Securing Software Defined Wireless Networks

Daojing He, Sammy Chan, and Mohsen Guizani

Similar to its wired counterpart, SDWN is expected to introduce a wide range of benefits to the operation and management of wireless networks. Security is always important to any network. On one hand, SDWN enables new security mechanisms. On the other hand, some new threats are introduced due to the separation of the control and data planes and the consequent introduction of the logically centralized controller.

ABSTRACT

Software defined wireless networking (SDWN) is a new paradigm of wireless networking, physically separating the data and control planes of various elements in the wireless infrastructure. Similar to its wired counterpart, SDWN is expected to introduce a wide range of benefits to the operation and management of wireless networks. Security is always important to any network. On one hand, SDWN enables new security mechanisms. On the other hand, some new threats are introduced due to the separation of the control and data planes and the consequent introduction of the logically centralized controller. In this article, we discuss its security threat vectors as well as design issues in making it secure. Also, we analyze the security requirements of SDWN, and then summarize the security attacks and countermeasures in this area and suggest some future research directions.

INTRODUCTION

The protocol architectures of computer networks or telecommunications networks generally consist of a control plane and a data plane. The control plane manages the configuration of networking devices (i.e., switches or routers) and their forwarding functions. The data plane consists of protocols to execute the forwarding functions according to the rules configured by the control plane protocols. Traditionally, as shown in Fig. 1a, both control and data planes are implemented in each networking device. As a result, whenever device configurations or routing strategies need to be changed, the firmware of all involved networking devices have to be modified. This means high labor cost and long delay, which increase with the network size.

Software defined networking (SDN), as shown in Fig. 1b, is a new and promising networking paradigm in which the control and data planes are decoupled, network intelligence is logically centralized, and the underlying network infrastructure is abstracted from the applications [1]. It provides great advantages in simplifying network management such that network administrators have central programmable control of network traffic via controllers, and new functions can easily be supported without physical access to the networking devices. That is, SDN is an efficient technology capable of supporting the dynamic nature of future network functions and intelligent applications while lowering operating costs through simplified hardware, software, and management.

At the same time, mobile networks become more convergent as various wireless technologies, such as Long Term Evolution (LTE), WiMAX, and WiFi, are integrated into the network infrastructure. Such an infrastructure typically comprises networking devices from different vendors and involves multiple operators. Managing the interoperability of these devices with different configurations for different policies and security requirements imposes a challenge. Moreover, while mobile users roam between different networks managed by different operators, guaranteeing consistent security across multiple domains dynamically and efficiently adds complexity to network management. With its virtualized abstraction and programmability features, SDN can hide the complexity of wireless protocols and support granular policy control. Thus, it is natural to apply the SDN paradigm to wireless mobile networks, leading to software defined wireless networking (SDWN). It is expected that SDWN will also bring the benefits of cost-effective infrastructure upgrade, delivery of new service, and improvement of user experience to existing infrastructure. The conceptual architecture of SDWN is depicted in Fig. 2. Research work in SDWN is emerging, including SDN design for the cellular core infrastructure [2], supporting fine-grained policies in cellular networks through scalable architecture design [3], abstraction of multiple base stations into a single virtual big base station [4], and decoupling of protocol definition from the hardware and provision of the software abstraction layer to enable programmable MAC and physical layers [5].

From a security point of view, SDWN has both advantages and disadvantages. As for one advantage, it enhances network security with its capability of redirecting or filtering traffic flows based on packet contents or network states. Such functions normally require additional security modules in traditional networks (e.g., firewalls or intrusion detection systems). But they can be naturally supported in SDWN, just as in the case of SDN [6]. On the other hand, due to physical separation of the control and data planes, a disadvantage is that SDWN is vulnerable to more attack vectors than traditional network architectures. This means that the availability, authen-

Daojing He is with East China Normal University; Sammy Chan is with City University of Hong Kong; Mohsen Guizani is with the University of Idaho.

