Towards Secure Software-defined Networking Integrated Cyber-Physical Systems: Attacks and Countermeasures

Uttam Ghosh[1], Pushpita Chatterjee[2], Sachin S. Shetty[2], Charles Kamhoua[3], and Laurent Njilla[4]

[1] Vanderbilt University, Nashville, TN, USA
[2] Old Dominion University, Suffolk, VA, USA
[3] Army Research Lab, Adelphi, MD, USA
[4] Air Force Research Lab, Rome, NY, USA

[1] uttam.ghosh@vanderbilt.edu, [2]pchatter@odu.edu, [2]sshetty@odu.edu, [3]charles.a.kamhoua.civ@mail.mil, and [4]laurent.njilla@us.af.mil

Abstract: Cyber-physical system (CPS) refers to the next generation of engineered system that requires tight integration of cyber world and man-made physical world to achieve stability, security, reliability, robustness, and efficiency in the system. Emerging software-defined networking (SDN) can be integrated as the communication infrastructure with the critical physical infrastructure like smart power grid to accomplish such system. This Chapter gives an overview of SDN, smart grid and SDN-based smart grid. SDN can provide security against various types of attacks by providing consistent access control, applying efficient and effective security policies, and managing and controlling the network through the use of a centralized SDN controller. Thus, it has security advantages due to its design nature. However, SDN does suffer from security shortcomings too as all the layers in SDN architecture are vulnerable to attacks. In line of this, the Chapter discusses various types of attacks related to SDN architecture and their countermeasures. It also demonstrates the applicability of SDN to provide security in smart grids. Finally, the Chapter categorizes different types of attacks and their countermeasures related to SDN based smart grids. Keywords: CPS, SDN, Smart Grids, OpenFlow, SDN-based Smart Grids.

Keywords: CPS, SDN, Smart Grids, OpenFlow, SDN-based Smart Grids.

1 INTRODUCTION

Cyber-physical system (CPS) is an integration of cyber world (computation and communication systems) and man-made physical world (e.g., utility networks, vehicles, factories, etc.) as shown in Figure 1. CPS links between the physical world and cyber world by using sensors and actuators [1]. Sensors are used to measure physical quantities and convert them into an electrical signal. This electrical signal is sampled and quantized later for computing. The cyber system calculates based on these values and sends feedback to the physical world by actuators that convert electrical signals into a physical action.

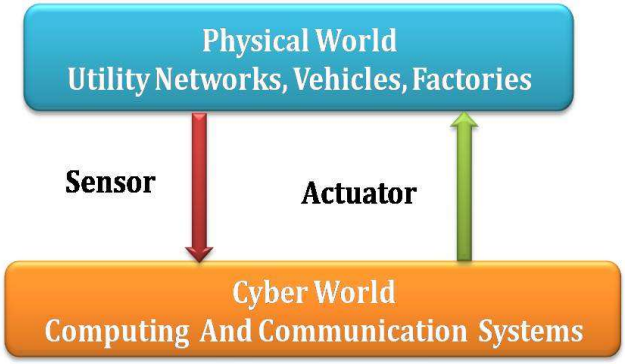

Figure 1: Cyber-physical system using sensors and actuators.

CPS combines digital and analog devices, computer systems and networks with the natural and man-made physical world. These man-made physical world includes buildings (homes, schools, offices, factories, etc.), utility networks (electricity, gas, water, etc.), transportation networks (roads, railways, airports, harbors, etc.), transportation vehicles (cars, rails, planes, etc.) healthcare systems, information technology networks, and so on. Theses physical infrastructures are integrated with cyber systems and broadly categorized them into social, medical, physical and enterprise systems as depicted in Figure 2. Cyber systems make the physical infrastructures more smart, secure and reliable, and fully automated systems.

