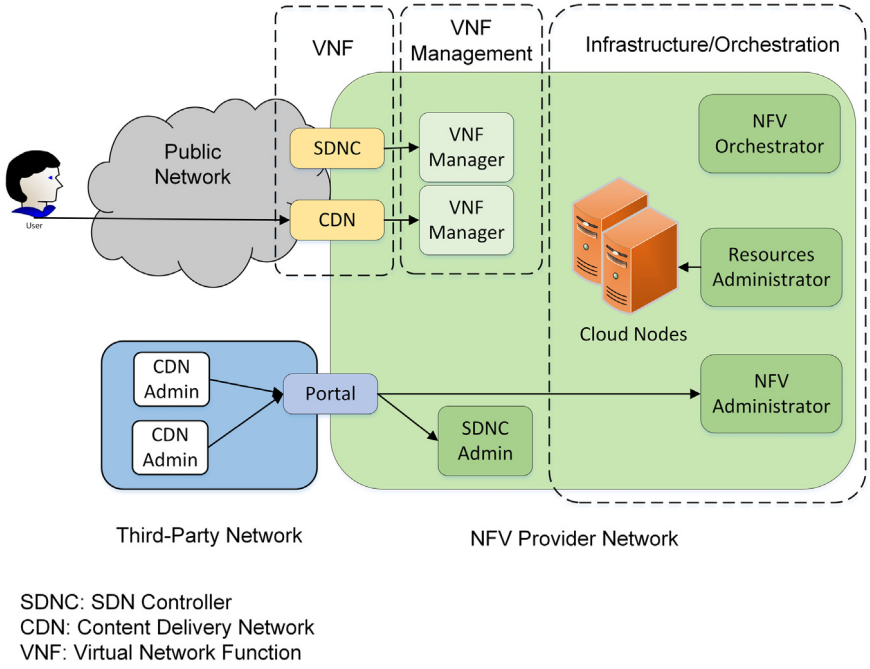

Fig. 3. Public deployment of NFV, accessible by clients through public network and third-party network. *Source:* Adapted from [14].

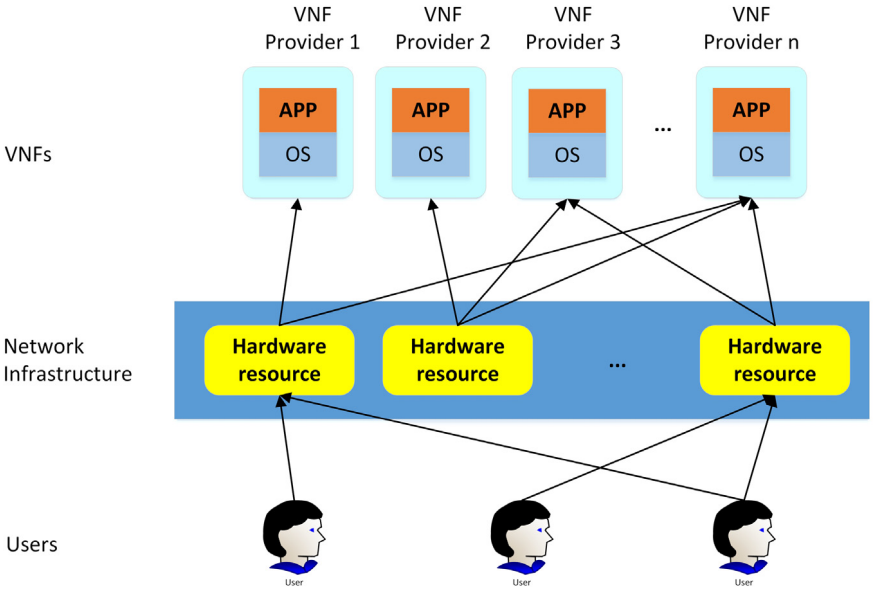

Fig. 4. Participating entities of NFV; User, network infrastructure, and VNF provider.

hosted network infrastructure introduces new security vulnerabilities. As illustrated in Fig. 4, the general platform of network virtualization consists of three entities; the providers of the network infrastructure, VNF providers, and users. Since the system consists of different operators, undoubtedly, their cooperation cannot be perfect and each entity may behave in a non-cooperative or greedy way to gain benefits [22].

The virtualization threats of NFV can be originated by each entity and may target the whole or part of the system. In this view, we need to consider the threats, such as side-channel or flooding attacks as common attacks, and hypervisor, malware injection or VM migration related attacks as the virtualization and cloud specific attacks. In the following subsections, we try to categorize these threats, according to their origination.

3.2.1. *Infrastructure-target threats*

Operational interference. Due to the common accessibility of the infrastructure, a malicious user or a compromised provider of VNF can interfere with the operations of the infrastructure by inserting malware or manipulating network traffic. The increasing network processing power and the programmability of the network interface cards (NICs) and exploiting new routers with the programmable packet processors [7,23] make it possible for a malicious entity to violate a network's operations. For instance, to achieve a scalable VM networking, Luo et al. [7] proposed to offload virtual switching from hosts to programmable NICs (PNICs). Although PNIC-based virtual switching improves the network performance and reduces the overhead and complexity by isolating computing and packet switching, it introduces vulnerabilities

threatening network infrastructure. Incorporating NICs with programmable packet processors in the network switching makes it feasible for an attacker to modify the operations of the packet processor by injecting the specially purposed data packets. In the proposed model of [7], to handle the traffic congestion of a VM, another virtual switch connected to the VM can buffer the packets inside itself and forward them to the destination. The code vulnerability of the hosted virtual switch allows for the chance for an attacker to trap the packets of a victim host or generate the packets which lead to network congestion or packet retransmission.