0163-6804/16/$25.00 © 2016 IEEE

ticity, confidentiality, consistency, and integrity of network and control traffic could be severely affected. Obviously, these security issues have to be addressed before SDWNs are adopted in production networks.

This article focuses on security issues of SDWNs. It is structured as follows. The next section describes the security threats to SDWN, which are inherited from SDN. We present some specific design issues of securing SDWN. We discuss the security requirements of SDWN. Then some possible attacks to SDWNs and their countermeasures are discussed. Subsequently, some future research directions are provided.

Security Threats and Challenges of SDWNs

In traditional networks, forwarding devices are distributed in different geographical locations. If an attack to multiple forwarding devices is made, it needs to be carried out in a cooperative manner; thus, launching such an attack is not straightforward. On the other hand, even though SDWNs bring the benefits of network programmability and logically centralized control, it is exactly these benefits that expose SDWNs to new threats or those treats that are harder to exploit in traditional networks. For example, just a single attack on the controller unit can compromise the entire network.

As shown in Fig. 3, a vector of threats to SDWNs have been identified in [1]. They are tabulated in Table 1 and described in this section.

1. Forged or faked traffic flows: Both forwarding devices and controllers are vulnerable to this attack. Either a non-malicious faulty device or an adversary could trigger this threat. An attacker can launch denial-of-service (DoS) attacks to exhaust the resources in forwarding devices and controllers. Certainly, this problem can be mitigated by an authentication mechanism. However, if the attacker has compromised an application server that holds the credentials of many users, it can easily inject forged flows, which are authorized, into the network.

2. Attacks on forwarding devices: Such attacks can easily devastate the network. One single forwarding device could be used to discard, slow down, or deviate network traffic. Even worse, forged requests could be injected to overload the controller.

3. Attacks on control plane communications: Such attacks can be used to generate DoS attacks or divert flows of network traffic for the purpose of data theft. Various weaknesses of the transport layer security/secure sockets layer (TLS/SSL) communications and the public key infrastructure have been reported [7]. As a result, the controller can be compromised. The security of those communications suffers from a single point of failure, which may be a self-signed certificate or a compromised certificate authority. For example, many implementations of SSL currently used in mission-critical systems suffer from man-in-the-middle attacks [8]. Moreover, the TLS/SSL model is not sufficient to establish trust between controllers and forwarding devices. Once an attacker has gained access to the control

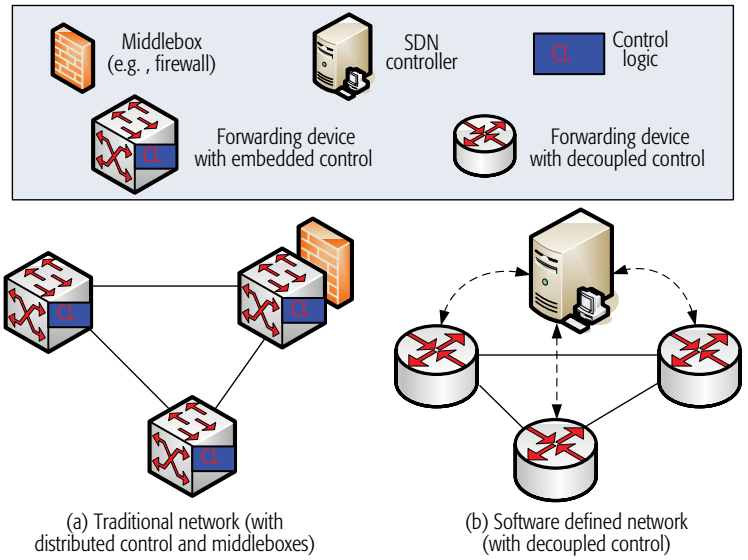

Figure 1. a) Distributed control plane in traditional networks; b) logically centralized control plane in SDN.