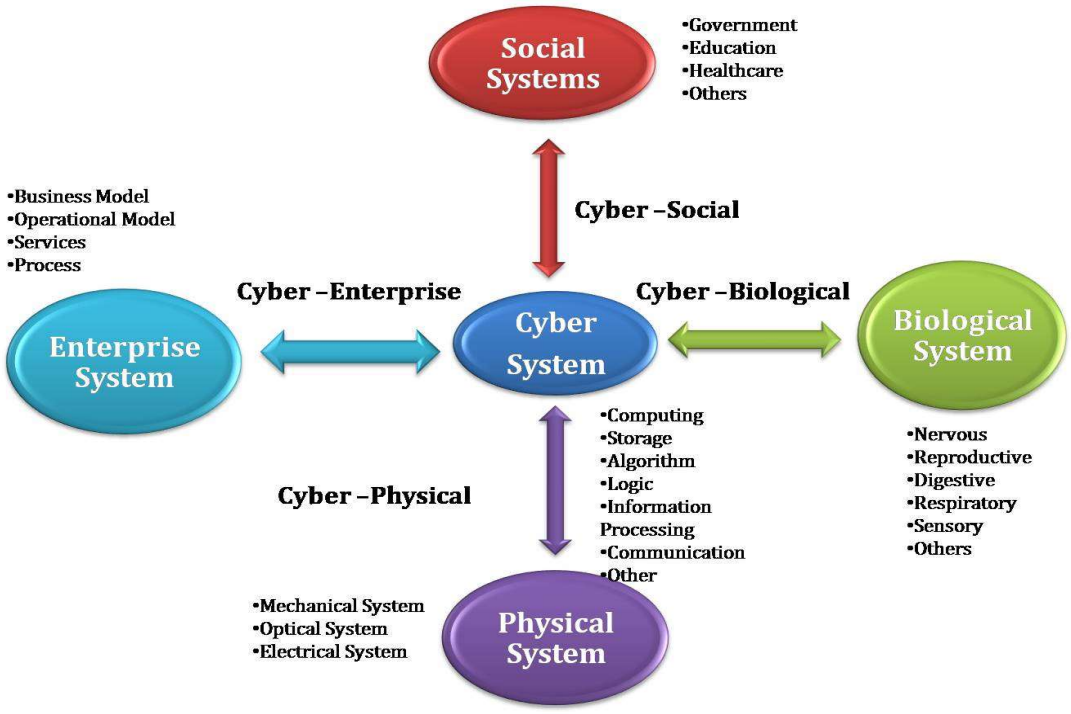

Figure 2: Categorization of Cyber systems based on application domains

Emerging software defined networking (SDN) paradigm provides flexibility to program the network centrally (logically) in controlling, managing, and dynamically reconfiguring the network. It decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. SDN mainly provides security and network virtualization for enhancing the overall network performance. SDN provides protection against various types of attacks by providing consistent access control, applying efficient and effective security policies, and managing and controlling the network through the use of a centralized SDN controller [2], [3].

In CPS, the deployed sensors generate a massive amount of real-time data from the physical infrastructure and send to the cyber systems using the communication infrastructure (such as switches, routers, etc.). The cyber systems also send feedback to the physical devices using the communication infrastructure. Thus, the communication infrastructure in CPS must be scalable, reliable, secure and efficient. Software defined networking (SDN) can be integrated with the physical infrastructure to achieve such communication infrastructure. SDN can manage and verify the correctness of the network operations at run time. The globalized view of SDN controller allows control, configure, monitor and also fault (due to accidental failures and malicious attacks) detection and remediates of abnormal operation in the SDN-based cyber-physical systems more efficiently as compared to the traditional based networks. This Chapter considers SDN as the cyber system and smart grid as the physical system.

It is becoming increasingly important to develop security and privacy mechanisms in cyber-physical system (CPS) applications. The security mechanisms are required to mitigate negative implications associated with cyber-attacks and privacy issues in the CPS. This Chapter aims to provide latest research developments and results in the areas of security and privacy for SDN-enabled smart grid networks. It presents insights into attacks in networking and security related architectures, designs, models for SDN enabled smart grid networks. In an effort to anticipate the future evolution of this new paradigm, present Chapter discusses the main on-going research efforts, frameworks, challenges, and research trends in this area. With this Chapter, readers can have a more thorough understanding of SDN architecture, different security attacks and their countermeasures in SDN and SDN-enabled CPS environments.

Chapter Organization: The rest of the Chapter is organized as follows: Section 2 presents a background of SDN, smart power grids and SDN-based smart grids. The attacks and countermeasures in SDN and SDN-based smart grids are presented in Section 3 and Section 4 respectively. Finally, Section 5 concludes the Chapter.

