18

Security Aspects of SDMN

Edgardo Montes de Oca and Wissam Mallouli
Montimage EURL, Paris, France

18.1 Overview

This chapter presents the security issues introduced by software defined networking (SDN), network function virtualization (NFV), and future mobile networks that integrate these technologies to become software defined mobile networks (SDMNs). Even though existing fault management and network security solutions used in traditional networks are sometimes also applicable in SDMN, the concepts introduced by these technologies bring new opportunities, challenges, and vulnerabilities that need to be investigated or addressed.

The introduction of centralized controllers, network virtualization, programmability, and NFV; the separation of the control plane and the data plane; the introduction of new network functions; and even the introduction of new stakeholders such as mobile virtual network operators (MVNO) will all have impact on how security needs to be assured and managed.

To better understand these issues, in Section 18.2, we present an overview of the state of the art; in Section 18.3, we give a more detailed analysis of security monitoring techniques; and in Section 18.4, other important issues are presented: reaction and mitigation techniques, economic viability, and secure services.

18.2 State of the Art and Security Challenges in SDMN Architectures

Existing security techniques applied or applicable in SDMN will be presented in this section, including techniques for end-to-end security and privacy, monitoring techniques (IDS, IPS, behavior, QoS statistics, etc.), security of virtual and physical network elements (NEs) and interfaces, and reaction and mitigation techniques.

SDMNs impose new challenges on network security involving LTE-EPC mobile network security, cloud security, Internet security, and SDN security.

Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture, First Edition.
Edited by Madhusanka Liyanage, Andrei Gurtov, and Mika Ylianttila.

18.2.1 Basics

Security in networks involves making sure that the network provides the services expected from it and that the subscribers can rely on them without prejudice. Several issues need to be considered that include the following main categories:

Identification: Users need to be identified in a unique manner. In LTE-EPC, the International Mobile Subscriber Identity (IMSI) is provided via the USIM card and is stored in the Home Subscriber Server (HSS) database.

Mutual authentication: Users (e.g., subscribers, administrators) and NEs need to be able to interact with the assurance that all parties involved are who they claim to be. LTE-EPC provides similar security features as its predecessors (UMTS and GMS).

Access control: Prevents unauthorized use of the network and services by maintaining a user equipment (UE) profile in the HSS database.

Integrity: Interactions include the communication of control plane and user plane data that should not be modified in an unauthorized or undetected manner. In LTE-EPC, this is assured for control plane data only. For the Nonaccess Stratum (NAS) network, both encryption and integrity are provided.

Confidentiality: Privacy, or the ability to control or restrict access so that only authorized individuals or elements can view or understand sensitive information, also needs to be assured. LTE-EPC defines mechanisms to ensure data security during its transmission over the air interface and through the LTE-EPC system by encryption of both user plane and control plane data (e.g., in the Radio Resource Control (RRC) layer). LTE and SDN security will be presented in this section.

Privacy: Keeping identity and location confidential. In LTE-EPC, the MME provides a Globally Unique Temporary Identity (GUTI) to the EU to temporarily replace the IMSI.

Availability: The users need assurance that the network and services are available when required. There is no LTE-EPC integrated feature that deals with this. LTE networks must be strongly safeguarded and proactively monitored from end to end in order to avert casual as well as advanced persistent threats. Monitoring and cyberthreat mitigation will be presented in the next section.

18.2.2 LTE-EPC Security State of the Art

Global System for Mobile Communications (GSM) mobility networks were designed to address mainly privacy and authentication. Encryption and authentication were improved in UMTS and LTE-EPC, and most important, mutual authentication was introduced.

The security model adopted in mobile LTE-EPC networks integrates different security mechanisms at different levels. First of all, it reuses the authentication mechanisms from UMTS, in other words USIM cards in the mobiles, mutual authentication with the network, and key generation (e.g., Ck, Ik). LTE introduces new mechanisms, such as key derivation during mobility to and from LTE (KASME), high-level protection of signaling (including NAS integrity control and ciphering, end-to-end security from mobiles to MME), protection of radio interfaces (Packet Data Convergence Protocol (PDCP) frames, user session ciphering, RRC radio signaling integrity control, and ciphering); and use of HMAC-SHA-256 for successive key derivations. These mechanisms will continue to be used in future 4G and 5G networks, but how they are impacted in NFV contexts is yet to be studied.

EPS adapts GSM and 3G security mechanisms for obtaining an optimized architecture by embedding confidentiality and integrity mechanisms in the EPS protocol stack (as shown in Fig. 18.1). It also needs to interwork with legacy systems. The UE is identified by the Mobility Management Entity (MME) in the serving network that uses authentication data from the home network and triggers the Authentication and Key Agreement (AKA) protocol in the UE. This allows to share a Key Access Security Management Entity (KASME). Further keys can be derived for confidentiality and integrity protection at the NAS level. More keys are derived for confidentiality and integrity protection of the signaling data between the eNB and the UE at the Access Stratum (AS) level. AS signaling integrity and encryption protects the RRC protocol. Confidentiality protection between the UE and the eNB is embedded in the PDCP that performs IP header compression and decompression. No layers below PDCP are confidentiality protected. Integrity protection is not applied between the UE and the eNB, but IPsec can optionally be used to encrypt user data. Likewise, signaling and user data between the eNBs and the core network can be protected using IPsec on the X2, S1-MM2, and S1-U interfaces.

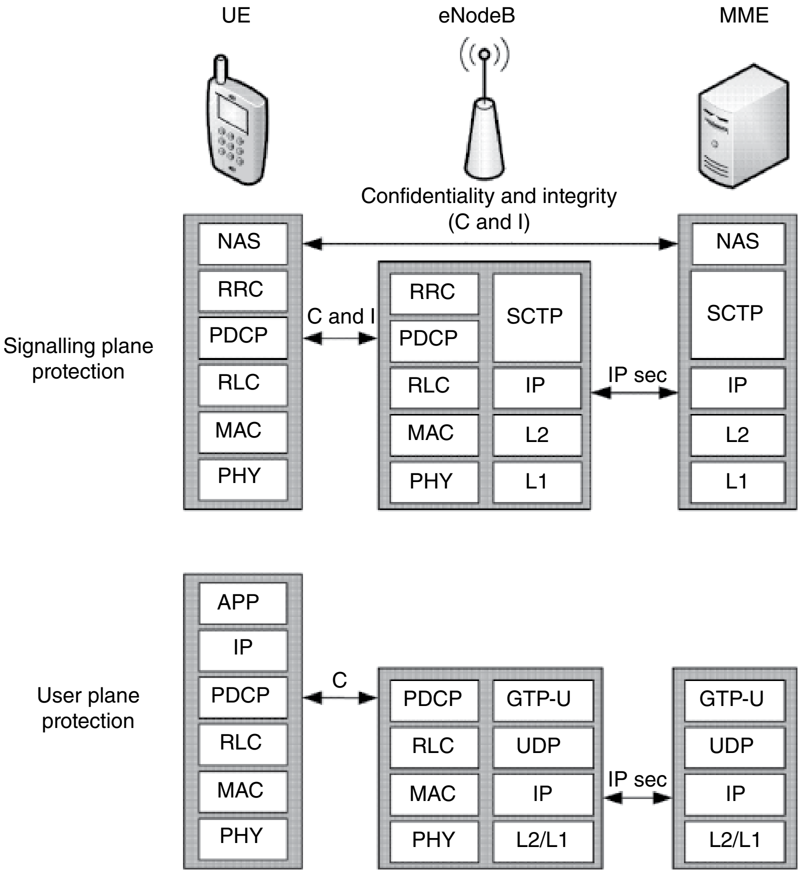


Figure 18.1 LTE-EPC security architecture.

To resume, the role of the different protocol layers is as follows:

- NAS (i.e., all functions and protocols used between the UE and the core network): Performs NAS key handling and integrity and confidentiality protection of NAS. NAS is the layer in charge of managing the establishment of communication sessions and for maintaining continuous communications with the UE as it moves.
- AS (i.e., all functions and protocols used between the UE and the access network): The RRC messages rely on integrity and confidentiality protection from key handling and security activation in PDCP. AS is the layer responsible for carrying information over the wireless portion of the network. PDCP also performs confidentiality protection in the user plane.

The main vulnerabilities in the LTE-EPC security framework concern the system architecture, the access procedures, the handover procedure, and the security mechanism of IP Multimedia System (IMS), Home eNodeB (HeNB), and Machine-Type Communications (MTC). Many vulnerabilities existing in the security framework and the security mechanisms of 4G LTE networks need to be addressed (for a detailed description, see Ref. [1]).

18.2.3 SDN Security in LTE-EPC State of the Art

SDN allows the separation of the control plane and the data plane, enabling the programmability and centralized control of the network infrastructure. From the security point of view, this brings many advantages and disadvantages that will be discussed in the following subsections.

18.2.3.1 Advantages When Introducing SDN

One of the main advantages of SDN is that it simplifies network management and facilitates the upgrade of functionality and debugging. Consequently, introducing SDN in wireless mobile networks allows enhancing security and accelerates innovation in the area. Programmability allows fast and easy implementation and deployment of the new functionality at both hardware and software levels. Automated management reduces operational expenditure (OPEX), while capital expenditure (CAPEX) can be reduced by making it unnecessary to replace the underlying hardware.

SDN-enabled centralized control and coordination make it possible to deliver the state and policy changes more efficiently. SDN introduces vulnerabilities inherent to software-based systems, as we will describe in the next subsection, but at the same time allows improving the resiliency and fault tolerance of centralized controllers using well-known techniques such as automated failovers. Reaction to vulnerabilities and attacks is also improved by giving the ability to quickly assess the network from a centralized viewpoint and making it possible to apply fast dynamic changes and automate mitigation actions.