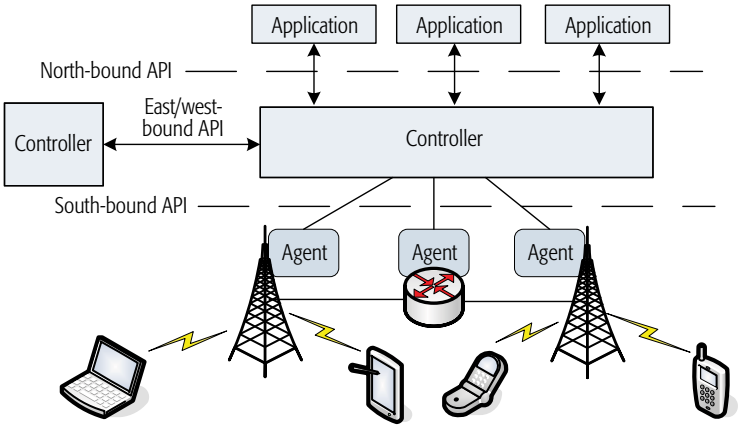

Figure 2. The conceptual architecture of SDWN.

plane, it may be able to launch distributed DoS attacks by aggregating the resources of forwarding devices under its control.

4. Attacks on controllers: These could probably be the most severe threats to SDWNs. A malicious or faulty controller could compromise the entire network. Since it could be difficult to identify the exact combination of events that cause a particular malicious behavior, commonly used intrusion detection systems may not be applicable. Similarly, a malicious application can virtually do anything to the network, as the controller only provides abstractions that are used to issue configuration commands to the underlying infrastructure.

5. Lack of trust mechanisms between the controller and management applications: This is similar to threat 3 because trusted relationships cannot be established between applications and controllers. The major difference is how the certification is done since the techniques for certifying forwarding devices are different from those for applications.

6. Attacks on administrative stations: These

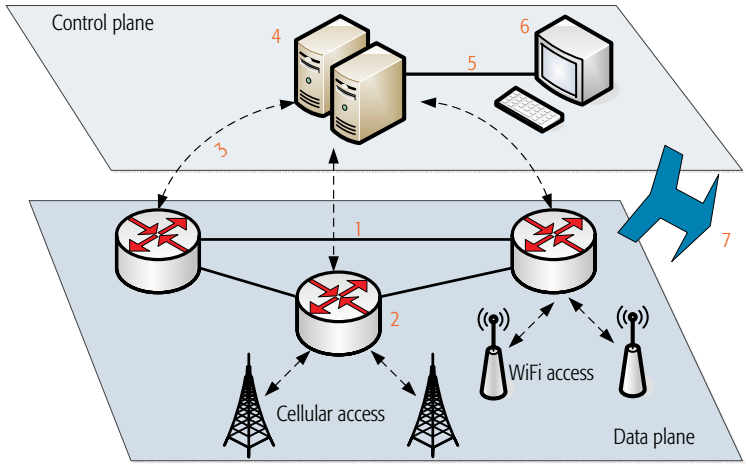

Figure 3. A vector of threats to SDWNs.

machines are used in SDN to access the controller. In fact, they are already exploitable targets in traditional networks. For the case of SDN, the threat surface as seen from a compromised machine is even larger. For example, reprogramming the entire network from a single location becomes much easier.

7. Lack of trusted resources for forensics and remediation: Such resources help to understand the cause of a detected problem and carry out subsequent recovery to secure mode. Without reliable information from various components and domains of the network, it is difficult, if not impossible, to investigate and establish facts about incidents in question. Moreover, such information is useful only if it is trustworthy. Similarly, for remediation, reliable system snapshots are required for recovering network elements to a known working state quickly and correctly.

More Issues for Securing SDWN

Besides handling the above threats, the following issues also need to be taken into account when designing mechanisms to secure SDWN.

- **User Mobility**: Since users generally roam between networks using different access technologies, it becomes more difficult to detect anomalous activities and exchange security credentials.
- **Multiple Operators**: In a typical SDWN network, multiple operators are involved. This complicates the negotiation process between networks, raises privacy issuesm and possibly causes policy and quality of service (QoS) requirements conflicts. Also, interoperability needs to be supported.
- **Overhead**: Monitoring functions such as that provided by OpenFlow incur high overhead and have the weakness of incomplete sample information.
- **Compatibility**: Since different generations of mobile technologies are likely to be deployed at the same time by operators, any proposed security solutions should be backward compatible.