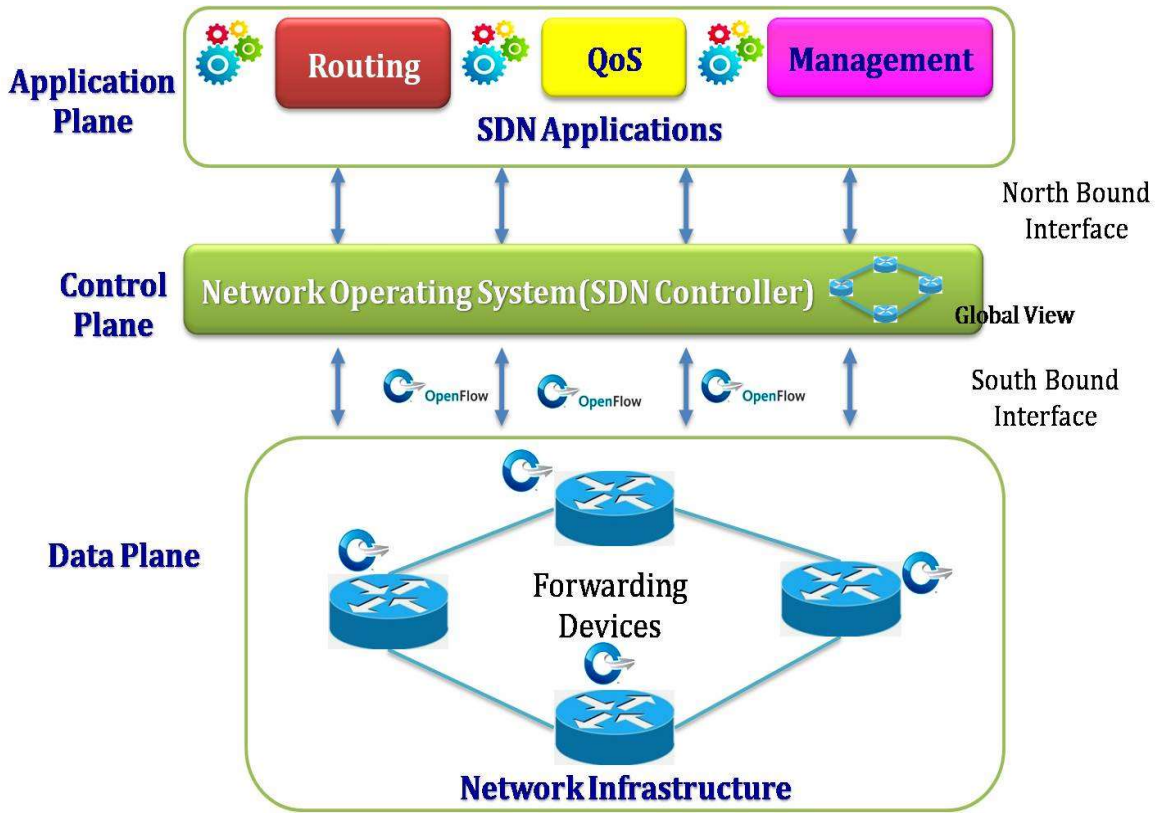

Figure 3: An Architecture of SDN

2 BACKGROUND

2.1 An Overview of SDN

This Section gives an overview of SDN architecture and working principles. It further discusses the importance of SDN and how SDN differs to the traditional networking.

2.1.1 SDN Architecture

Figure 3 presents major elements and interfaces of the SDN architecture. The SDN architecture has three layers: infrastructure layer, control layer and application layer.

2.1.1.1 Infrastructure Layer

The infrastructure layer (also known as data plane) consists of a set of one or more traffic forwarding devices. These forwarding devices are known as OpenFlow (OF) switches and responsible for forwarding the data from source to destination in a SDN network based on instructions (flow rules) received from control layer.

2.1.1.2 Control layer

The control plane consists a set of SDN controllers. The SDN controller, also known as network operating system, is a logical entity (software programs) that receives instructions or requirements from the SDN application layer and relays them to the OF switches of infrastructure layer. The controller keeps track of the network topology (global view of the network) and the statistics of the network traffic periodically. Thus, the controller is responsible for providing routing, traffic engineering or quality of services (QoS), load balancing and also security in the network.