Another aspect is that it enables NFV. In this way, Internet and cloud service providers can differentiate themselves and propose improved solutions in terms of quality of service (QoS) and security. By introducing virtualized abstraction, the complexity of hardware devices is hidden from the control plane and SDN applications. Furthermore, managed network can be divided into virtual networks (VN) that share the same infrastructure but are governed by different policy and security requirements. SDN and NFV make possible the sharing, aggregation, and management of available resources; enable dynamical reconfiguration and changes of policy; and provide granular control of network and services through the abstraction of the underlying hardware.

The introduction of open SDN standards, such as OpenFlow, not only promotes research and collaborations between different operators and providers but improves the possibility of interoperability in multiservice and multivendor environments and with the legacy systems.

18.2.3.2 Disadvantages When Introducing SDN

One of the main security issues introduced by SDN is that the controllers act as centralized decision points and, as such, become potential single points of attack or failure. Also, the southbound interface (e.g., OpenFlow) between the controller and data-forwarding devices is vulnerable to threats that could degrade the availability, performance, and integrity of the network.

Controllers become a security concern and where they are located and who has access to them needs to be managed correctly. Communications between the controllers and NEs need to be assured by encryption techniques (e.g., SSL), and the keys need to be managed securely. But these techniques are not sufficient to assure high availability because denial-of-service (DoS) attacks remain difficult to detect and counter. Controllers are vulnerable to these types of attacks, and guaranteeing that they are available at all times is a complex task that requires guaranteeing resilience using redundancy and fault tolerance mechanisms. Furthermore, every change and access needs to be monitored and audited for troubleshooting and forensics; and this is more complicated in virtual environments where visibility is often reduced. Thus, the following challenges need to be addressed:

- Secure the controller: Contrary to traditional network architectures where the security functions and mechanisms are orchestrated in a distributed manner, the controller in SDMN architecture is the centralized decision point. Access to such controller needs to be tightly secured and monitored to avoid that an attacker takes control of the NEs.
- Protect the controller: If the controller goes down (e.g., because of a DDoS attack), so goes the network, which means the availability of the controller needs to be maintained.
- Establish trust: Protecting the communications throughout the network is critical. This means ensuring the controller, the applications loaded on it, and the devices it manages are all trusted entities that are operating as they should.
- Create a robust policy framework: What’s needed is a system of checks and balances to make sure the controllers are doing what you actually want them to do.
- Conduct forensics and remediation: When an incident happens, you must be able to determine what it was, recover, potentially report on it, and then protect against it in the future.

In Ref. [2], the authors identify the main threat vectors in an SDN-type architecture (depicted in Fig. 18.2).

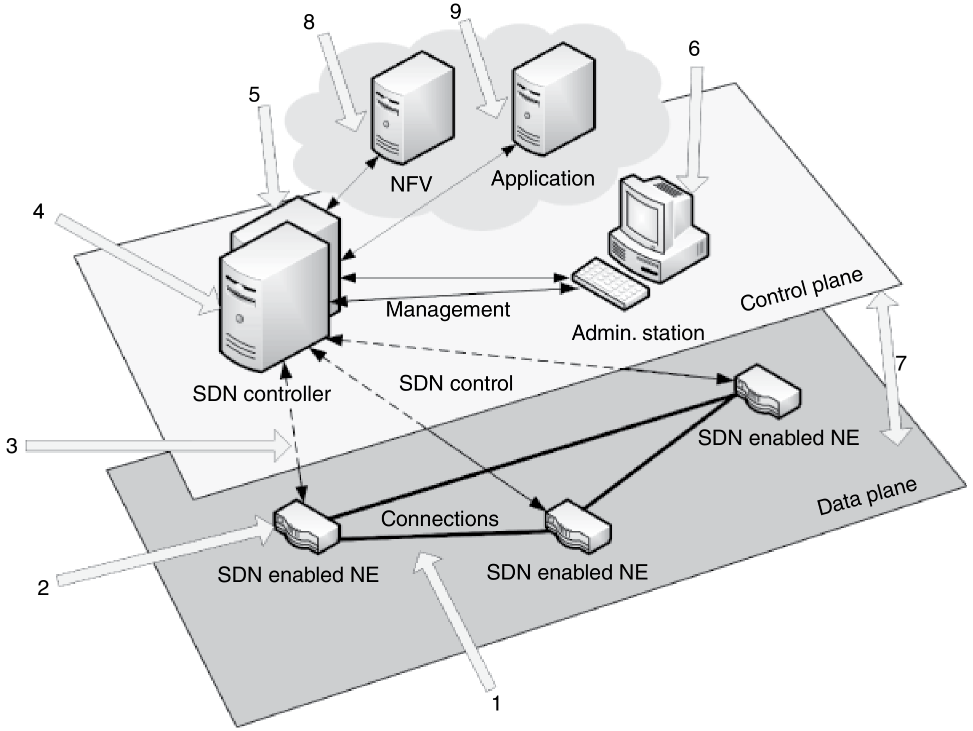


Figure 18.2 Main SDN architecture threat vectors [2].

The seven threat vectors that may enable the exploit of SDN vulnerabilities are:

1. Forged or faked traffic flows: Encryption is not completely reliable and not always possible.
2. Attacks on switches: Introduction of programmability makes them more vulnerable.
3. Attacks on control plane communications: Same issues regarding encryption.
4. Attacks on controllers: Introduction of a new NE or a set of controllers organized hierarchically that need to act in a secure concerted manner.
5. Lack of mechanisms to ensure trust between the controller and management applications: Public key management can be vulnerable.
6. Attacks on administrative stations: Same issues regarding encryption.
7. Lack of trusted resources for forensics and remediation.

To these "classical" vulnerabilities, we need to add vulnerabilities specific to NFV and network programmability:

8. Attack on virtualized network functions.
9. Programmability of network via the controller by untrusted applications.

NFV Specific

Implementing network functions in the cloud introduce vulnerabilities typical in cloud computing. The main security challenges introduced are:

- Introduction of new elements that need to be trust assured, such as virtual machines (VM), virtual switches (VS), hypervisors, controllers, and management modules
- Reduced isolation of network functions
- Resilience dependencies due to resource pooling and multitenancy
- Control of cryptographic keys of hosted network functions

In a cloud environment, multitenancy drives the need for logical separation of virtual resources among tenants. Through orchestration, certain virtualized NFs can be deployed on separate compute nodes, and they can be further segregated by using separate networks. In addition, the use of security zones allows virtualized NFs to be deployed on, or migrated to, hosts that satisfy security-pertinent criteria such as location and level of hardening (e.g., some hosts may employ trusted computing technology).

Automated incident response should include rapid and flexible reconfiguration of virtual resources. If a virtualized NF is suspected of having been compromised (e.g., through unauthorized access via a back door), an uncompromised version can be instantiated to replace it, and the compromised version can be deactivated and be saved for forensic analysis.

In this context, encryption allows protecting the integrity and confidentiality of the signaling and transmitted data, but it is not enough for the following reasons: software is vulnerable and encryption algorithms themselves can be vulnerable (e.g., the OpenSSL Heartbleed bug and backdoors that bypass built-in computer security); the cost of encrypting everything is too high, making it necessary to limit security according to the cost and the risk involved; encryption doesn't mitigate all types of attacks (e.g., DoS attacks); if public keys are used, then how they are managed and stored becomes critical; and a compromised SDN controller potentially allows eavesdropping, exfiltration of data, and unwanted network behavior. Traditional network management tools didn't allow the flexibility to dynamically change the behavior of a network on a node-by-node basis as is possible with SDN.

The Open Networking Foundation (ONF) has identified the southbound communications between controllers and data-forwarding devices as vulnerable. Southbound interface protocols such as OpenFlow have authentication technology that prevents spoofing flow commands from a controller to a switch, but this can be vulnerable if the authentication certificates between controllers and SDN switches are not implemented correctly. Furthermore, authentication cannot prevent DoS attacks from saturating the interface between the control and data planes. To assure secure interactions, they need to be ciphered and monitored, but also, software and hardware need to be kept up to date and also monitored, and unusual behavior that potentially implies a certain level of risk needs to be detected, analyzed, and dealt with.

Programmability Specific

The ONF Northbound Interface Working Group has also been investigating vulnerabilities in northbound communications between the applications and the controllers. Programmability allows installing security applications on the controller's northbound interface to easily introduce new ways to apply security policies on a network. These applications instruct the controller to use the switches and routers that it controls as policy enforcement points. However, the programmable northbound interface is also a potential vulnerability. Here, applications can reprogram the network through the controller, and these can be compromised or contain exploitable vulnerabilities.

Furthermore, as in the case of traditional incompatibilities in routing tables, OpenFlow-type applications have the ability to insert rules that, when combined, may have unexpected results. SDN controllers generally lack the sophistication to understand that security applications should have priority over other applications that communicate with it. Even a harmless application can break security policies if the controller doesn't understand how to handle application requests that contradict security policies. For instance, a security application might quarantine an infected machine, but a load balancing application might still divert traffic to it.

18.2.3.3 Conclusion

To resume, the security issues in SDN are concentrated around the following main areas: (i) application plane, (ii) control plane, (iii) data plane, and (iv) communication security including controller-data path (southbound) and the controller-application (northbound) communication security.