The same security threat can be considered for other programmable network interface nodes. Despite the benefits of FPGA- and software-based programmable routers or routers with programmable packet processors to improve the network performance and flexibility [23–26], it leads to new vulnerabilities of the infrastructure. To get some insight, SwitchBlade [27] can be mentioned as a FPGA programmable and customizable data plan for a virtual router. The DoS attack for this SwitchBlade is mostly predictable if an attacker takes the control of the operations of the packet processor of a router [22].

Cooperating chance for malicious provider. Having access to the resources of the network infrastructure, VNF providers can take participation in network operations. Network-as-a-Service (NaaS) [28] is a good example, which VNF providers use in order to support customized forwarding decisions based on each application's needs in the cloud computing. Although this model makes it possible to offer efficient in-network services, such as data aggregation, stream processing, caching, and redundancy elimination protocols, it lets VNF providers to take sabotage activities against their competitors or the network infrastructure. The opportunity to detect the vulnerabilities of the network infrastructure and launch a flooding attack (e.g., DoS) to bring down the entire infrastructure and moreover extract secret information [22], is conceivable for a malicious VNF provider in competition with a co-hosted VNF. By detecting excessive resource consumption by a virtual network, the hypervisor can prevent this problem from happening. As the cost for reasonable prevention of DoS, Shafieian et al. [6] suggest restarting a malicious virtual network instead of restarting the entire physical system.

Misuse of shared resources. Some cloud-specific attacks like resource freeing attacks (RFAs) [29] or fraudulent resource consumption attacks (FRC) [30] are possibly executable over NFVs. Generally, the misuse of the shared resources of the infrastructure, such that the victim can have no benefit of the shared or dedicated resources, is the basement of these threats. Creating dedicated instances for users and verifying malicious requests according to a blacklist of IP addresses can be mentioned as solutions for these threats [6].

3.2.2. VNF-target threats

Outsourcing challenges. To support the characteristic of NFV, it is possible to permit the hosting of different virtual functions in a common infrastructure or the cloud computing. Furthermore, NFV allows for outsourcing the core computing and software capabilities to the third-parties' networks. A significant security issue of NFV may be caused by releasing the resources of the cloud and transferring the workload to an off-device network in order to manage the workload potentially [31]. Offering security services in the cloud for mobile devices is a good example for this kind of threat. Oberheide et al. [31] propose a solution to avoid the complexity and intensiveness of security functions which are heavy in both computation and power for mobile devices. In this solution, the security functions are moved to an off-device network employing multiple virtualized malware detection engines. However, it allows for a good opportunity for an attacker in the shared network infrastructure to take the control of the services or to compromise the confidentiality.

Despite the benefits of the outsourcing, these conditions increase the probability of security problems related to virtual networks which are targeted by other entities of NFV, such as network infrastructure operator, co-hosted VNFs and users. Each entity, which is based on its relation to the virtual network, can create a risk on the function or secure information of the virtual networks. For instance, malware injection is a feasible attacks to eavesdrop or violate the functionality of the virtual networks. To prevent this threat, it is necessary to control and regulate the malicious entity not to make malware be public in NFV [6].

Side-channel attacks. Logical isolation, as a required property of NFV [15,22], improves the control and manageability of a shared infrastructure system. The isolation can be performed in different levels, such as address spaces or SDN virtualization system [5]. A variety of mechanisms (e.g., virtual LAN or programming isolated slices) provide a way to establish traffic, physical, and control isolation [32]. Relying only on the traditional access-control mechanisms to perform logical VMs' isolation is not sufficient for modern virtualization systems [33]. Based on this insufficiency, some cross virtual network side-channel attacks may threaten co-hosted VNFs in the shared infrastructure [22,34]. As a practical side-channel attack, a covert channel attack bypasses mandatory auditing and access controls to violate resource isolation. Wu et al. [35] showed the insufficiency of cache channel techniques in a virtualized environment due to addressing uncertainty, scheduling uncertainty, and cache physical limitations. Wu et al. also suggested a scheme of timing-based data transmission and bus locking mechanism to address these obstacles. As another example, Zhang et al. [33] showed an access-driven side-channel attack on a virtualized symmetric multiprocessing system, such that an attacker alternates execution to observe the behavior of a victim. In this attack, the attacker frequently alternates execution on the same core with the victim so that it can measure side-effects of the victim's execution.

In software environments, Osvik et al. in [36] described how a process in a side-channel attacks can threat other processes running on the same processor, despite partitioning methods such as memory protection, sandboxing and virtualization. As a key point for an attacker, it is important to know where the victim's access tables reside in the common resources (e.g., memory). In a public cloud infrastructure, checking the assigned IP address, round-trip time (RTT), and clock-skews can help to determine VM co-residency [37]. The placement vulnerabilities in the multi-tenancy infrastructure make an adversary be able to launch its VM co-resident with a victim VM on the same physical host. Varadarajan et al. in [38] investigated these vulnerabilities in three major infrastructure providers (e.g., Amazon EC2) and showed in spite of enhancement against past co-location verifying techniques, insufficient performance isolation in hardware still allows for such co-residency. Therefore, hiding the access management from NFV is important to make a secure environment for virtual networks.