2.1.1.3 Application Layer

The application layer comprises of one or more applications (software programs) and controls the network resources with the SDN controller through the use of application programming interface (APIs). It collects information from the controller periodically for decision-making purposes. These applications provide routing, quality of services (QoS) and network management. This layer further provides an interface to the network administrator for developing several applications according to the requirements of the network. For instance, an application can be built to monitor the network traffic and behavior of the nodes periodically for detecting attacker nodes in SDN network.

Northbound API defines the communication between application layer and control layer whereas southbound API defines the connection between the control layer and infrastructure layer. OpenFlow protocol has been used as an southbound API. The SDN controller sends flow rules into the OpenFlow switches using OpenFlow protocol for delivering data from source to destination. OpenFlow protocol uses secure socket layer and TCP to provide security and reliability respectively.

2.1.2 Working Principles of SDN:

Figure 4 presents working principle of SDN. Here a node H2 (source) sends the packets of a flow to another node H1 (destination) in SDN using the following operations [4] :

(1) H2 sends the packet to SDN OpenFlow (OF) switch S3.
(2) On receiving the packet from H2, S3 checks in its flow table for a matching flow rule. If the flow rule exists then switch forwards the packet according to the flow rule towards H1. Otherwise, S3 forwards the packet (i.e., first packet of the flow) to SDN controller.
(3) SDN controller on receiving the packet from S3 computes the shortest path between H2 and H1 for the new flow. In this example, the shortest path is H2-S3-S2-S1-H1. In response, SDN controller writes flow rules on all switches S3, S2 and S1 on the path. It may be noted here that SDN controller has the global view and traffic statistics of the network. Hence, SDN controller can choose different metrics instead shortest path for providing secure routing and quality of services.
(4) OF switch S3 forwards the packet to S2, S2 forwards the packet to S1, and S1 finally forwards the packet to the actual destination H1 as per flow rule (received from SDN controller in previous step).
(5) Thereafter, H2 sends all the packets of the flow to H1 through OF switches S3-S2-S1.