Security Requirements and Solutions

Although SDWN is a new networking paradigm, the standard network security requirements (i.e., confidentiality, integrity, availability, authenticity, authorization, non-repudiation, and consistency) are still applicable. At the same time, there are new requirements due to the specific characteristics of SDWN. Here, we discuss the security requirements of SDWN, which are also listed in Table 2 [10].

Authenticity: Authenticity refers to the property that entities are actually the ones they claim to be. The issue of authenticity for forwarding devices in SDWN networks is similar to that in traditional networks. In both cases, the well established techniques for mutual authentication are applicable. Moreover, authentication and key establishment are closely related. Once two entities verify the authenticity of each other, they can establish some secret keys over the open wireless channel for subsequent secure information exchange. Traditional key generation and renewal algorithms should be adapted to take into account the characteristics of the wireless channel of SDWN. However, the authenticity of the controller and applications in SDWN is critical because the entire network can be compromised by a malicious controller or application. This issue is not relevant to traditional networks as they do not have the centralized controller and applications.

For SDWN, concerns have been raised about authentication and authorization mechanisms that simultaneously allow multiple entities to access network resources and provide appropriate protection of resources [11]. Generally, applications require different network privileges, thus a security model is needed to isolate applications and support network resources protection. Role-based authorization, such as FortNOX [12], is a potential solution. It resolves the situation in which a controller needs to handle conflicting flow rules from two different applications. However, role-based authorization alone is not sufficient to deal with the complexity of SDWN to isolate applications or resources. The controllers are particularly vulnerable to attacks in the SDWN architecture open to unauthorized access and exploitation. Moreover, without a robust and secure controller platform, an attacker could possibly masquerade as a controller to carry out malicious activities. These threats can be mitigated by existing security technologies such as TLS to enable mutual authentication between the controller and forwarding devices. However, the use of TLS in the current specifications of OpenFlow [13] is only optional. A full specification for secure interface between controller and forwarding devices is needed.

Confidentiality: Confidentiality prevents information disclosure to unauthorized third parties. Its impact on SDWN networks and traditional networks is similar. Many techniques for the two common methods to ensure confidentiality, encryption, and access control have been developed for traditional networks and could be adapted to SDWN networks. Encryption of the communication channel between the controller and a forwarding device means that an attacker cannot recover the original plaintext even though it has access to the cipher-text. For example, TLS can be used to establish an encrypted channel.

Access control ensures that only authorized entities have access to system data of the controller and forwarding devices. It may be enforced by the operating system.

Availability: Availability means that authorized users can access data, devices, and services whenever they have the need. In SDWN, if a forwarding device is not available due to either technical errors or DoS attacks, it can be mitigated by the controller to dynamically re-establish the network paths. Thus, similar to traditional networks, the issue of availability of forwarding devices is not as critical because network paths can be changed accordingly. However, non-availability of the controller in SDWN is critical. If the controller is unavailable due to a technical error, misconfiguration, or a DoS attack, the forwarding devices are only able to operate in a predefined way. For DoS attacks, possible solutions are rate-limiting mechanisms, discarding DoS-attack packets [14], or redundant controllers [15].

Integrity: It ensures that information has not been modified by any adversary. In SDWN, mainly the integrity of flow rules and protocol messages exchanged between the layers need to be ensured. The message authentication code is a commonly used approach to ensure integrity. The issue of flow rule integrity is critical in both SDWN networks and traditional networks since undesirable effects are caused by modified rules.

Consistency: This is about network traffic and control data. Generally, multiple applications could define flow rules and, as a result, could be inconsistent. One possible solution is to deploy a mediator between the controller and applications to resolve conflicting rules. FortNOX [12] is an example of such a mediator. This issue is critical in both traditional networks and SDWN networks because it can cause unpredictable behavior in both types of networks.