Application Plane Security

SDN enables applications to interact with and manipulate the behavior of network devices through the control layer. SDN has two properties that can be seen as attractive to malicious users and problematic for operators. These properties are, first, the ability to control the network by software and, second, the centralization of network intelligence in network controllers. Since there are no standards or open specifications to facilitate open APIs for applications to control the network services and functions through the control plane, applications can pose serious security threats to the network resources, services, and functions. Although OpenFlow enables deploying flow-based security detection algorithms in the form of security applications, there are yet no compelling OpenFlow security applications [3, 4].

Control Plane Security

In SDNs, the controllers are a particularly attractive target of attack for unauthorized access and exploitation. Without robust and secure controller authentication platform, it is possible to masquerade the controller to carry out malicious activities. Mechanisms to deal with DoS and distributed denial-of-service (DDoS) attacks in large networks are not yet proved viable. Similarly, the controller can become a single point of failure or bottleneck, since the controller southbound and northbound interface securities are also not confirmed. In OpenFlow, most of the complexity is pushed toward controller where forwarding decisions are taken in a logically centralized manner [5]. A challenge for the currently available controller implementations is specifying the number of forwarding devices to be managed by a single controller to cope with the delay or latency constraints. In multiple OpenFlow infrastructures, inconsistency in the controller configurations will result in potential interfederated conflicts [6].

Data Plane Security

In SDNs, switches are most often considered as the basic forwarding hardware accessible via an open interface, while the control logic is moved to the control plane as opposed to the legacy networks where decisions are based on the local configuration of the devices. There are many security challenges for such architectures. For example, if the control plane is compromised, the data plane is handicapped. The data plane is also prone to saturation attacks since it has limited resources to buffer flow initiation (e.g., using TCP/UDP mechanisms) until the controller issues flow rules. Thus, the failure of the control plane has direct implications on the data plane [4]. Recognizing and differentiating genuine flow rules from false rules is another challenge for the data path elements.

Communication Security

The OpenFlow specification defines Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) for the controller–switch communication. The switch and controller mutually authenticate by exchanging certificates signed by a site-specific private key. The switches must be user configurable with one certificate for authenticating the controller and another for authenticating to the controller. Similarly, in the case of the User Datagram Protocol (UDP), the security features are optional and the TLS version is not specified. OpenFlow implementations that use TLS 1.0 may be subjected to man-in-the-middle attacks, as well as other existing attacks against TLS 1.0. OpenFlow implements nonsecure control channel connectivity to ensure interoperability among different systems. However, the standard does not describe how to fall back in case of an authentication failure. Similarly, no mechanisms are demonstrated for application plane and control plane communication.

SDMN Security

SDMNs, carrying the security issues of SDN, have its own set of security concerns. The end user devices in this case often do not have enough processing capabilities, memory, and battery power. Since the communication is IP based, these user devices are prone to the same security threats as their fixed counterparts. The air interface is open to the feats of hacks and thefts; hence, securing the air interface to counter malicious programming of open and programmable network devices is a real challenge. Since the mobile users are mostly on the move and topological changes are frequent, updating the security procedures according to mobility and topological changes is very important. The security between the controller and switches specified in the OpenFlow switch specification is using TLS to secure the channel between the controller and the switch. Similarly, in the case of the UDP, DTLS is described but no mechanism for its usage is currently available. Since there is yet no mobility option available for the use of OpenFlow in mobile networks, SDMNs must develop mobility architectures and the required security mechanisms. SDMNs however must not be limited to an OpenFlow-based architecture, since it is not the only alternative available even in fixed SDN architectures.

18.2.4 Related Work

The concept of a centralized control plane, together with the control channel that is used to exchange information with network devices, introduces new security issues that need to be characterized. From an attacker's perspective, the network controller is attractive due to its important role and so requires specific protection mechanisms. This is an example of a case where network security solutions are more application specific and less dependent on specialized hardware solutions. The scope of such concepts and concrete application scenarios requires a more complete understanding by the research community and stakeholders. As examples, we briefly describe some of the recent research work that is being done.

In Ref. [7], the authors focus on how to utilize SDN to enhance network security. They categorize the target environments into four groups: enterprise networks, cloud and data center, home and edge access, and general design. They analyze different existing security solutions (e.g., OF-RHM, NetFuse, CloudWatcher, AVANT-GUARD, FRESCO, OpenWatch, NIDS Arch., FleXam) and identify the main challenges that include the following: "mobility and roaming" adding dynamicity and therefore complexity to the diagnosis and detection of anomalous activities and security credential exchanges; the "monitoring overhead of OpenFlow-based systems" limiting the effectiveness in the case of high bandwidth and incomplete sample information; "multiaccess and multioperator environment" leading to complex negotiation process, privacy concerns, and potential conflicting policy and QoS requirements that pose a challenge to the security enforcement; and challenges related to the deployment, backward compatibility, interoperability (e.g., between 3G and 4G), and intercommunication with other providers.

Particularly interesting is FleXam [8] that takes into account the optimizations and the dynamics required by a mobile environment. It proposes a flexible sampling extension for OpenFlow to promote the development of security applications such as monitoring. Inspired by this and other works, Ding et al. [7] propose an architecture with local agents that are deployed close to the wireless-edge access to meet the requirements of responsiveness, adaptation, and simplicity. These agents include flow sampling, tracking client records, and mobility profile, and instead of inserting actuation triggers in the data plane, they allow to adaptively query information from the underlying devices and report to the controller, hence alleviating the monitoring load on the central controller.

This architecture has similar objectives as the one proposed in the SIGMONA [9] (a more detailed description can be found in Section 18.3.3).

Besides studying the security vulnerabilities introduced by SDN itself, the authors of Ref. [2] propose a security-by-design technique to achieve secure and dependable SDN platforms based on replicated controllers. Several studies propose solutions based on redundancy. To address the issues of scalability and reliability of centralized controllers, Dixit et al. [10] propose ElastiCon, an elastic distributed controller architecture, in which the controller pool is dynamically grown or shrunk according to traffic conditions and the load is dynamically shifted across controllers.

On the other hand, Araújo et al. [11] study how SDN can be used to guarantee network transport resilience by maintaining multiple virtual forwarding planes that the network assigns to flows. This could be used to mitigate certain types of attacks that provoke path failures. Similarly, Reitblatt et al. [12] present FatTire, a language for writing fault-tolerant network programs based on regular expressions that allows developers to specify the set of paths that packets may take through the network as well as the degree of fault tolerance required. This is implemented using fast-failover mechanisms provided by OpenFlow.

In Ref. [13], the authors propose a hierarchical model of SDN that reduces the number of points of serious failure. Hierarchical deployment of both public key and shared key protocol mechanisms has so far been abstract and largely limited to scalability of cryptographic technology. For the authors, SDN provides an environment with a real need for hierarchical security and raises the question of whether we can use delegation with public key mechanisms, or hierarchical Kerberos mechanisms, to support tiered security in networks. The authors also explain the need for a monitoring service that in turn feeds relevant data back into the management service, completing the loop by connecting to the root-level controllers.

In case TLS is used, a public key infrastructure (PKI) manager is needed, but an alternative could be a Kerberos-type system. Note that the PKI, management, and monitoring services represent concepts and are not necessarily physically separate from the main hierarchy.

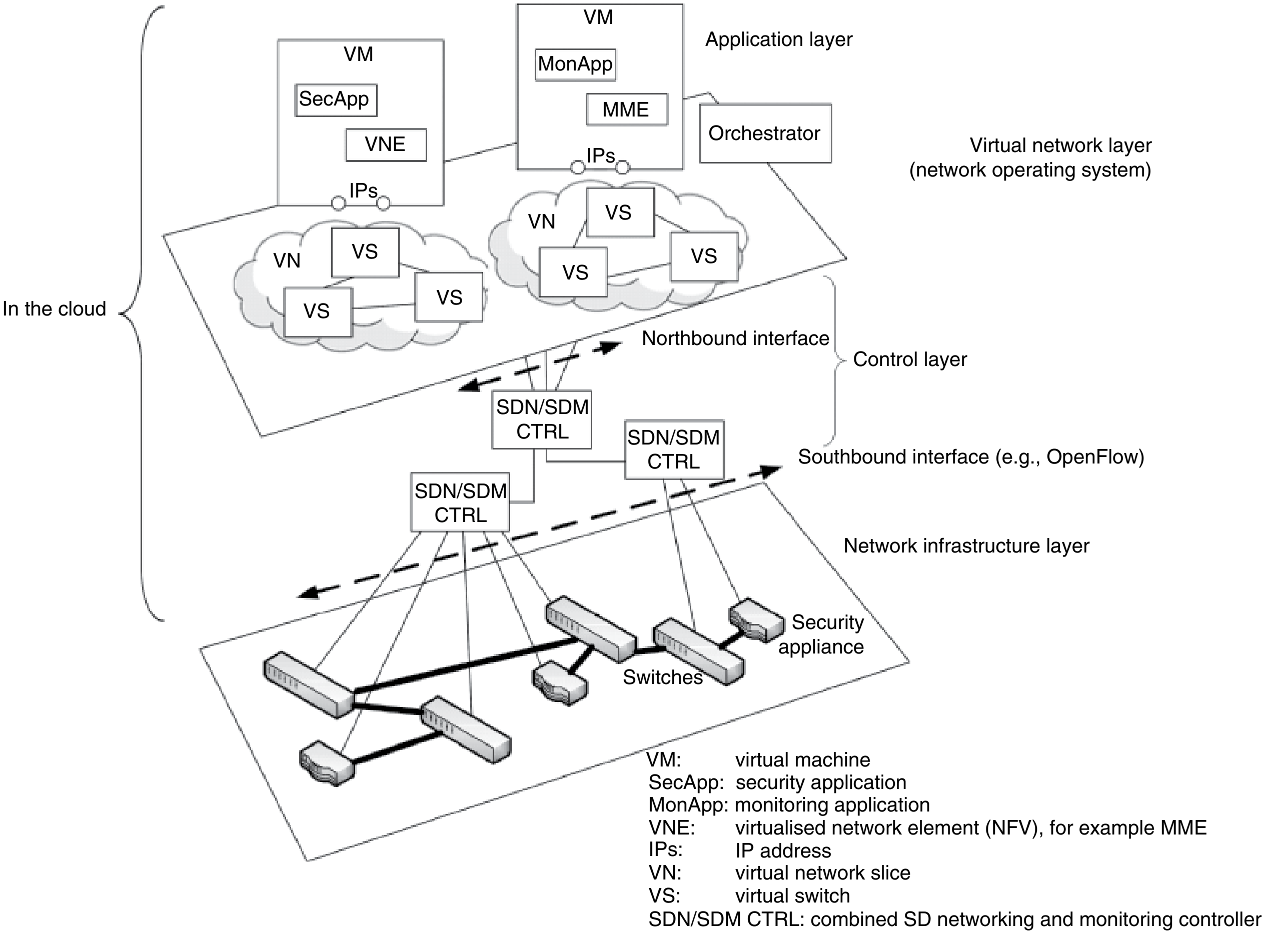
From another perspective, introducing SDN also makes it possible to define high-level configuration and policy statements, which can then be translated to the network infrastructure via OpenFlow-type switches. This eliminates the need for individually configuring network devices each time an endpoint, service, application, or policy changes. Thus, SDN controllers can provide improved visibility and control over the network to ensure access control and security policies are enforced end to end. On the other hand, SDN controllers become single points of failure that need to be integrated into the threat and security model.