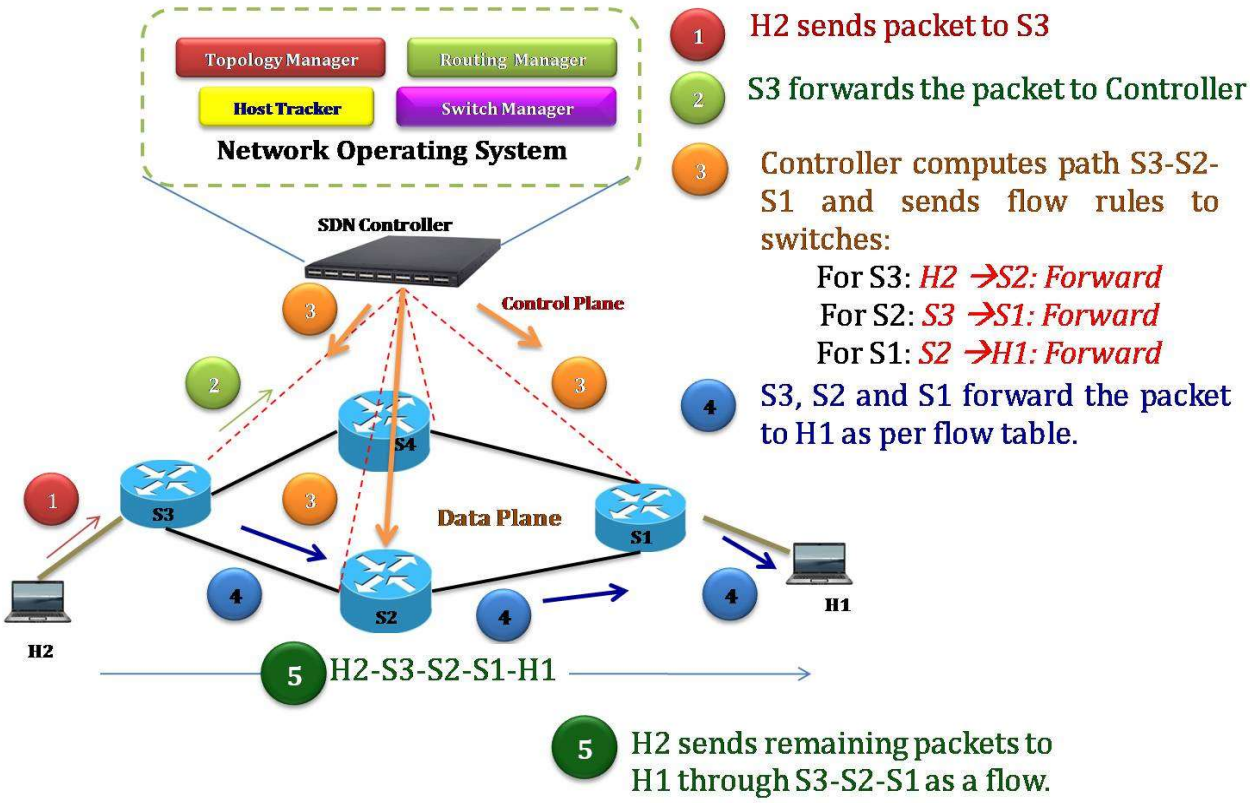

Figure 4: SDN Working Principles

2.1.3 Traditional Networking versus SDN

In traditional or legacy networks, the forwarding devices (switches) are complicated and vendor dependent. These switches are strongly coupled between control plane and data plane. Thus it is not easy to include new functionalities (applications) to the traditional networks, the fact is illustrated in Figure 5. The tight coupling of the control plane and data plane makes the development and deployment of new networking functionalities (e.g., routing, load balancing algorithms) very difficult. This is due to the fact that it needs a modification of the control plane of all the distributed switches in the network- through the installation of new firmware and, in some cases, up-gradation of hardware. These legacy switches are distributed in a large area that makes it even more difficult to later change the network topology, configuration, and functionality. In contrast, SDN decouples the control plane from switches of the data plane and becomes a separate entity: SDN controller or network operating system (NOS). It has several advantages over traditional networks:

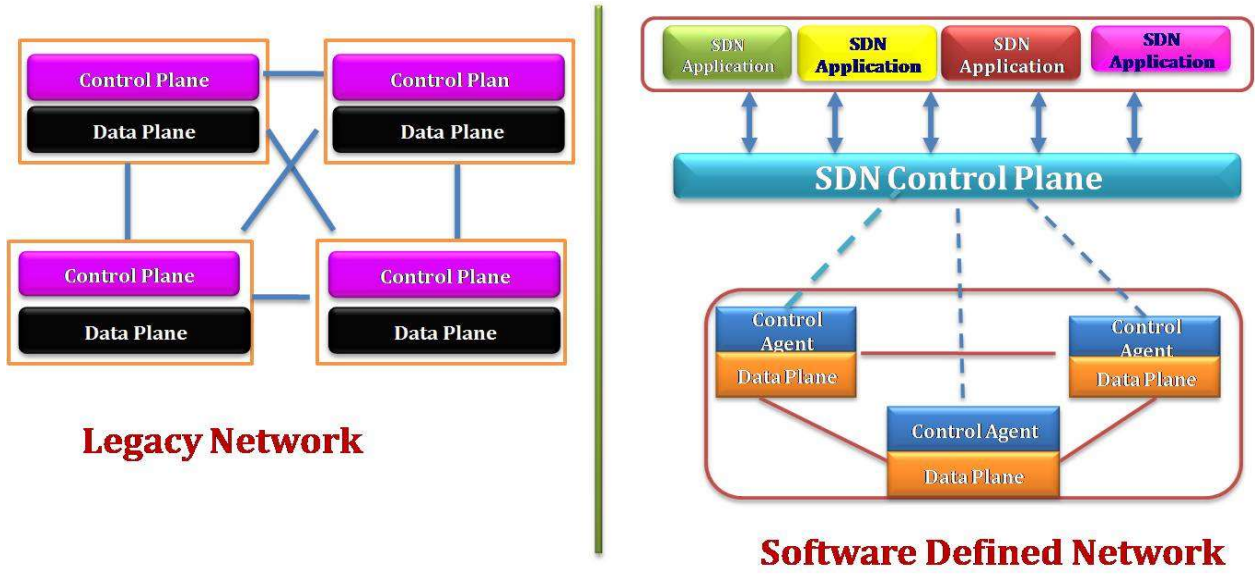

Figure 5: Traditional (legacy) networking versus Software-Defined Networking (SDN).