Fast Responsiveness: No matter whether security events are processed in reactive or proactive ways, it should be done in a timely fashion. This may involve efficient triggering and local optimization.

Adaptation: To take into account user mobility and dynamic network conditions, SDWN should be made adaptive by using mechanisms such as monitoring tools for network and user activities.

Attacks and Countermeasures on SDWNs

SDWNs are subject to a variety of security attacks such as spoofing, tampering, repudiation, information disclosure, DoS, and elevation of privileges. Table 3 maps attacks to the properties that guard against them. In the following, we give a glimpse of recent developments of countermeasures to attacks by focusing on solutions to information disclosure and DoS attacks.

Information Disclosure: The objective of this attack is to exploit the use of flow aggregation to extract some network state information. An attacker can use such information to determine the nature and presence of services on a network, which may be useful in a later stage of an attack. Approaches to mitigate this attack should aim to prevent the internal system states from being disclosed in the observable system parameters. The following approaches are potential candidates.

Num	Threats	Consequences in SDWN
1	Forged or faked traffic flows	Can be a door for DoS attacks.
2	Attacks on forwarding devices	The impact is potentially augmented.
3	Attacks on control plane communications	Communication with logically centralized controllers can be explored.
4	Attacks on controllers	Controlling the controller may compromise the entire network.
5	Lack of trust mechanisms between the controller and management applications	Malicious applications can now easily be developed and deployed on controllers.
6	Attacks on administrative stations	Now the impact is potentially augmented.
7	Lack of trusted resources for forensics and remediation	It is still critical to ensure fast recovery and diagnosis when faults happen.

Table 1. Threats inherited from SDN.

Property	Description
Confidentiality	To prevent information disclosure to unauthorized third parties.
Integrity	To ensure that information is not modified by any adversary.
Availability	To ensure that authorized users can access data, devices, and services whenever they have the need.
Authenticity	Entities are ensured to actually be the ones they claim to be.
Authorization	Only legitimate users can access resources.
Nonrepudiation	Users cannot deny any action that they have performed.
Consistency	To ensure that flow rules defined by different applications have no conflict.
Fast responsiveness	Security events should be processed in a timely fashion.
Adaptation	To take into account user mobility and dynamic network conditions.

Table 2. Security requirements.

1. Proactive Strategies: The establishment of proactive flow rules make the response time independent of the network states. Of course, this situation may be worsened by automatic flow aggregation techniques as an attacker might infer the presence of another connection, which is aggregated with its current one.

2. Randomization: The statistical uncertainty of an attacker can be increased while the strength of the attack can be reduced by increasing the variance of response times. For example, in OpenFlow, timeouts of the installed flow rules can be randomized to introduce unpredictable behavior. This prevents an attacker from having a coherent view of network states.

3. Attack Detection: Attacks based on timing analysis would exhibit distinctive and repetitive patterns. They could be exploited by controller applications to detect attacks and trigger countermeasures. Possible countermeasures include dropping suspicious traffic or adapting the forwarding strategies accordingly.

Denial of Service: DoS attacks can target forwarding devices and the controller, aiming to drain their resources so that the intended ser-

In SDWN, the controller sends policies to forwarding devices to instruct them how to deal with flows. Also, some policies may have to be sent to multiple forwarding devices. It is very important to keep these policies authentic and confidential because the wireless channel is insecure.

Attack	Security property
Spoofing	Authentication
Tampering	Integrity
Repudiation	Non-repudiation
Information disclosure	Confidentiality
Denial of service	Availability
Elevation of privilege	Authorization

Table 3. Attacks and security properties.

vices are not available. One example is to send a large number of requests to the controller to install new flow rules in forwarding devices, leading to flow table overflow. The following are potential approaches to mitigate the attacks.

1. Packet Dropping and Timeout Adjustment: If attackers can be detected, flow rules can be installed to identify malicious traffic and drop the misbehaving packets. On the other hand, if it is not possible to detect attackers, traffic prioritization and QoS mechanisms can be deployed to cope with the load. Moreover, flow timeouts can be adjusted to reduce the impact of DoS attacks.