Live migration. The live migration of VM is regarded as an advantageous feature of virtualization, because it makes it possible to relocate VMs without any service interruption in NFV [4,39,6]. The benefit of migration is significantly evident in the workload balancing and system management. However, it might be vulnerable under some threats, such as a replay attack, Man-in-the-Middle (MitM) attack risen by sniffing the traffic and DDoS flooding attack [4,22] if the protection for the migration is not carefully designed. Normally, the VM migration is performed by copying its memory pages from a source hypervisor to a destination hypervisor while a VM is running within the source hypervisor [39]. Initiating unauthorized migration to the attacker's network, which leads to taking control over a victim's VM or initiating the migration of a

large number of VMs to a victim's network in order to break down it are the possible results of these threats.

To prepare a safe environment for the live migration, some protection solutions rely on cryptographic methods. In this regard, the virtual trusted platform module (vTPM) can use TLS protocol to provide confidentiality and authentication [40]. By virtualization, all cryptographic operations of hardware-based TPM (will be discussed later), e.g., key generation, hashing and migration are implemented in software and are available to VMs. As another example, a security-preserving live migration protocol [41] can provide integrity and privacy protection during and after a live migration on a hypervisor-based process protection systems by encrypting and hashing. Basically, these solutions lead to the computational overhead of encryption which is not desirable to have an agile NFV. To avoid this computational overhead, solutions like live migration defense framework (LMDF) or Intel's trusted execution technology (Intel's TxT) define non cryptographic methods for safe migration. The LMDF tries to preserve confidentiality with no encryption by detecting a live migration in the early phase and also executing integrity and confidentiality measurements before the migration [42]. In the Intel's TxT solution, the live migration is only possible between some pools of trusted hosts, with Intel's TxT enabled and such that the platform launch integrity has been verified [43]. Despite avoiding the computational overhead, the mentioned solutions have their own limitations. For instance, creating and updating the trusted pools lead to some considerations in the Intel's TxT. Furthermore, all trusted hosts in the pool should be Intel's TxT enabled. This condition especially limits the live migration in the multi-vendor platforms.

Non-neutral infrastructure provider. Potentially, the provider of network infrastructure can make threats over its hosted VNFs. Due to the responsibility of the network infrastructure as the basement of NFV, it should prepare the resources and interconnections required by the hosted VNFs. To this end, the infrastructure provider has the authority to monitor a network's activities and intervene them (if needed) to control the congestion and accessibility of the network. Obviously, the malicious or non-neutrality behavior of the provider [22,34] can threat the operations or privileged information of virtual networks.

3.2.3. User targeted threats

User's privacy and confidentiality. A user as an end-point of the network is the easiest target to other malicious entities of NFV. The user's traffic is under monitoring and sniffing of a VNF provider for proper service quality (QoS). Offering virtualized network services, such as firewall, intrusion detection, DDoS detection, SSL gateway, and media cache [26], gives service providers a complete dominance over the user's information. It leads to a new trust relationship such that users must trust their VNF providers with respect to the privacy of the users' data and the integrity of the computations [34].

On the other hand, the privacy and confidentiality of the users are open to a network infrastructure provider. In order to control the access and network congestion, the network traffic is under monitoring of the infrastructure provider. Disturbing peer-to-peer (P2P) connections and sniffing protocol headers in the pretext of traffic shaping are examples relevant to this vulnerability [22]. The infrastructure can also introduce non-obvious threats to other users due to the subtlety of how physical resources can be transparently shared between VMs [34].

Malicious instances. Obviously, the user of NFV can be targeted by attacks originated by other malicious users who utilize the flaws of VNFs or the infrastructure. For instance, as cloud malware injection attack on Amazon EC2 public IaaS Cloud, a malicious user by editing the permission of the image of its VM (Amazon Machine Image called AMI) can make this image be public in the cloud. This malicious image will be visible to other users, so they can launch VM instances based on this image, which causes some threats such as the leakage of the victim users' information [6]. As a consequence, to provide a safe environment for users, it is necessary for the infrastructure to detect and prevent any malicious instance. For this purpose, it should not be possible for an attacker to determine where an instance is located or is co-located with its own instance in the infrastructure.

4. Secure virtualization environments

As stated earlier, security threats of NFV are categorized as network function-specific threats and virtualization-related threats. Generally, the first group of threats is limited to networking function issues and does not occur from the virtualization. On the other hand, as mentioned in Section 3.2, the virtualization threats of NFV include a wide range of issues from resource management to logical isolation and cloud computing. To resolve those issues, in the following parts, we suggest the recommended protection in regard to virtualization threats and depict the suggested platform for secure virtualization environments.

4.1. Security protection for virtualization environments

Trustworthy assurance. Cooperation among the entities of NFV requires a trust management to ensure the integrity provision. The privacy and confidentiality of a user's information should be respected by both providers of VNF and infrastructure. While the cryptographic techniques, as basic solutions (e.g., message stream encryption (MSE) [44]), are efficient to guarantee the confidentiality of the data, the key management and access control policies can be compromised when the hypervisor is accessible by any third-party. In this view, Intels single root I/O virtualization (SR-IOV) [45] technology with the enhancement of VANFC [46] can be a safe solution for a user's instance in the presence of untrusted providers.

In the view of a VNF provider, a neutral and trusted infrastructure provider is expected for a multi-tenant network. To this end, by traffic validation and monitoring mechanisms (e.g., DynaFL [47] for fault localization in the network with no requirements on path durability or the source node knowing the outgoing paths), the irregularity and malicious behaviors of the infrastructure provider should be detected. Furthermore, the accountability and trust management can be utilized by VNF provider to know whether its software is running without any modification in the infrastructure provider's network. In this case, dynamic trust management protocols, such as a protocol suggested by [48], make it feasible to provide a secure routing optimization in the presence of untrustworthy network. Remote attestation service (e.g., Intel's OAT [49]) and trusted network access control [50] are other feasible validation mechanisms for this goal.