A number of challenges need to be addressed related to the introduction of SDN controllers that act as important centralized decision points. They need to be secured, their availability needs to be assured, they need to be integrated in the policy framework, and it needs to be assured that they are acting as expected (e.g., via monitoring as in Fig. 18.4) but also enabling forensics, troubleshooting, and remediation.

Furthermore, security should be deployed, managed, and controlled in an SDN environment. For this, we need to add the possibility of both virtualizing the security functions (i.e., NFV of security) and allowing the security functions to act on virtualized networks and functions. This can be called software defined security (SDS), which is to provide network security enforcement by separating the security control plane from the security processing and forwarding planes. This will result in a dynamic distributed system that virtualizes the network security enforcement functions, scales like VM, and can be managed as a single logical system. In Figure 18.3, a possible architecture that implements SDS is represented. Here, all the functions above the southbound interface can be in the cloud where we have VM that are created by an Orchestrator where the virtualized network functions or virtualized network elements (VNE) are run. These VM are connected via VS forming VN. The software defined network and software defined monitoring controllers (SDN/SDM CTRL) translate the requests from the applications (including network management applications) and configure the physical NEs (e.g., switches, appliances, load balancers).

In this architecture, security analysis and monitoring can be done in the cloud where the virtualization is visible and encryption can be managed (e.g., by supplying the security application with the necessary keys). Nevertheless, security analysis and monitoring can also be done, at least in part, by hardware-specific security appliances, even though doing this might no longer be necessary.

With the logically centralized control plane, SDN enhances network security through global visibility of the network state where a conflict can be easily resolved from a remotely monitoring device. The logically centralized SDN architecture supports highly reactive security monitoring, analysis, and response systems to facilitate network forensics, security policy alteration, and security service insertion [3]. For network forensics, SDN facilitates quick and adaptive threat identification through a cycle of harvesting intelligence from the network to analyze network security, update the security policies, and reprogram the network accordingly. Following this logic, Yu et al. [14] propose a software defined security service (SENSS) solution to facilitate attack diagnosis and mitigation. SENSS has three key features: it is victim oriented, that is, the users have access to network information that concerns their address space and can request security services from multiple remote ISPs (such as statistics gathering, traffic filtering, rerouting, or QoS guarantees); a simple detection/mitigation interface that



VM:	virtual machine
SecApp:	security application
MonApp:	monitoring application
VNE:	virtualised network element (NFV), for example MME
IPs:	IP address
VN:	virtual network slice
VS:	virtual switch
SDN/SDM CTRL:	combined SD networking and monitoring controller

Figure 18.3 Software defined security.

needs to be implemented by the ISPs makes these requests possible; and a programmable attack detection and mitigation across ISPs allows users to program their own attack detection and mitigation solutions across autonomous systems.

An example of an SDS solution is Catbird (www.catbird.com). Here, all security “devices” are managed and controlled by a common security policy language in which the underlying rules are translated by software. The policy is tied to an asset, with potential for many different policies within the same organization depending on the particular requirements of the people and resources within that organization. Security policies are automatically executed, allowing for quick response time while significantly reducing human error. In an SDS environment, it is easy to imagine assets of different “scopes” safely coresiding in the same virtualized host but subject to very different security policies centrally controlled.

There are several key attributes of software defined network security:

- Abstraction: Security is abstracted away from physical constructs such as stateful port firewalls (FW) and wire sniffers and replaced by a set of flexible controls in the form of policy envelopes blanketing the virtualized (or physical) assets. Abstraction is the foundation for establishing common security models that can be deployed repeatedly without concern for underlying physical hardware capabilities.
- Automation: As each asset is redeployed, its security policy trails it. Concerns about inadvertent operator error are eliminated, as SDS can ensure that no asset can be created without being automatically put into a security trust zone. Role-based controls assure that only properly-privileged administrators can make modifications. SDS automation also means wire-speed reaction to anomalous security events, instantly alerting and quarantining as policy would indicate. By contrast, traditional security is still heavily dependent on manual detection, action, and administration.
- Scalability and flexibility: Eliminating dependencies on physical hardware and expense means security can be deployed on a scale appropriate to each host hypervisor, growing in scope commensurate with business needs. Because this is software only, security policy is elastic and can extend across a cluster or a data center. It also means that security is available “on demand.”
- Control orchestration: SDS is designed to integrate a range of network security controls (intrusion detection and prevention, vulnerability management, network segmentation, monitoring tools, etc.) into a single coordinated engine for intelligent analysis and action. Unlimited sources of security input can be funneled into a policy-driven orchestration system, greatly improving the accuracy of the data and attendant action. Orchestration is critical for successful compliance enforcement, as all major compliance standards dictate a variety of controls as parts of the specifications.
- Portability: In a data center governed by SDS, assets carry their security settings with them as they move or scale.
- Visibility: By virtue of being software and thus living within the virtualized infrastructure itself, SDS dramatically improves visibility of network activity. Network administrators and security personnel can detect anomalous behavior that would be blind to them with physical devices and can therefore thwart and protect with a greater degree of accuracy. Network informatics are augmented by this additional data, and NetFlow mapping becomes more extensive and precise.

- Economically viable: The virtualization of the security functions allows dynamically deploying them on existing network infrastructure with minimum CAPEX costs; and the adoption of the SDMN makes their management more flexible (dynamic configuration, countermeasures, etc.) and as a consequence reduces the OPEX costs. These characteristics are unique to SDS and are difficult to attain with traditional security appliances.

18.3 Monitoring Techniques

Network monitoring is required for the verification and validation of SLAs, managing performance (QoS) and user experience (QoE), troubleshooting, assessment of optimizations, and use of resources. In the context of SDMN, network virtualization, and NFV, monitoring needs to be rethought to be able to deal with requirements introduced by virtualization and profit from the flexibility obtained from NFV.

Performance, resource, and security monitoring can be viewed as complementary. Monitoring can provide the knowledge necessary to assure the network's QoS and security. To be able to detect certain types of security issues, performance analysis is necessary. On the other hand, both security breaches and security enforcement mechanisms will have impact on the performance.

LTE-EPC connectivity management of extremely large amounts of devices with various capabilities and intelligence (e.g., mobile phones, ePads, M2M, IoT, etc.) requires automated security services to assure confidentiality and integrity. This leads to high signaling and processing costs and the need for new strategies for cost-effective adaptive security. For this, it is necessary to have a clear view of what is happening in the network and the devices used and how they are used. Monitoring is instrumental for understanding the network traffic and how the services and applications are being used, enabling improved and automated security assurance.

Existing security solutions (e.g., SIEM, IDS, IPS, FW) need to be adapted and correctly controlled since they were meant mostly for physical and not virtual systems and boundaries and do not allow fine-grained analysis adapted to the needs of LTE-EPC and SDN network management. The lack of visibility and controls on internal VN created and the heterogeneity of devices used make many security applications ineffective.

On one hand, the impact of virtualization on these technologies needs to be assessed. For instance, security applications need to be able to monitor virtual connections. Virtualization can help isolate systems but can also be used to make malicious systems that are difficult to detect; for instance, virtualization creates boundaries that could be breached by exploiting vulnerabilities and bugs in the virtualization code (e.g., hypervisors), and the whole systems actually become files that can more easily be stolen.

On the other hand, the security technologies need to cope with ever-changing contexts and trade-offs between the monitoring costs and risks involved. Here, virtualization and SDN facilitate changes, making it necessary for security applications to keep up with this dynamicity.

Security Information and Event Management (SIEM)-type solutions are necessary in order to gain security and status awareness. If an incident happens, the system should be able to determine the source, recover, and protect against it in the future. It should be verified that everything that comes out of the system is logged. Managers have centralized control over the network, and it is necessary to log every change and treat it accordingly in a management solution. Log analysis and event correlation in SDN will fast become a "big data" issue. Tools also are needed that can address all the forensics and compliance requirements.