- SDN controller is programmable. It is easier to include new network functionalities through programs (as applications) at the top of SDN controller.
- SDN controller is logically centralized. It keeps track of the network topology and statistics of the network traffic periodically. The controller is consistent and effective in taking decisions for routing, QoS and load balancing dynamically.
- Logically central SDN controller can control, configure and monitor the distributed devices of the data plane.
- Logically central SDN controller can monitor all the devices and their traffic in data plane. The controller can run an application that can detect the malicious device whenever the device injects data falsely into the network or behaves abnormally. It can further eliminate the malicious device from the network on-fly by writing effective policies (flow rule) on the switches.

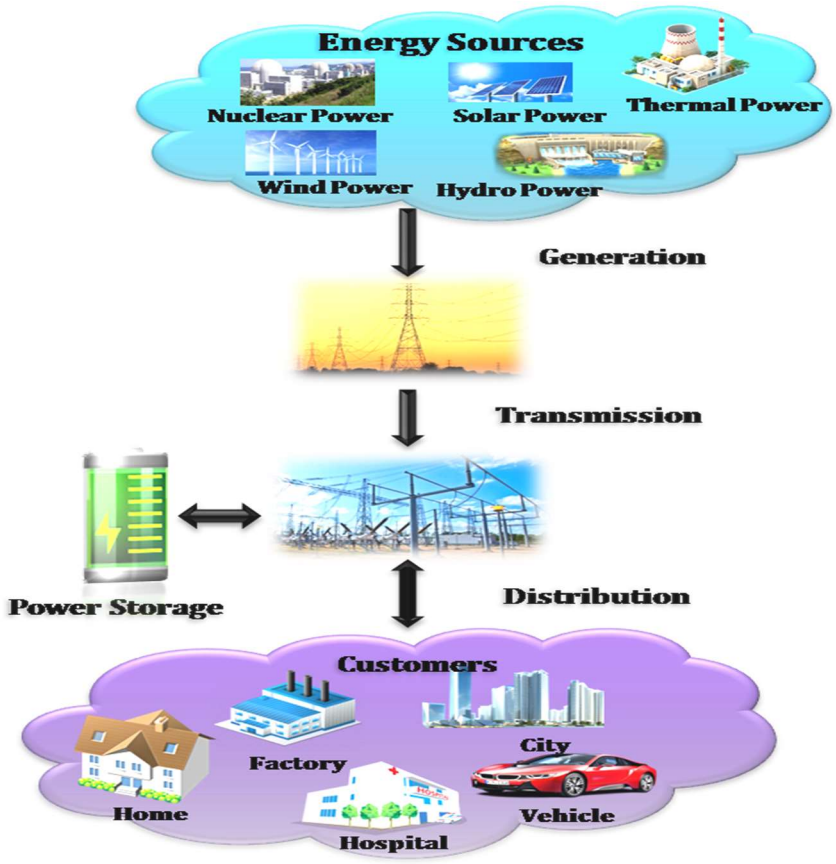

Figure 6: An Architecture of Smart Power Grid

2.2 An Overview of Smart Power Grid

A power grid refers to an electric grid which consists of generating stations that produce electrical power, high voltage transmission lines that carry power from a few centralized generators to demand centers, and distribution lines that connect a several number of individual customers. Smart grid is an up-gradation of existing power grid with intelligent smart communication and computing devices. It forms a large-scale heterogeneous complex network between a large number of sensors, actuators, control and data acquisition (SCADA) systems, and also smart meters located in residential and commercial premises. An architecture of smart grid is presented in Figure 6 where energy can be generated seamlessly from different power sources (such as wind, solar, nuclear) and transmitted using transmission lines to distribution center and finally distributed to the individual customers depending upon their requirements. The excessive power generated by smart grid can be stored for future use. Smart grid delivers high quality of electricity to the users with information of consumers about their electricity consumption in real time by using the smart devices (such as sensors, switches) to make flows of electricity and information two-way. It enhances the capacity and efficiency, and also improves the reliability, quality and resiliency to disruption of existing power grid networks. In summary, smart grid is an automated, self-healing and distributed advanced energy delivery network that provides the following features with power grid [4]:

- Supports two-way communication of electricity and information

- Provides interaction between users and the electricity market
- Monitors the power network manages electrical energy consumption in real time
- Optimizes the network and power resources
- Enables integration, monitoring, control, security and maintenance
- Provides security against attacks and threats

2.3 An Overview of SDN-based Smart Grid

An architecture of SDN-base smart grid is presented in Figure 7. The architecture mainly comprises of three segments: (i) a control center; (ii) communication network that consists of OpenFlow switches and links between them, and (iii) substations consist of physical power grid devices (known as SCADA slaves in general) with sensors and actuators. Control center runs the commodity computers and servers for the SDN controller and SCADA master. The SCADA master is responsible for controlling, configuring and managing the grid devices in substations, whereas the SDN controller is responsible for configuring and managing the networking devices. The SCADA master collects the measurement data periodically from the SCADA slaves (sensors) in the substations through the communication network. It estimates the state (stability) of the smart grid by processing the received data. Based on the estimated state of the smart grid, SCADA master then sends the control-command (such as read, write or execute) **[5]** back to the SCADA slaves (actuators) in the substations through the communication network. Table I summarizes the control-commands issued by the SDN controller and SCADA master in SDN-enabled smart grid networks in order to configure and manage the network and grid devices respectively **[2]**, **[5]**.

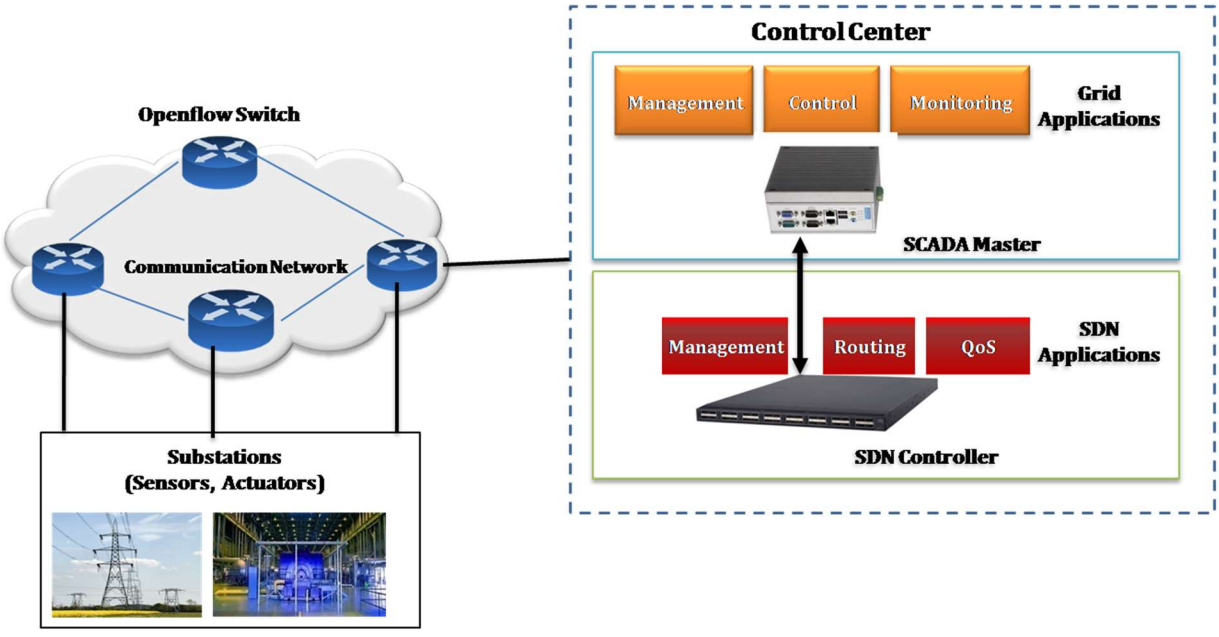

Figure 7: An Architecture of SDN-based Smart Grid

Table I: Control-commands in SDN-based smart power grid networks

Control-commands by SCADA master	Functionalities	Control-commands by SDN Controller	Functionalities
Read	Retrieve measurements from substations (SCADA slaves) by SCADA master	Add _Flow	Add a new flow rule to OF switches by SDN controller
Write	Configure smart grid devices by SCADA master	Del_Flow	Remove a flow from OF switches by SDN controller
Execute	Operate smart grid devices by SCADA master	Mod _Flow	Edit a flow in OF switches by SDN controller