Logical and physical isolation. In the view of network infrastructure, which includes physical resources, the complete and agile isolation of resources accessible for co-hosted VNFs is necessary. Despite logical and physical isolation in NFV, an adversary might penetrate the isolation between VNFs due to some vulnerabilities which escape from hypervisor [34] or allow for data breach [21]. To reduce the possibility of side-channel attacks, some preparations, such as hiding the access management, utilizing secure database interfaces, dedicating resource instances, and obfuscating service structures, are necessary. Essentially, it should not be possible for an attacker to detect other co-hosted VNFs or a user's instance in the same physical resource. As a practical solution, STEALTHMEM, which is introduced in [51], manages a set of locked cache lines per core and multiplexes them. In this model, each VM can load its own

Table 1
Security threats of network infrastructure and recommended solutions.

Security threats	Countermeasures
Incorporating Programmable devices: DoS, Network congestion	Over-encryption connection, Traffic validation: MSE [44], DynaFL [47], Intel's OAT [55]
Resources controlled by VNF provider: Flooding attacks, Information leakage	Detecting excessive resource consumption, Resource isolation: Intel's VT-d [56], SR-IOV [45], VANFC [46], and Remote attestation (e.g., Intel's OAT)
Misuse of shared resources (by user or VNF): Resource misuse attacks e.g., RFA and FRC attacks	Dedicated instance, Behavior monitoring, Recovery capability: Trusted Network Access Control [50], and Intel's OAT

sensitive data into the locked cache lines which are never evicted. To detect undesired co-residency, Zhang et al. [52] introduced HomeAlone, which allows a VNF provider to remotely verify that its VMs are physically isolated. To detect any unexpected activity, HomeAlone measures cash usage in a period of time while its VMs have no activity in a selected cache region. The virtualization technologies based on trusted platform module (TPM), such as Intel's VT-d and Intel's TxT technologies, which are discussed as practical models at the next section, prepare the conditions to protect against side-channel attacks.

Although an infrastructure provider should not interfere with the activity of a hosted VNF, monitoring a VNF's behavior, such as extensive resource consumption or instance controlling, is necessary to detect any malicious treatment like shared resource misuse which leads to FRC or RFAs attacks.

Secure outsourcing and live migration. Despite the tremendous benefits of the offload of network traffic and the task outsourcing in the cloud computing, it is compromising to transfer the sensitive information or key management to external networks. Thus, secure outsourcing services are in great need to not only protect sensitive information but also validate the integrity of the computation results. Although the over-encryption methods are effective to achieve secure multiparty computation (SMC), their computational overload should be considered. For instance, an oblivious transfer and a fully homomorphic encryption (FHE) [53] protocol, which performs computations on encrypted data without decrypting, are possible solutions. Melis et al. [54] investigated the effect of using cryptographic primitives, e.g., FHE and public-key encryption, in order to preserve the privacy of the outsourcing of network functions. These solutions need to use heavy cryptographic primitives, which result from tradeoffs between privacy and performance.

Evidently, live VM migration in a secure manner requires a secure interface with the authorized access for source and destination parties. Furthermore, a mechanism to detect and report any malicious activities during the migration is required. To address these considerations, the root of trust prepared by TPM, enables remote attestation by hashing the software components [40]. In this view, vTPM offered by the Intel's TxT solution, notwithstanding the mentioned limitations, is qualified. Moreover, the solutions such as security-preserving live migration protocol and LMDF might be considered to perform a secure live migration.

Tables 1–3 summarize the common virtualization threats of NFV's entities and recommend the techniques and solutions which address these threats.

4.2. Practical models

Basically, the secure isolation and introspection are the fundamental bases for NFV environment. To achieve a secure environment, secure computing platforms based on the trusted computing

Table 2
Security threats of VNF provider and solutions.

Security threats	Countermeasures
Task outsourcing: Malware injection, Eavesdropping, Functional violation, Confidentiality compromising	Secure outsourcing services, Over-encryption connection, Integrity validation: TPM [57], Intel's TxT [43], and Remote Attestation (e.g., Intel's OAT)
Multi-tenancy: Side-channel attacks between co-hosted VNFs	Hiding the access management, Secure database interfaces, Dedicating resource instance, Obfuscating the service structure: Intel's VT-d, OAT, TxT, SGX [58], STEALTHMEM [51], HomeAlone [52], VANFC
Live VM migration: MitM attack, Traffic sniffing, Stackover flow, DDoS attack	Source and destination authentication, Authorized access to interface, Detecting malicious activity: Security-Preserving Live Migration [41], LMDF [42], and vTPM [40]
Compromise infrastructure provider: Interfering the function of the hosted VNF, Violate the operation, Threatening privilege information	Monitoring mechanism to detect anomalous behavior, Traffic validation techniques: TPM, Intel's OAT, and DynaFL

Table 3
Security threats of user and recommended solutions.

Security threats	Countermeasures
Compromise VNF provider and infrastructure provider: Privacy and confidentiality violation	Trust management to ensure information integrity provision, Cryptographic techniques: TPM, and Intel's TxT
Malicious user: Information leakage, Service violation	Cryptographic techniques, Hiding co-serving instance: Intel's SR-IOV

(TC) technologies are recommended. The TC technologies, developed by the trusted computing group (TCG), are a set of specifications, products, and techniques that are designed in both hardware and software to provide resource isolation, validation mechanism, and trust management.