With SDN, it is possible to create network monitoring applications that collect information and make decisions based on a network-wide holistic view. This enables centralized event correlation on the network controller and allows new ways of mitigating network faults.

Many types of network monitoring techniques exist today that offer different capabilities. First, we have router-based monitoring protocols that allow gathering information supplied by the NEs:

- Simple Network Monitoring Protocol (SNMP): Management of NEs and high-level information on resource use (e.g., monitor bandwidth usage of routers and switches port by port, device information like memory use, CPU load, etc.)
- Remote Monitoring (RMON): Exchange of network monitoring data
- NetFlow or sFlow: Collect information on IP network flows and bandwidth usage

These protocols are dedicated more for performance analysis and network management; but they have also been used for detecting some security problems, as, for instance, NetFlow [15].

We also have packet sniffing, deep packet inspection (DPI), deep flow inspection (DFI), virus scanners, malware detectors, and other techniques for analyzing network packet headers, complete packets, or packet payloads. They are used by Network Intrusion Detection Systems (NIDS), Intrusion Detection and Prevention Systems (IDPS), FW, antivirus scanning appliances, content filtering appliances, etc. and combined with different methods (e.g., statistics, machine learning, behavior analysis, pattern matching, etc.) to detect security breaches (i.e., passive security appliances) or prevent/block detected security problems (i.e., active security appliances).

The adoption of all-IP-type networks introduces vulnerabilities and attacks inherent to the Internet that can be passive or active (i.e., attacks affecting the behavior of the network and services or just dedicated to recuperate information as in the case of scanning and eavesdropping), localized or global (i.e., attacks targeting the network or specific entities or services), and, though less common, insider attacks (i.e., compromised NEs). Also, SDMN introduces new vulnerabilities since it combines mobile phones, the Internet, SDN, network virtualization, and NFV. Some examples are:

- Internet based:
  - DDoS, Smurf, and cyberattacks
  - Spoofing, man in the middle, and ARP poisoning
  - Buffer and heap overflow
  - Format string attack and SQL injection
  - Malware distribution and phishing
  - Data exfiltration
  - Wiretapping and port scanning
- LTE-EPC and mobile network based:
  - Radio and femto-based jamming and saturation of the wireless interface
  - NE vulnerabilities (e.g., in eNodeB, MME), LTE-EPC signaling, and saturation-based attacks

  - M2M-based attacks
  - Infected mobile phones (e.g., same types of attacks as for the Internet)
  - Theft of Service (ToS) (e.g., access to unauthorized services, billing avoidance)
  - Protocol misbehavior
  - Interoperability between network providers and with legacy networks
- SDN based (see Fig. 18.2)
- NFV based (same as cloud computing vulnerabilities)

Furthermore, amplification effects inherent to the operation of mobility networks are made possible due to centralized authentication nodes and HSS [16]. Some research studies have identified the potential risks that amplification attacks brings to LTE-EPC. For instance, a single event triggered on the phone (a state transition in the RRC state machine) implies a substantial number of messages exchanged among several LTE-EPC nodes. This could be exploited to become a DDoS attack by infecting many phones as explained in Ref. [17].

Many other studies have been made to identify and find solutions to the different types of attacks. In the following, we briefly describe some of them addressing LTE-EPC and SDN vulnerabilities:

- Bassil et al. [18] investigate the effects of signaling attacks that consist of malicious users who take advantage of the signaling overhead required to set up and release dedicated bearers in order to overload the signaling plane by repeatedly triggering dedicated bearers requests.
- Jover [16] analyzes attacks that can affect the LTE-EPC network availability. Common DoS and DDoS attacks could have a severe effect on network performance as already demonstrated, for instance, by a fortuitous error in an android application that created havoc in one of the mobile networks [19]. In Ref. [16], the authors identify that advanced persistent threats, which are well organized and financed, can have very negative effects and provoke both general and very targeted attacks. They propose enhancing mobility network security particularly by improving attack detection techniques and constructing a mobile security architecture based on the following main areas:
  1. Introduction of multiple antennas at the eNodeB to enable the possibility of advanced antijamming techniques [19].
  2. Analysis of traffic and signaling load to modify the network configuration to mitigate DDoS attacks. Common NAS operations, such as idle-to-connected and connected-to-idle RRC state transition, can provoke signaling overloads and are potentially a way of attacking mobile networks and M2M.
  3. Introduction of software defined cellular networks allowing deploying network functions in the cloud. This makes it possible to obtain flexible and adaptable security to counter attacks.
  4. Enhancing SDMN standards and architecture to include other security techniques besides encryption and authentication. Interconnectivity and heterogeneity need to be properly addressed. In particular, SDMN needs to take into account M2M and IoT that require being addressable from the Internet in order to deploy services. This opens new attack vectors, especially for multihomed devices.

Monitoring is an important function that is required for addressing points 2 and 4. To be effective, this function needs to be improved in the following main areas:

- Information extraction: Understanding how to deal with virtualization to obtain information on traffic flows, profiles, and properties by means of extracted protocol metadata, measurements, data mining, and machine learning techniques.
- Scalability and performance issues: The design of the monitoring architecture and the location of the observation points need to be done in such a way as to assure scalability and different monitoring use cases that need to be studied to obtain the best balance between performance, cost, and completeness of the results. Furthermore, hardware acceleration and packet preprocessing technologies need to be integrated and controlled by applications and functions to obtain highly optimized solutions.
- Analysis of different control and user plane traffic flows over the network domains and new interfaces between SDMN and existing networks and identification of related flows in different network domains.
- Dynamicity: Changes in virtualized networks and applications become more easy and frequent. Monitoring solutions need to be able to adapt to these changes.

18.3.1 DPI

DPI is a form of network traffic analysis that involves the act of examining the header and payload content of a packet. Initially, DPI was used to help tackle harmful traffic and security threats and to throttle or block undesired or “bandwidth hog” applications. This role has evolved very fast, including in the mobile sector, where DPI can be deployed for a wide range of use cases aimed at helping to assure and improve the performance of individual customer services and to improve the customer quality of experience.

The key function of the DPI is traffic flow identification and classification. To achieve this, the DPI engine can internally utilize various classification methods from explicit layer information to pattern matching, behavior analysis, and session-level correlation. The classification methods make it possible to support a wide range of protocols and applications without the extensive use of resources in the inspection phase.

Moreover, DPI has the capability to extract traffic information (i.e., metadata) from the inspected packet and the related data stream or application sessions. Typically, this includes:

- Extraction of application and quality metrics such as packet loss, jitter, burstiness, and MOS
- Extraction of protocol details such as IP addresses, HTTP URIs, and RTP audio codecs used

KPI values can be calculated based on DPI extraction results and then integrated with predefined threshold values or optionally with trend analysis that helps identify abnormal changes in the traffic profile.

Thus, the functionality of DPI engines and IDS remains the same, essentially the classification of traffic, metadata extraction, data correlation, and identification of malicious or unwanted traffic. The question is how DPI and IDS need to be adapted to deal with SDN, mobile networks, VN, and VNF.

One critical aspect in all of this is that the applications and associated control elements need a holistic view of infrastructure conditions. This is a central and something that DPI, in principle, can provide, by gathering information throughout the network and feeding it back to the control layer (i.e., the controller) and to the applications so as to ensure that the right resources and capabilities are made available and that the security requirements are met.

It is important to note that at all times legal aspects need to be considered. This includes the storing of information as required by law and protecting the privacy of citizens and organizations. As a legally sanctioned official access to private communications, lawful interception is a security process in which a service provider or a network operator collects and provides law enforcement officials with intercepted communications of private individuals or organizations.

18.3.2 NIDS

The nature of mobile networks and virtualization creates new vulnerabilities that do not exist in fixed wired networks. Currently, many of the proven security techniques (e.g., FW, encryption, IDS) are ineffective in these types of networks and applications since they rely on protecting localized physical assets and interfaces. This is not the case when mobility and virtualization are used since they introduce the need for wireless connections open to eavesdropping and active interfering, virtual boundaries not visible and more vulnerable, virtual applications with remote storage and execution, and mobile devices that connect to unprotected networks. Thus, new architectures, techniques, and tools need to be developed to protect the virtual wireless networks and mobile applications; mobile nodes and the infrastructure must be prepared to operate in a mode that trusts no peer [21].

In this article, the authors argue that intrusion detection can complement intrusion prevention techniques (such as encryption, authentication, secure MAC, secure routing, etc.) to secure the mobile computing environment. However, new techniques must be developed to make intrusion detection work better for wireless networks. This is also true for virtualized networks and functions. In SDN, the SDN-enabled switches can make a first evaluation to detect suspicious traffic. The controller can then mirror this traffic so that it can be analyzed by the IDS appliance (i.e., off-path detection). In this way, one avoids interfering with the traffic flow, but there will be a delay in the blocking of unwanted traffic. In the case of online detection, the IDS would intercept real traffic and act as an FW. This will have impact on the latency. The controller could interact with the SDN switch, the FW, or the IDS to filter unwanted traffic.

The use of IDS appliances in the core network has several drawbacks that reduce its effectiveness. First, encryption will make it harder or impossible to analyze the traffic. This means that IDS appliances are effective only at the edge or at the users' premises where the traffic is lower and the encryption is easier to manage. Another problem is the amount of rules that need to be managed by the appliances. Today, the order of magnitude is in the hundreds of thousands where the requirements are more in the order of millions.