2. Flow Aggregation: With this proactive strategy, multiple network flows are matched to a flow rule; the number of flow rules required to match traffic is thus reduced. It has the advantage that flow tables are less prone to overflows, and there is less load on the controller. Aggregated flow rules are suitable for networks such as backbone carriers, which deploy proactive strategies, but may not be applicable to enterprise networks, which deploy fine-grained control.

3. Access Control: This approach enforces access control lists stored as flow rules in forwarding devices. For example, traffic originating from a trusted domain is allowed to pass, while other incoming traffic is compared against a set of flow rules representing whitelists. This approach is particularly suitable for corporate networks, which have traffic coming from internal users or trusted external hosts.

Future Directions

SDWN influences defense against DoS attacks in negative ways. The middle-boxes or devices distributed within traditional networks are now located on top of the controller. Compared to packet processing based on hardware, it is much slower to process packets in software. The traffic overhead and network delay caused by the communications between the defense mechanisms and the forwarding devices could be the new attack surface. Thus, in the design of defense against DoS attacks, the computation and communication overhead must be taken into account in order to avoid introducing new security vulnerabilities.

The centralized and fine-grained control that comes with SDWN introduces a greater risk of outages due to errors made by network administrators. Misconfiguring of a controller by an administrator could significantly degrade network performance, even if the controller is functioning properly and the forwarding devices have no problematic flow rules installed. Detecting degraded network performance is challenging and needs further research. For example, even if Byzantine controller failures are assumed, it is difficult to determine what constitutes a fault for a properly functioning but misconfigured controller.

The threats to confidentiality, authenticity, integrity, availability, and consistency discussed in earlier sections are not totally new. Future work can leverage on established solutions to tackle these threats. Especially, there is increasing demand in the integration of public key infrastructures into SDWN in order to protect communications between different components of SDWN networks and to ensure authenticity of components.

Since a compromised controller can take control of the whole network, one simple attack is to suspend the functions of the controller. To deal such an attack, we can deploy multiple controllers in the network. Unfortunately, there are other possible attacks, such as issuing malicious commands from a compromised controller. Deploying multiple controllers would open more possible points of attacks. Therefore, more effective and secure mechanisms to protect controllers are needed.

In SDWN, the controller sends policies to forwarding devices to instruct them how to deal with flows. Also, some policies may have to be sent to multiple forwarding devices. It is very important to keep these policies authentic and confidential because the wireless channel is insecure. Work on designing policy distribution schemes that ensure the polices to be authentic and confidential simultaneously is needed.

Conclusion

SDWN, resulting from the extension of the SDN concept into wireless networks, will enjoy the benefits of cost-effective infrastructure upgrade, delivery of new services, and improvement of user experience to existing infrastructure. Similar to SDN, SDWN is vulnerable to new attacks due to physical separation of the control plane and data plane. Research work to address these issues have just commenced. We hope this article stimulates development of effective defense solutions to make SDWN attack-resilient.

Acknowledgment

This research is supported by a strategic research grant from City University of Hong Kong [Project No. 7004429], the Pearl River Nova Program of Guangzhou (No. 2014J2200051), the National Science Foundation of China (Grants: 51477056 and 61321064), the Shanghai Knowledge Service Platform for Trustworthy Internet of Things (No. ZF1213), and the Specialized Research Fund for the Doctoral Program of Higher Education. Daojing He is the corresponding author of this article.

References

[1] D. Kreutz *et al.*, "Software-Defined Networking: A Comprehensive Survey," *Proc. IEEE*, vol. 103, no. 1, 2015, pp. 14–76.