The TC technology leads to hardware-based "roots of trust" at the edge of the network and at the endpoints in the boot and launch environment (e.g., BIOS and virtual machine managers). The TPM [57], initially specified by TCG to deploy on PC platform, but it is deployed widely on the virtualized platforms [59]. Although the early versions of TPM (e.g., TPM 1.1 or TPM 1.2) showed some shortcomings, but gradually, TPMs (TPM 2.0) have been able to overcome the limitations and become a mainstream component to provide isolated and secure storage, attesting the identity of the running software [60]. Basically, TPM is a trusted hardware component to store encryption keys and hash measurements to provide a chain of trust [47]. As the practical models based on the TC technology to memory protection and resource isolation, there are the follow technologies such as the Intel's VT-d, and Intel's TxT technologies. The Intel's VT-d technology [56] is a hardware-based virtualization solution which provides isolation and protection across partitions, and Intel's TxT technology [43] defines platform-level enhancements based on cryptographic hash algorithm to provide the building blocks for creating trusted platforms.

To provide a safe access management, the remote attestation service (e.g., Intel's Open Attestation (OAT) [49]) and trusted network access control can be considered. The remote attestation service provides the verification of the software state of a remote embedded device to manage network access control and detect software attacks [61]. The trusted network access control [50] utilizes a cryptographic proof or attestation to manage access to the

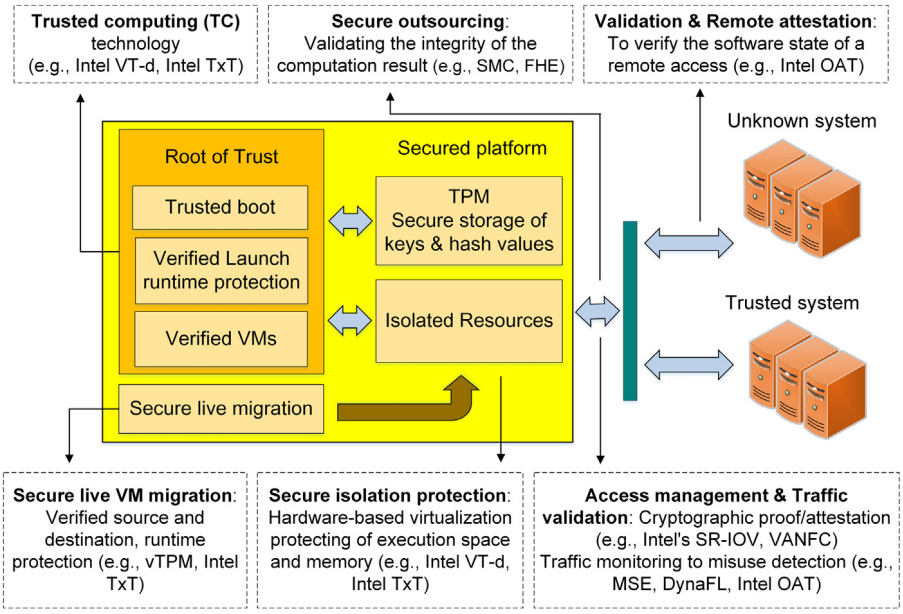

Fig. 5. A suggested platform for a secure virtualization environment.

network resources. Furthermore, Intel's SR-IOV technology [45], which is a feature of Intel virtualization technology to implement virtual access of a physical network interface directly while bypassing the hypervisor's virtual switch, is a good solution for isolation. This technology with the enhancement of virtualization-aware network flow controller (VANFC) [46] is a safe solution for a NFV's user to access the network interface in the presence of suspicious providers. The VANFC prevents untrusted VM from controlling the throughput and latency of other unrelated VMs using network flow control functionality.

To implement a secure virtualization environment, we need to consider several security mechanisms. Relying on trusted third party to perform some mechanisms, such as remote attestation and secure outsourcing, leads to some costs.

As a sequence, a suggested platform for NFV which is protected with the virtualization technologies is depicted in Fig. 5. With regard to the mentioned security protections, the suggested platform is equipped with the proper protection technologies and methods. To address each security issue, at least one protection method is suggested for this platform to provide a secure virtualization environment. In the view of performance overhead, however, the suggested platform exploits some technologies which lead to overall time and computational overheads. For instance, cryptographic solutions for secure outsourcing (as shown in [54] for FHE) or access management lead to key management and computational overheads. As another example, TPMs seem improper to perform bulk data cryptographic operations due to their performance limitations. Moreover, in conventional TPMs implemented in hardware, it is required to have a separated bus to CPU, which may cause severe latency issues [60]. To overcome the limitations of hardware-based TPM, firmware-based TPM platforms [60] and TPM emulators [62,63] could alternatively be used since they have shown better performance compared with hardware-based TPM.

As a practical model based on the secure computing platforms, the High assurance platform (HAP) [64] program, which is a national security agency (NSA) initiative, is a good example to create a secure networked enterprise environments. In this platform, the root of trust is established based on the TC technologies and all outsourcing and remote accesses are verified by attestation and runtime protected. The HAP exploits TC technologies based on TPM, such as Intel's VT-d, TxT, OAT, and SR-IOV, to perform resource isolation, launch-time protection, trustworthiness verification, and remote attestation which are required for secure virtualization environments.

5. Conclusion

The considerable properties of NFV lead to a global trend that network functions are implemented in cloud through virtualization. NFV provides many benefits of the virtualization by introducing software-based appliances and utilizing cloud computing. Even with such many advantages and revenues, NFV also faces several significant security challenges driven by the virtualization and network infrastructure.

In order to understand those security challenges and defense mechanisms, we presented a categorization of security threats on the network functions virtualization and their characteristics. We also proposed possible solutions to address those threats. As future work, we will implement a testbed for an NFV framework with security resilience through simulation, emulation, or real implementation in order to prove our concepts for security in NFV.