The authors of Ref. [21] show that an architecture for better intrusion detection in mobile computing environment should be distributed and cooperative. Anomaly detection is a critical component of the overall intrusion detection and response mechanism. They stipulate that trace analysis and anomaly detection should be done locally in each node and possibly through cooperation with all nodes in the network. Furthermore, intrusion detection should take place in all networking layers in an integrated cross-layer manner. This type of solutions has been studied for ad hoc networks and is also relevant for VN. This collaboration could be done in the SDN context where the SDN controllers exchange data supplied by the IDSs in order to correlate it and make the appropriate decisions.

18.3.3 Software Defined Monitoring

Different architectural possibilities are studied and proposed in the SIGMONA [9] project where an extension of OpenFlow-type interfaces, SDN CTRL INTERFACE (referred to as SDM CTRL INTERFACE in Fig. 18.4), allows obtaining the packet and flow data and metadata needed by the security applications (e.g., the modules referred to as Management/ Monitoring/Security, Applications, and Network Services) from either the switches or the

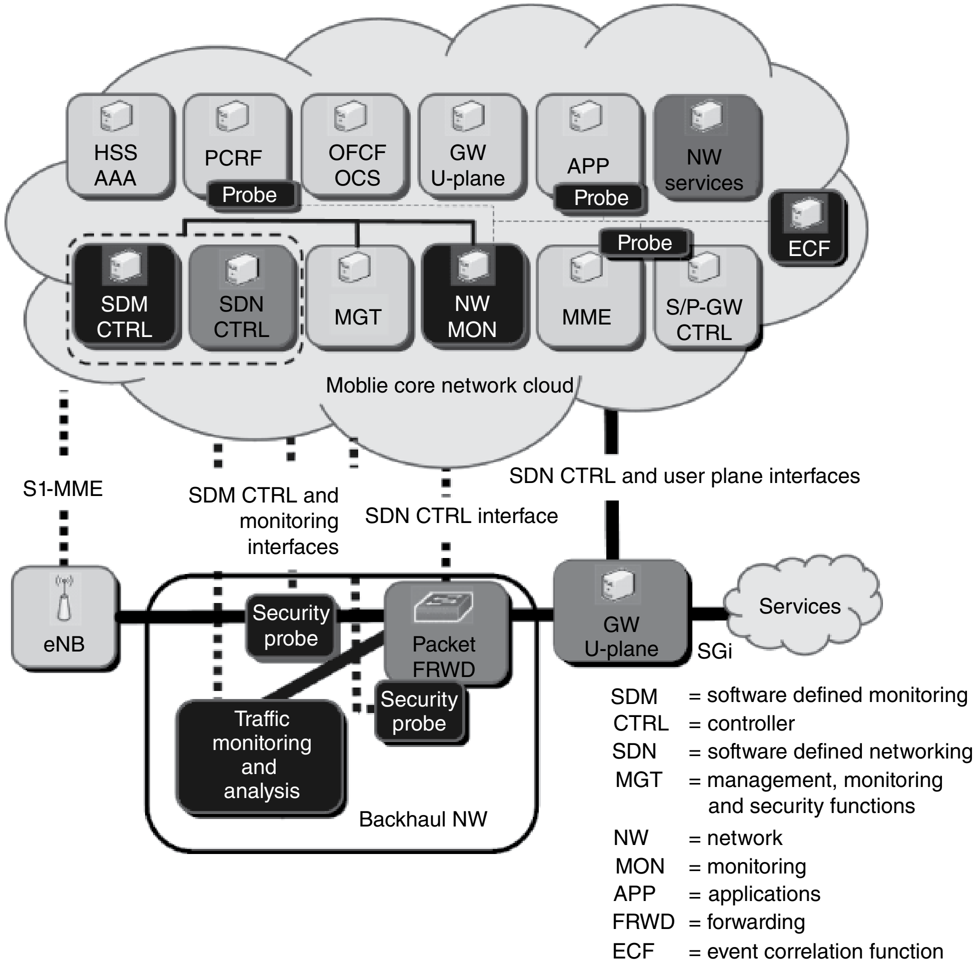


Figure 18.4 Security enhancement framework for SDMN [9].

probes (i.e., agents). The probes can be passive (e.g., the module Traffic Monitoring and Analysis that analyzes mirrored traffic) or active (e.g., the module Active Probe acting as an FW that filters traffic). The SDM CTRL acts as a controller for the software and hardware security devices and could be integrated to the SDN CTRL or separate. If separate, then it will interact with the SDN CTRL via an OpenFlow-type interface. The architecture of the devices and controllers can be hierarchically organized or distributed (e.g., with peer-to-peer communications between the controllers).

The added modules and interfaces are:

- Modules:
  - Security Sensor: An active monitoring probe for the detection of security- and behavior-related information (e.g., security properties and attacks) and mitigation (e.g., filtering). It can be installed on the NEs or in network taps (passive network observation points).
  - SDM CTRL: A new module or extension of SDN CTRL to allow control of monitoring function (e.g., management of network monitoring appliances, traffic mirroring, traffic load balancing and aggregation); accepts requests from network functions and applications. SDM CTRLs are distributed following either a peer-to-peer or hierarchical model. They interact with the Management/Monitoring/Security function and act as distributed analysis or decision points for the defined security policies (security SLAs).
  - Network Monitoring: A virtualization of monitoring function (i.e., part of the traffic analysis moved to the cloud).
  - Traffic Mirroring and Analysis: A passive backhaul traffic monitoring device required by different network functions.
- Interfaces:
  - SDM CTRL INTERFACE: An interface that allows controlling the use of monitoring resources and recuperating traffic or metadata for analysis. It allows performing monitoring requests and obtaining status, so that applications and network functions can send requests for monitoring-based information and monitoring functions can send status and recommendations.
- To this architecture, we need to add virtual monitoring probes (i.e., probes in the cloud) and a virtual event correlation function (ECF) that will allow correlating information captured by the different probes and inform the monitoring function.

By programming flexible switches and other network devices to act as packet interception and redirection platforms, it is potentially possible to detect and mitigate a variety of attacks. By introducing SDN-driven security analysis or software defined monitoring (SDM), SDN-enabled switches, COTS packet processing, and security appliances can act as packet brokers. Controllers can act to aggregate and correlate distributed metadata (e.g., flow and statistical data). This information can be sent to monitoring and analysis appliances and applications. In this way, it is possible to obtain adaptive and optimized monitoring, analysis, and mitigation.

Examples of recently published monitoring-related research work are as follows:

- Distributed monitoring systems are needed to improve both the scalability and accuracy of the security analysis of networks. Yu et al. [22] propose a distributed and collaborative monitoring (DCM) system that allows switches to collaboratively achieve flow monitoring tasks, balance measurement load, and perform per-flow monitoring. It relies on a two-stage Bloom filter to represent monitoring rules using small memory space and centralized SDN control to manage it; but only two rather basic functionalities have been evaluated: flow size counting and packet sampling.

- Choi et al. [23] study scalability problems introduced by centralized SDN control that leads to excessive control traffic overhead to obtain the needed global network visibility. To solve this, they propose a software defined unified monitoring agent (SUMA) that acts as a management middlebox to provide intelligent control, management abstraction, and a filtering layer. Choi et al. [24] also propose a layered control and monitoring management abstraction and filtering solution: software defined unified virtual monitoring function (SuVMF) for SDN-based networks.
- Niels et al. [25] propose a monitoring solution for capturing per-flow metrics (e.g., delay and packet loss) in OpenFlow networks. But the adaptive polling rate technique used that increases when flow rates differ between samples and decreases when flows stabilize to minimize the number of queries could be applied to detecting end-to-end performance problems due to DoS and DDoS attacks. Similarly, Bianchi et al. [26] introduce SDN techniques to improve programming and deployment of online (stream-based) traffic analysis functions that can also be used for detecting security breaches.
- Adapting monitoring techniques to effectively deal with virtualized context is also a major research area. In Ref. [27], SDN, NV, and traditional methods are adapted for gathering evidence and auditing activities on a per-tenant basis, allowing monitoring tenants' VN. Zaalouk et al. [28] study how to adapt the SDN architecture for security use cases. OrchSec, an Orchestrator-based architecture that utilizes network monitoring and SDN control functions to develop security applications, is proposed. Even though limited to sFlow-type data analysis, it shows the benefits that can be derived from SDN-enabled flexibility.
- Wenge et al. [29] outline the current research areas in Security as a Service (SaaS), especially SIEM.

18.4 Other Important Aspects

18.4.1 Reaction and Mitigation Techniques

Cloud computing attack mitigation has been studied by several research teams (e.g., Refs. [30, 31]), but little or no research has been published on mitigation of attacks on LTE and SDN networks.

The work done is mainly how SDN can be used to detect and mitigate attacks on the network. For instance, in the NOVI project Networking Innovations over Virtualized Infrastructures [32], the authors study extending SDN functionalities for performing anomaly detection and mitigation. It is based on flow statistics that may be used to reveal massive DDoS attacks. They demonstrate that OpenFlow statistics collection and processing overloads the centralized control plane and propose a modular architecture for the separation of the data collection process from the SDN control plane with the employment of sFlow monitoring data. The results show that the sFlow-based mechanism is more effective than the native OpenFlow approach and that the OpenFlow protocol can effectively mitigate attacks via flow table modifications.

Similarly in Ref. [33], SDN is used to mitigate attacks detected on virtual appliances. Vizváry and Vykopal [34] present a very succinct analysis of current and future possibilities for the detection and mitigation of DDoS attacks in SDN environments but says little on the vulnerabilities introduced by them.