[2] L. Li, Z. Mao and J. Rexford, "Toward Software-Defined Cellular Networks," *IEEE EWSDN*, Oct. 2012, pp. 7–12.
[3] X. Jin *et al.*, "SoftCell: Scalable and Flexible Cellular Core Network Architecture," *Proc. ACM CoNEXT*, 2013, pp. 163–74.
[4] A. Gudipati *et al.*, "SoftRAN: Software Defined Radio Access Network," *Proc. ACM HotSDN*, 2013, pp. 25–30.
[5] M. Bansal *et al.*, "OpenRadio: A Programmable Wireless Dataplane," *Proc. ACM HotSDN*, 2012, pp. 109–14.
[6] S. Ali *et al.*, "A Survey of securing Networks Using Software Defined Networking," *IEEE Trans. Reliability*, vol. 64, no. 3, Sept. 2015, pp. 1086–97.
[7] R. Holz *et al.*, "X.509 Forensics: Detecting and Localising the SSL/TLS Men-in-the-Middle," *Proc. ESORICS*, 2012, pp. 217–34.
[8] M. Georgiev *et al.*, "The Most Dangerous Code in the World: Validating SSL Certificates in Non-Browser Software," *Proc. ACM CCS*, 2012, pp. 38–49.
[9] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A Survey of Security in Software Defined Networks," *IEEE Commun. Surveys & Tutorials*, DOI: 10.1109/COMST.2015.2453114, 2015.
[10] A. Y. Ding *et al.*, "Software Defined Networking for Security Enhancement in Wireless Mobile Networks," *Computer Networks*, vol. 66, June 2014, pp. 94–101.
[11] L. Schehlmann, S. Abt, and H. Baier, "Blessing or Curse? Revisiting Security Aspects of Software-Defined Networking," *Proc. CNSM*, 2014, pp. 382–87.
[12] P. Porras *et al.*, "A Security Enforcement Kernel for OpenFlow Networks," *Proc. ACM HotSDN*, Aug. 2012, pp. 121–26.
[13] ONF, "Software-Defined Networking: The New Norm for Networks," ONF White Paper, Apr. 2012.
[14] R. Kloti, V. Kotronis, and P. Smith, "OpenFlow: A Security Analysis," *Proc. IEEE ICNP*, Oct. 2013, pp. 1–6.
[15] B. Heller, R. Sherwood, and N. McKeown, "The Controller Placement Problem," *Proc. HotSDN*, Aug. 2012, pp. 7–12.

Biographies

Daojing He [S'07, M'13] (djhe@sei.ecnu.edu.cn) received his B.Eng.(2007) and M. Eng. (2009) degrees from Harbin Institute of Technology, China, and his Ph.D. degree (2012) from Zhejiang University, China, all in computer science. He is currently a professor in the Software Engineering Institute, East China Normal University, P.R. China. His research interests include network and systems security. He is an Associate Editor or on the Editorial Board of some international journals such as *IEEE Communications Magazine* and *IEEE/KICS Journal of Communications and Networks*.

Sammy Chan [S'87, M'89] (eeschan@cityu.edu.hk) received his B.E. and M.Eng.Sc. degrees in electrical engineering from the University of Melbourne, Australia, in 1988 and 1990, respectively, and a Ph.D. degree in communication engineering from the Royal Melbourne Institute of Technology, Australia, in 1995. From 1989 to 1994 he was with Telecom Australia Research Laboratories, first as a research engineer, and between 1992 and 1994 as a senior research engineer and project leader. Since December 1994 he has been with the Department of Electronic Engineering, City University of Hong Kong, where he is currently an associate professor.

Mohsen Guizani [S'85, M'89, SM'99, F'09] (mguizani@ieee.org) received his B.S. (with distinction) and M.S. degrees in electrical engineering, and M.S. and Ph.D. degrees in computer engineering from Syracuse University, New York, in 1984, 1986, 1987, and 1990, respectively. He is currently a professor and chair of the Electrical and Computer Engineering Department at the University of Idaho. He was previously a professor and associate vice president for graduate studies at Qatar University, Doha, Qatar. His research interests include computer networks, wireless communications and mobile computing, and optical networking. He currently serves on the Editorial Boards of six technical journals and is the Founder and EIC of the *Wireless Communications and Mobile Computing Journal* (Wiley; http://www.interscience.wiley.com/jpages/1530-8669/). He is a Senior Member of ACM.