Our observations in this study can generally be extended to other distributed systems such as grid computing, which enables the sharing, selection, and aggregation of a wide variety of geographically distributed computational machines [65]. In grid computing, if a computational machine is attacked, the entire system may have also become completely compromised since the system could be synchronized with the compromised machine. As an extension to this paper, we plan to consider a security threat categorization to generalize our results for such grid computing situations.

Acknowledgments

This work was supported in part by the NRF Korea (No. 2014R1A1A1003707), the ITRC (IITP-2015-H8501-15-1008), and the MSIP/IITP (R0166-15-1041, R-20160222-002755). Authors would like to thank all the anonymous reviewers for their valuable feedback.

References

[1] ETSI, Network Functions Virtualisation; Introductory White Paper. Technical Report, SDN and OpenFlow World Congress, 2012.
[2] ETSI, Network Functions Virtualisation (NFV); Infrastructure Overview. Technical Report, ETSI GS NFV-INF 001 V1.1.1, 2015.
[3] ONF, Software-Defined Networking: The New Norm for Networks. Technical Report, ONF White Paper, 2012.
[4] M. Aiash, G. Mapp, O. Gemikonakli, Secure live virtual machines migration: Issues and solutions, in: 28th International Conference on Advanced Information Networking and Applications Workshops, 2014, pp. 160–165.
[5] F. Hu, Q. Hao, K. Bao, A survey on software-defined network and openflow: From concept to implementation, IEEE Commun. Surv. Tutor. 16 (4) (2014) 2181–2206.
[6] S. Shafieian, M. Zulkernine, A. Haque, Cloud Computing: Challenges, Limitations and R & D Solutions, Springer, 2014, pp. 3–22.
[7] Y. Luo, E. Murray, T.L. Ficarra, Accelerated virtual switching with programmable NICs for scalable data center networking, in: Proceedings of the Second ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures, 2010, pp. 65–72.
[8] B. Han, V. Gopalakrishnan, L. Ji, S. Lee, Network functions virtualization: Challenges and opportunities for innovations, IEEE Commun. Mag. 53 (2) (2015) 90–97.
[9] P. Veitch, M.J. McGrath, V. Bayon, An instrumentation and analytics framework for optimal and robust NFV deployment, IEEE Commun. Mag. 53 (2015) 126–133.
[10] Y. Li, M. Chen, Software-defined network function virtualization: A survey, IEEE Access 3 (2015) 2542–2553.
[11] A. Blenk, A. Basta, M. Reisslein, W. Kellerer, Survey on network virtualization hypervisors for software defined networking, IEEE Commun. Surv. Tutor. 18 (1) (2016) 655–685.
[12] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, R. Boutaba, Network function virtualization: State-of-the-art and research challenges, IEEE Commun. Surv. Tutor. 18 (1) (2016) 236–262.
[13] D. Cotroneo, L. De Simone, A.K. Iannillo, A. Lanzaro, R. Natella, J. Fan, W. Ping, Network function virtualization: Challenges and directions for reliability assurance, in: IEEE International Symposium on Software Reliability Engineering Workshops, ISSREW, 2014, pp. 37–42.
[14] Alcatel-Lucent, Providing Security in NFV- Challenges and Opportunities, Alcatel-Lucent White Paper. Technical Report, Alcatel-Lucent, 2014.
[15] H. Hawilo, A. Shami, M. Mirahmadi, R. Asal, NFV: state of the art, challenges and implementation in next generation mobile networks (vEPC), IEEE Netw. Mag. 28 (6) (2014) 18–26.
[16] ETSI, Network Functions Virtualisation (NFV); NFV Security; Security and Trust Guidance. Technical Report, ETSI GS NFV-SEC 003 V1.1.1, 2014.
[17] ETSI, Network Functions Virtualisation (NFV); Infrastructure; Hypervisor Domain. Technical Report, ETSI GS NFV-INF 004 V1.1.1, 2015.
[18] BROCADE, Intel and Brocade Network Functions Virtualization Solution. Technical Report, Brocade Communications Systems, 2014.
[19] V. Ashktorab, S.R. Taghizadeh, Security threats and countermeasures in cloud computing, Int. J. Appl. Innov. Eng. Manag. 1 (2) (2012) 234–245.
[20] M.R. Ogiela, U. Ogiela, Shadow generation protocol in linguistic threshold schemes, Commun. Comput. Inf. Sci. 58 (2009) 35–42.
[21] CSA, The Notorious Nine: Cloud Computing Top Threats in 2013. Technical Report, Cloud Security Alliance, 2013.
[22] S. Natarajan, T. Wolf, Security issues in network virtualization for the future internet, in: International Conference on Computing, Networking and Communications, ICNC, 2012, pp. 537–543.
[23] W. Eatherton, The push of network processing to the top of the pyramid, in: Keynote Presentation at ACM/IEEE Symposium on Architectures for Networking and Communication Systems, ANCS, 2005.
[24] L. Rizzo, M. Carbone, G. Catalli, Transparent acceleration of software packet forwarding using netmap, in: Proceedings of IEEE INFOCOM'2012, 2012, pp. 2471–2479.
[25] A. Shieh, S. Kandula, E.G. Sirer, SideCar: Building programmable datacenter networks without programmable switches, in: Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, Hotnets-IX, 2010, pp. 21:1–21:6.
[26] C. Kachris, G. Sirakoulis, D. Soudris, Network function virtualization based on FPGAs: A framework for all-programmable network devices, CoRR, 2014.
[27] M.B. Anwer, M. Motiwala, M. Tariq, N. Feamster, SwitchBlade: A platform for rapid deployment of network protocols on programmable hardware, ACM SIGCOMM Comput. Commun. Rev. 40 (4) (2010) 183–194.
[28] P. Costa, M. Migliavacca, P. Pietzuch, A.L. Wolf, NaaS: Network-as-a-service in the cloud, in: 2nd USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services, 2012.
[29] V. Varadarajan, T. Kooburat, B. Farley, T. Ristenpart, M.M. Swift, Resource-freeing attacks: Improve your cloud performance (at your neighbor's expense), in: Proceedings of the ACM Conference on Computer and Communications Security, CCS'12, 2012, pp. 281–292.