All the studies are in laboratory settings and are not necessarily adapted to operating network environments. Nevertheless, it has been shown that with SDN flexibility, it is possible to more rapidly react to detected DDoS attacks. But many issues need to be studied, as, for instance, how to block these attacks without blocking legitimate traffic, how to balance risk and cost to obtain efficient solutions, how to scale to large networks, and how to limit the number of false positives detected. Also, work has mainly been oriented toward mitigating DDoS attacks, but many more mitigation types need to be addressed, including attacks on signaling and control plane, more localized DoS attacks difficult to detect using global performance statistics, advanced persistent threats, and compromised network functions.

18.4.2 Economically Viable Security Techniques for Mobile Networks

Economic viability of the solutions and their estimated cost levels is an important aspect that needs to be studied in the context of SDMN.

For instance, securing the controller forcibly has a cost. First of all, it is necessary to know and audit who has access to the controller and where it resides on the network, and security between the controller and end nodes (routers or switches) needs to be assured; likewise, high availability, changes that need to be logged and controlled, existing security devices, and applications need to be configured and integrated correctly.

Many studies have been made to evaluate cost and optimize security mechanisms for the cloud and data centers (e.g., the recent studies: Refs. [35, 36]) and others for mobile applications (e.g., the recent studies: Refs. [37, 38]). In Ref. [39], the authors study new strategies that need to be devised for cost-effective security provision and propose a context-aware security controller for LTE-EPC networks to minimize the overall security cost that activates security mechanisms according to the contextual information such as the application type and the device capabilities. More specific to SPAM mitigation but applicable in general, Bou-Harb et al. [40] identify the high cost of centralized security solutions in LTE-EPC and proposes a distributed architecture made cost-effective by utilizing commercial-of-the-shelf (COTS) low-cost hardware in the distributed nodes to mitigate SPAM flooding attacks.

In Ref. [41], the authors study how to obtain virtualized cost-effective DPI monitoring solutions based on genetic algorithms. They argue that any network function (e.g., DPI, FW, caching, ciphers, load balancers) can be virtualized, but deployment and operation costs due to licensing and power consumption need to be optimized. The genetic algorithms allow this by minimizing the number of deployed DPI engines and determining their location and at the same time minimizing network load introduced by DPI.

Overall, there are two types of costs that need to be considered: CAPEX (i.e., investments costs) and OPEX (i.e., operation costs). In the following, we give some observations that need to be considered when trying to obtain the best cost-effective security.

18.4.2.1 CAPEX

First of all, NFV provides lower total cost of ownership, reducing CAPEX by migrating functions from proprietary to commodity hardware and from dedicated NE to VM, including security appliances and functions.

To illustrate this, for example, for their data centers, Microsoft has developed an OpenFlow-based network tap aggregation platform (Distributed Ethernet Monitoring) for analyzing huge volume of traffic within the cloud network. Traditional network packet brokers that do tap and SPAN port aggregation or mirror ports did not scale. By introducing OpenFlow, it was possible to reduce CAPEX costs since it allowed using single merchant silicon switches (switches using of-the-shelf chip components), replacing more expensive specialized tap/mirror aggregation appliances. OpenFlow controllers allowed easy tailoring of the monitoring and aggregation to adapt to the requirements in an optimized way.

Nevertheless, licensing costs and deployment of network functions need to be optimized for the best results.

Security based on policies requires the ability to deploy large number of rules, for example, when using FW to filter network traffic. Existing FW, even when based on very sophisticated hardware like Content-Addressable Memory (CAM) [42], are limited in the number of rules they can handle without introducing exorbitant costs or affecting traffic latency.

18.4.2.2 OPEX

NFV can ease the operational impact of deploying security updates. An upgraded instance of the virtualized NF can be launched and tested while the previous instance remains active. Services and customers can then be migrated to the upgraded instance. The older instances with security flaws can be deactivated and analyzed once this is complete.

Deployment of new or modified security policies is also facilitated. Security has been difficult to implement even in the traditional networks because of difficulty in enforcing the required security policies in a continually changing environment. SDMN provides new ways of dealing with this problem by enabling introduction of sophisticated network architecture that allows network administrators to dynamically enforce and control fine-grained security mechanisms by relying on NFV. Logically centralized SDN control can help simplify security policy deployment and prevent conflicts and inconsistencies in the security procedures by providing a global view of the configurations of different network devices. Formal methods and proof techniques can also be applied more easily to detect misconfigurations.

The introduction of standards (not all yet completely adopted in the case of SDMN) is also a determining factor to reduce CAPEX and OPEX costs by allowing improved competition between stakeholders, eliminating the need for gateways, allowing the use of commodity hardware, and reducing the learning curves.

18.4.3 Secure Mobile Network Services and Security Management

Mobile services are vulnerable and have become a main target for attacks that include unreliable authentication mechanisms, nonencrypted or poorly secured communications, inauspiciously installed malware, lack of security applications, out-of-date systems and applications, and unauthorized modifications (e.g., jailbreaking, rooting). Virtual and software defined network techniques will facilitate modifying and configuring network functions using centralized controllers, making it easier to adapt security functions to the needs of the mobile services and their users.

Future mobile networks will be a composite of multiple architectures and infrastructures to cover different geographic locations and support different set of services with high data rate. In the current mobile networks, DoS attacks, authorization vulnerability, service degradation attacks, location tracking, and bandwidth stealing are common threats, and more will appear as the usage of the Internet in mobile networks increases. With the migration of IP service, the security challenges will also migrate to mobile devices and networks with the IP services. Mobile devices, though, have lesser resources to counter attacks than fixed stable devices. Therefore, it is widely accepted that security measures incorporated to the network itself must be strengthened first to protect the network infrastructure and architecture and then the mobile devices and its users. The network operators thus will face challenging security issues since security will soon be a key differentiator in their commercialization efforts. Stable and robust security policy deployment would require global analysis of policy configuration of all the security devices in networks to avoid conflicts and inconsistencies in the security procedures. These policies diminish the chances of serious security breaches and network vulnerabilities. Therefore, SDMNs will be strong candidates for future secure mobile networks, since in SDMN the logically centralized control can provide a global view of the configurations of different network devices and hence mitigate the risks of security breaches.

Service chaining is not a new concept, but the trend has taken on a new importance with the rise of SDN and NFV. A service chain is a carrier-grade process for continuous delivery of services based on network function associations, such as FW or application delivery controllers (ADCs) interconnected through the network to support an application. SDN and NFV make the service chain and application provisioning process faster and easier [43, 44]. In the past, building a service chain to support a new application required installing specialized hardware and tailored configurations. Furthermore, service chains did not adapt easily to changes in application needs, so they needed to be overprovisioned and made as generic as possible to support multiple applications. By separating management functions from the infrastructure, SDN and NFV allow standardized automated reconfiguration. Network security functions to support mobile services execute as VM under control of a hypervisor and can be easily adapted to the context and needs of the applications.

For instance, a service chain can consist of an edge router at the customer premises, followed by a DPI service that determines the type of traffic, which in turn informs the controller to create a service chain specific for the customer and the traffic. Another example is an email or web service chain that includes virus, spam, and phishing detection, routed through connections offering the required performance. Thus, service chains allow automated tailoring of network security functions adapted to the needs of the services and customers.

18.5 Conclusion

On the one hand, the introduction of NFV, network virtualization, and SDN facilitates the security assessment and mitigation in future mobile networks. On the other hand, these techniques introduce new vulnerabilities inherent to software- and Internet-based systems and the addition of new elements, for example, SDN controllers. In this chapter, we have presented both the advantages and disadvantages of introducing these technologies in future mobile networks (4G/5G). We also presented ongoing work and possible solutions. We also briefly addressed other issues: mitigation, economic viability, and mobile service security. Other topics that were not covered but are important are standardization, open source solutions, and legal and network neutrality considerations.

References

[1] J. Cao, M. Ma, H. Li, Y. Zhang, Z. Luo; A survey on security aspects for LTE and LTE-A networks; IEEE Communications Surveys and Tutorials 16(1):283–302 (2014).

[2] D. Kreutz, F. Ramos, P. Verissimo; Towards secure and dependable software-defined networks; in Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. ACM, New York, 2013, pp. 55–60; ACM SIGCOMM 2013, Hong Kong, August 12 and August 16, 2013.

[3] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, N. Rao; Are we ready for SDN? Implementation challenges for software-defined networks; Communications Magazine, IEEE 51(7):36–43 (2013).

[4] S. Shin, V. Yegneswaran, P. Porras, G. Gu; Avant-guard: scalable and vigilant switch flow management in software-defined networks; in Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security. ACM, New York, 2013, pp. 413–424; CCS 2013, November 4–8, 2013 Berlin, Germany.

[5] J. Naous, D. Erickson, G. A. Covington, G. Appenzeller, N. McKeown; Implementing an openflow switch on the NetFPGA platform; in Proceedings of the Fourth ACM/IEEE Symposium on Architectures for Networking and Communications Systems. ACM, New York, 2008, pp. 1–9; ANCS 2008, November 6–7, 2008, San Jose, California, USA.

[6] E. Al-Shaer, S. Al-Haj; Flowchecker: configuration analysis and verification of federated openflow infrastructures; in Proceedings of the Third ACM Workshop on Assurable and Usable Security Configuration. ACM, New York, 2010, pp. 37–44; CCS 2010, October 4–8, 2010, Chicago, IL, USA.

[7] A. Yi Ding, J. Crowcroft, S. Tarkoma, H. Flinck; Software defined networking for security enhancement in wireless mobile networks; Computer Networks 66:94–101 (2014).

[8] S. Shirali-Shahreza, Y. Ganjali; Efficient implementation of security applications in OpenFlow controller with FleXam; in Proceedings of IEEE Symposium on High-Performance Interconnects. ACM, New York, 2013, pp. 167–168; ACM SIGCOMM 2013, Hong Kong, August 12 and August 16, 2013.