[30] J. Idziorek, M. Tannian, D. Jacobson, Detecting fraudulent use of cloud resources, in: Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop, CCSW'11, 2011, pp. 61–72.
[31] J. Oberheide, K. Veeraraghavan, E. Cooke, J. Flinn, F. Jahanian, Virtualized in-cloud security services for mobile devices, in: Proceedings of the First Workshop on Virtualization in Mobile Computing, MobiVirt'08, 2008, pp. 31–35.
[32] S. Gutz, A. Story, C. Schlesinger, N. Foster, Splendid isolation: A slice abstraction for software-defined networks, in: Proceedings of the First Workshop on Hot Topics in Software Defined Networks, HotSDN'12, ACM, 2012, pp. 79–84.
[33] Y. Zhang, A. Juels, M.K. Reiter, T. Ristenpart, Cross-VM side channels and their use to extract private keys, in: 2012 ACM Conference on Computer and Communications Security, CCS'12, 2012, pp. 305–316.
[34] T. Ristenpart, E. Tromer, H. Shacham, S. Savage, Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds, in: Proceedings of CCS 2009, 2009, pp. 199–212.
[35] Z. Wu, Z. Xu, Wang, Whispers in the hyper-space: High-speed covert channel attacks in the cloud, in: 21st USENIX Security Symposium, (USENIX Security 12), USENIX, 2012, pp. 159–173.
[36] D.A. Osvik, A. Shamir, E. Tromer, Cache attacks and countermeasures: The case of AES, in: Proceedings of the Cryptographers' Track at the RSA Conference on Topics in Cryptology, CT-RSA'06, 2006, pp. 1–20.
[37] Clementine Maurice, Christoph Neumann, Olivier Heen, Aurelien Francillon, C5: Cross-cores cache covert channel, in: Detection of Intrusions and Malware, and Vulnerability Assessment, DIMVA'15, Springer, 2015, pp. 46–64.
[38] V. Varadarajan, Y. Zhang, T. Ristenpart, M. Swift, A placement vulnerability study in multi-tenant public clouds, in: 24th USENIX Security Symposium (USENIX Security 15), 2015, pp. 913–928.
[39] J. Oberheide, E. Cooke, F. Jahanian, Empirical exploitation of live virtual machine migration, in: BlackHat DC Convention, 2008.
[40] S. Berger, R. Cáceres, K.A. Goldman, R. Perez, R. Sailer, L. van Doorn, vTPM: Virtualizing the trusted platform module, in: 15th Conference on USENIX Security Symposium, in: USENIX-SS'06, vol. 15, 2006.
[41] F. Zhang, H. Chen, Security-preserving live migration of virtual machines in the cloud, J. Netw. Syst. Manage. 21 (4) (2013) 562–587.
[42] S. Biedermann, M. Zittel, S. Katzenbeisser, Improving security of virtual machines during live migrations, in: 2013 Eleventh Annual International Conference on Privacy, Security and Trust, PST, 2013, pp. 352–357.
[43] Intel Trusted Execution Technology (Intel TxT), Hardware-based Technology for Enhancing Server Platform Security, 2012.
[44] MSE: Message Stream Sncryption protocol, Azureu Wiki, 2014. http://wiki.vuze.com/w/MessageStreamEncryption.
[45] PCI-SIG SR-IOV Primer, An Introduction to SR-IOV Technology, Intel SR-IOV, 2011.
[46] I. Smolyar, M. Ben-Yehuda, D. Tsafrir, Securing self-virtualizing ethernet devices, in: 24th USENIX Security Symposium, USENIX Security 15, 2015, pp. 335–350.
[47] X. Zhang, Ch. Lan, A. Perrig, Secure and scalable fault localization under dynamic traffic patterns, in: Proceedings of the IEEE Symposium on Security and Privacy, SP'12, 2012, pp. 317–331.
[48] I. Chen, F. Bao, M. Chang, J. Cho, Dynamic trust management for delay tolerant networks and its application to secure routing, IEEE Trans. Parallel Distrib. Syst. 25 (5) (2014) 1200–1210.
[49] Intel Trust Execution Technology, (Intel TXT) Enabling Guide, 330139-001US, 2015.
[50] F. Bernal, M. Sanchez, G. López, A.F. Gomez-Skarmeta, O. Canovas, Trusted network access control in the eduroam federation, in: Third International Conference on Network and System Security, 2009, NSS'09, 2009, pp. 170–175.
[51] T. Kim, M. Peinado, G. Mainar-Ruiz, STEALTHMEM: System-level protection against cache-based side channel attacks in the cloud, in: 21st USENIX Security Symposium, USENIX Security 12, 2012, pp. 189–204.
[52] Y. Zhang, A. Juels, A. Oprea, M.K. Reiter, HomeAlone: Co-residency detection in the cloud via side-channel analysis, in: 2011 IEEE Symposium on Security and Privacy, SP'11, 2011, pp. 313–328.
[53] C. Gentry, Fully homomorphic encryption using ideal lattices, in: Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing, STOC'09, 2009, pp. 169–178.
[54] L. Melis, H.J. Asghar, E. De Cristofaro, M.A. Kaafar, Private processing of outsourced network functions: Feasibility and constructions, in: The 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, SDN-NFV Security'16, ACM, 2016, pp. 39–44.
[55] OpenAttestation, Build and Install OpenAttestation (2.0), 2013. https://github.com/OpenAttestation/OpenAttestation/wiki.
[56] Intel Virtualization Technology for Directed I/O, Architecture Specification, 2014.
[57] Trusted Platform Module Library, TPM 2.0, 2014.
[58] Intel Software Guard Extensions Programming Reference, Intel-SGX, 2014.
[59] J.D. Osborn, D.C. Challener, Trusted Platform Module Evolution. Technical Report 2, Johns Hopkins APL Technical Digest, 2013.
[60] H. Raj, S. Saroiu, A. Wolman, R. Aigner, J. Cox, P. England, C. Fenner, K. Kinshumann, J. Loeser, D. Mattoon, M. Nystrom, D. Robinson, R. Spiger, S. Thom, D. Wooten, fTPM: A Firmware-based TPM 2.0 Implementation. Technical Report MSR-TR-2015-84, Microsoft Research, 2015.
[61] A. Francillon, Q. Nguyen, K.B. Rasmussen, G. Tsudik, A minimalist approach to remote attestation, in: Proceedings of the Conference on Design, Automation & Test in Europe, DATE'14, Vol. 466, 2014, pp. 1–6.
[62] M. Strasser, H. Stamer, A software-based trusted platform module emulator, in: 1st International Conference on Trusted Computing and Trust in Information Technologies: Trusted Computing - Challenges and Applications, Trust'08, Springer-Verlag, 2008, pp. 33–47.