[9] http://celticplus.eu/project-sigmona/ (project where the authors participate in defining the SDMN architecture) (accessed January 24, 2015).

[10] A. Abhay Dixit, F. Hao, S. Mukherjee, T. V. Lakshman, R. Rao Kompella; Towards an elastic distributed SDN controller; Computer Communication Review 43(4):7–12 (2013).

[11] J. Taveira Araújo, R. Landa, R. G. Clegg, G. Pavlou; Software-defined network support for transport resilience; in Proceedings of Network Operations and Management Symposium (NOMS), 2014 IEEE, pp. 1–8; 5–9 May 2014, Krakow.

[12] M. Reitblatt, M. Canini, A. Guha, N. Foster; FatTire: declarative fault tolerance for software-defined networks; in Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, pp. 109–114, ACM New York; HotSDN '13, Hong Kong, August 12 and August 16, 2013.

[13] Y. W. Chen, J. T. Wang, K. H. Chi, C. C. Tseng; Group-based authentication and key agreement; Wireless Personal Communications, Springer US; February 2012, Volume 62, Issue 4, pp 965–979.

[14] M. Yu, Y. Zhang, J. Mirkovic, A. Alwabel; SENSS: software-defined security service; SIGCOMM Workshop ONS 2014, in Proceedings of the ACM conference on SIGCOMM; Pages 349–350; ACM New York, NY, USA 2014; ONS 2014, March 2014, Santa Clara, CA.

[15] M. Scheck; Cisco's whitepaper: "netflow for incident detection"; http://www.first.org/global/practices/Netflow.pdf (accessed January 24, 2015).

[16] R. Piqueras Jover; Security attacks against the availability of LTE mobility networks: overview and research directions; in Proceedings of 16th International Symposium on Wireless Personal Multimedia Communications (WPMC), 2013, IEEE, pp 1–9; 24–27 June 2013; Atlantic City, NJ.

[17] R. Bassil, A. Chehab, I. Elhajj, A. Kayssi; Signaling oriented denial of service on LTE networks; in Proceedings of the 10th ACM International Symposium on Mobility Management and Wireless Access. ACM, New York, 2012, pp. 153–158; MobiWac ‘12, October 21–25 2012, Paphos, Cyprus Island.

[18] R. Bassil, I. H. Elhajj, A. Chehab, A. I. Kayssi; Effects of signaling attacks on LTE networks; in Proceedings of the 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2013, IEEE, pp 499–504; Barcelona, Spain, 25–28 March 2013.

[19] M. Dano; The Android IM app that brought T-Mobile's network to its knees; Fierce Wireless, October 2010, http://goo.gl/O3qsG (accessed January 24, 2015).

[20] R. Jover, J. Lackey, A. Raghavan; Enhancing the security of LTE networks against jamming attacks; EURASIP Journal on Information Security 2014:7 (2014).

[21] Y. Zhang, W. Lee, Y.-A. Huang; Intrusion detection techniques for mobile wireless networks; Mobile Networks and Applications; 2003, Volume 9 Issue 5, September 2003 Pages 545–556.

[22] Y. Yu, Q. Chen, X. Li; Distributed collaborative monitoring in software defined networks; in Proceedings of the third workshop on Hot topics in software defined networking SIGCOMM'14 ACM Conference, ACM New York, pp. 85–90; August 17–22, 2014, Chicago, IL, USA.

[23] T. Choi, S. Song, H. Park, S. Yoon, S. Yang; SUMA: software-defined unified monitoring agent for SDN; in Proceedings of Network Operations and Management Symposium (NOMS), 2014 IEEE, pp. 1–5; 5–9 May 2014, Krakow.

[24] T. Choi, S. Kang, S. Yoon, S. Yang, S. Song, H. Park; SuVMF: software-defined unified virtual monitoring function for SDN-based large-scale networks; in Proceedings of CFI ‘14 Ninth International Conference on Future Internet Technologies, Article No. 4, ACM New York; CFI'14, June 18–20 2014, Tokyo, Japan.

[25] N. L. M. van Adrichem, C. Doerr, F. A. Kuipers; OpenNetMon: network monitoring in OpenFlow software-defined networks; in Proceedings of Network Operations and Management Symposium (NOMS), 2014 IEEE, pp. 1–8; 5–9 May 2014, Krakow.

[26] G. Bianchi, M. Bonola, G. Picierro, S. Pontarelli, M. Monaci; StreaMon: a data-plane programming abstraction for software-defined stream monitoring; in Proceedings of 26th International Teletraffic Congress (ITC), 2014, IEEE, pp. 1–6; 9–11 Sept. 2014, Karlskrona.

[27] A. TaheriMonfared, C. Rong; Multi-tenant network monitoring based on software defined networking; in Proceedings of On the Move to Meaningful Internet Systems (OTM 2013) Conferences, Lecture Notes in Computer Science Volume 8185, 2013, Springer Berlin Heidelberg, pp. 327–341; OTM 2013, September 9–13, 2013, Graz, Austria.

[28] A. Zaalouk, R. Khondoker, R. Marx, K. M. Bayarou; OrchSec: an orchestrator-based architecture for enhancing network-security using network monitoring and SDN control functions; in Proceedings of Network Operations and Management Symposium (NOMS), 2014 IEEE, pp. 1–9; 5–9 May 2014, Krakow.

[29] O. Wenge, U. Lampe, C. Rensing, R. Steinmetz; Security information and event monitoring as a service: a survey on current concerns and solutions; Praxis der Informationsverarbeitung und Kommunikation 37(2):163–170 (2014).

[30] J. Szefer, P. A. Jamkhedkar, D. Perez-Botero, R. B. Lee; Cyber defenses for physical attacks and insider threats in cloud computing; in Proceedings of the 9th ACM symposium on Information, computer and communications security, pp. 519–524, ACM New York; ASIA CCS ‘14, Kyoto, Japan, June 4–6, 2014.

[31] S. S. Alarifi, S. D. Wolthusen; Mitigation of cloud-internal denial of service attacks; in Proceedings of the 8th International Symposium on Service Oriented System Engineering (SOSE), 2014, IEEE, pp. 478–483; SOSE'14, 7–11 April 2014, Oxford, UK

[32] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, V. Maglaris; Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments; Computer Networks 62:122–136 (2014).

[33] G. Carrozza, V. Manetti, A. Marotta, R. Canonico, S. Avallone; Exploiting SDN approach to tackle cloud computing security issues in the ATC scenario, in: M. Viera, J.C. Cunha (eds), *Dependable Computing* Springer-Verlag, Berlin/Heidelberg 2013, pp. 54–60.

[34] M. Vizváry, J. Vykopal; Future of DDoS attacks mitigation in software defined networks; in Proceedings of the 8th IFIP WG 6.6 International Conference on Autonomous Infrastructure, Management, and Security, Springer Berlin Heidelberg, pp. 123–127; AIMS 2014, Brno, Czech Republic, June 30 – July 3, 2014.

[35] F. Malecki; The cost of network-based attacks; Network Security 2014(3):17–18 (2014).

[36] Y. Chen, R. Sion; Costs and security in clouds; Secure Cloud Computing:31–56 2014, ISBN 978-1-4614-9277-1.

[37] G. Moody, D. Wu; Security, but at what cost?—An examination of security notifications within a mobile application; in Proceedings of the 15th International Conference Human Interface and the Management of Information. Information and Interaction for Health, Safety, Mobility and Complex Environments, Springer Berlin Heidelberg, 2013, pp. 391–399; HCI 2013, Las Vegas, Nevada, USA, July 21–26, 2013.

[38] N. Vrakas, D. Geneiatakis, C. Lambrinoudakis; Evaluating the security and privacy protection level of IP multimedia subsystem environments; IEEE Communications Surveys and Tutorials 15(2):803–819 (2013).

[39] S. B. H. Said, K. Guillouard, J-M. Bonnin; On the benefit of context-awareness for security mechanisms in LTE-EPC networks; 24th IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, PIMRC 2013, London, United Kingdom, September 8–11, 2013. IEEE 2013; pp 2414–2118.

[40] E. Bou-Harb, M. Pourzandi, M. Debbabi, C. Assi; A secure, efficient, and cost-effective distributed architecture for spam mitigation on LTE 4G mobile networks; Security and Communication Networks 6(12):1478–1489 (2013).

[41] M. Bouet, J. Leguay, V. Conan; Cost-based placement of virtualized deep packet inspection functions in SDN; Military Communications Conference, MILCOM 2013—2013 IEEE, San Diego, CA, 2013, pp. 992–997.

[42] A. X. Liu, C. R. Meiners, E. Torng; Packet classification using binary content addressable memory; in Proceedings of INFOCOM, 2014, IEEE, pp. 628–636; INFOCOM'14, April 27 2014–May 2 2014, Toronto, ON.

[43] W. John, K. Pentikousis, G. Agapiou, E. Jacob, M. Kind, A. Manzalini, F. Risso, D. Staessens, R. Steinert, C. Meirosu; Research directions in network service chaining; in Proceedings of SDN for Future Networks and Services, 2013, IEEE, pp. 1–7; SDN4FNS'13, 11–13 Nov. 2013, Trento.

[44] Y. Zhang, N. Beheshti, L. Beliveau, G. Lefebvre, R. Manghirmalani, R. Mishra, R. Patney, M. Shirazipour, R. Subrahmaniam, C. Truchan, M. Tatipamula; StEERING: a software-defined networking for inline service chaining; in Proceedings of the 21st IEEE International Conference on Network Protocols (ICNP 2013), Gottingen, Germany, October 2013, pp. 1–10.