[63] S. Alsouri, T. Feller, S. Malipatlolla, S. Katzenbeisser, Hardware-based Security for Virtual Trusted Platform Modules. Technical Report, 2013, arXiv preprint.
[64] NSA, High Assurance Platform. Technical Report, HAP Fact sheet, National Security Agency, 2011.
[65] H. Dai, Sh. Zhao, Zhang, M.J. Qiu, L. Tao, Security enhancement of cloud servers with a redundancy-based fault-tolerant cache structure, Future Gener. Comput. Syst. 52 (C) (2015) 147–155.

Mahdi Daghmehchi Firoozjaei is a Ph.D. student in the Department of Computer Science and Engineering, College of Information and Communication Engineering, Sungkyunkwan University. He received the B.S. degree in telecommunication engineering from the Scientific-Applied Faculty of Post and Telecommunication, Tehran, Iran, in 2000, and the M.S. degree in cryptology from Imam Hossein Comprehensive University, Tehran, Iran, in 2005. In 2006, he joined the Telecommunication Company of Mazandaran (TCM) as a network expert until 2014. His research interest is focused on network security and privacy preserving in the cellular networks.

Jaehoon (Paul) Jeong is an assistant professor in the Department of Software at Sungkyunkwan University in Korea. He received his Ph.D. degree in the Department of Computer Science and Engineering at the University of Minnesota in 2009. He received the B.S. degree in the Department of Information Engineering at Sungkyunkwan University and the M.S. degree from the School of Computer Science and Engineering at Seoul National University in Korea, in 1999 and 2001, respectively. His research areas are cyber–physical systems, vehicular networks, wireless sensor networks, and Internet of things, and security in software-defined networking and network functions virtualization. His two data forwarding schemes (called TBD and TSF) for vehicular networks were selected as spotlight papers in IEEE Transactions on Parallel and Distributed Systems in 2011 and in IEEE Transactions on Mobile Computing in 2012, respectively. Dr. Jeong is a member of ACM, IEEE and the IEEE Computer Society.

Hoon Ko obtained his B.S. degree in computer science from Howon University, Gunsan, Republic of Korea, in 1998. He received his M.S. degree in computer science from Soongsil University, Seoul, Republic of Korea, in 2000, and his Ph.D. degree in 2004. He had joined Daejin University, Pocheon, Republic of Korea, as a Visiting Professor from 2002 to 2006. He had worked at Korea Advanced Institute of Science and Technology (KAIST), Seoul, Republic of Korea, in 2007. Ko had also joined GECAD, ISEP, and IPP in Porto, Portugal, as a Doctor Researcher from 2008 to 2013. Recently, he had worked at the Department of Informatics, J.E. Purkyně University, Ústí nad Labem, Czech Republic, as a Research Professor from June 10, 2013, to July 1, 2015. He has been a Researcher at GECAD, ISEP, since 2008. Now, he works at Department of Computer Science and Engineering, Sungkyunkwan University since March, 2016. Ko is interested in urban computing security, ubiquitous computing security, AMI security, context-aware security, multicast security (MSEC), RFID security, and home network security.

Hyoungshick Kim is an assistant professor in the Department of Computer Science and Engineering, College of Information and Communication Engineering, Sungkyunkwan University. He received a B.S. degree from the department of Information Engineering at Sungkyunkwan University, a M.S. degree from the Department of Computer Science at KAIST and a Ph.D. degree from the Computer Laboratory at University of Cambridge in 1999, 2001 and 2012, respectively. After completing his Ph.D., he worked as a post-doctoral fellow in the Department of Electrical and Computer Engineering at the University of British Columbia. He previously worked for Samsung Electronics as a senior engineer from 2004 to 2008. He also served as a member of DLNA and Coral standardization for DRM interoperability in home networks. His current research interest is focused on social computing and usable security.