Table II presents the comparison between smart grid and SDN-based grid networks. The communication infrastructure must be scalable, resilient, secure and efficient for the smart grid as it is a large-scale heterogeneous complex network that generates a massive amount of real-time data.SDN provides such communication infrastructure in SDN-based smart grids. SDN controller is programmable and it keeps track of the network traffic and topology periodically. Thus, it can be used for load balancing, for dynamically adjusting the routing paths for the control-commands **[6]**, fast failure detection **[7]**, security **[2]**, self-healing **[8]**, and also for monitoring and scheduling of critical traffic flows in smart grid networks. SDN controller can prioritize the traffic to increase the throughput and provide quality of service (QoS) in SDN-based smart grids. Moreover, it supports heterogeneous networks and does not depend on vendor and protocol and, operates open standard **[9]**.

Table II: Smart Grid versus SDN-based Smart Grid

Features	Smart Grids	SDN-based Smart Grid
Programmability	Smart grids are less programmable	SDN-based smart grids are programmable as SDN controller is programmable
Protocol independency	Smart grids depend on some specific protocols	SDN-based smart grids independent of protocol through the use of SDN controller
Granularity	Fully dependent on proprietary hardware	SDN-based smart grid networks are programmable and independent of proprietary hardware; SDN controller can identify the traffic at every flow and packet level and provide QoS

Resiliency	Limited resiliency to malicious attacks and failures	Resilient against malicious attacks and failures by using SDN
Interoperability	Smart grids depend on vendor dependent hardware and software (vendor-lock); Thus, it is difficult to configure and manage the smart grid with different vendor specific devices and protocols	SDN is not dependent on vendor and working on open standards; Thus, it is easy to configure and manage the smart grid with different vendor specific devices and protocols
Management of Network	Managing the smart grid network is complex and time consuming, and even sometimes manual	SDN controller is logically centralized and it can manage the smart grid network easily and automatically
Security	Several security schemes are proposed	SDN can provide security by using controller security policies; However, it needs to develop new security schemes as SDN controller may compromise or controller applications may get compromised

3 Attacks and Countermeasures in Software-Defined Networking

This Section presents the attacks and countermeasures in software-defined networking.

3.1 Attacks in SDN

This Chapter mainly focuses on the attacks to the application, control and infrastructure layers of SDN [10], [11]. Figure 8 presents the attacks that can be seen in SDN layers.

3.1.1 Attacks at Application Layer

In application layer, attackers can manipulate the network configuration, steal network information, seize network resources and so on through placing malicious computer programs (such as spyware, malware, virus) to the SDN applications. As described earlier, the Northbound API provides an interface between SDN controller and applications for managing and controlling the network . However, the lack of trust between the controller and SDN applications is a security concern as malicious SDN applications can send malicious commands to the network through the poorly designed Northbound API.

Resource Exhaustion: A malicious application can excessively use all available resources of the system that runs SDN controller. This may lead to degradation in the performance of SDN controller and other applications.

a) Memory Exhaustion: In order to exhaust the memory of the system where the controller is running, a malicious application can allocate memory continuously.

b) CPU Exhaustion: A malicious application can create working threads continuously to use up all available CPU resources. As a result of this, CPU may not be able to execute other applications.

Abuse of Control Message: A SDN application may arbitrarily issue control messages:

a) Flow Rule Modification: In the flow table of an OpenFlow switch, an existing flow rule can be overwritten by a malicious application to cause unexpected network behavior.

b) Flow Table Clearance: A malicious application may send the control messages to clear the flow table entries of a switch and may force to terminate all the communications in the network.

Service chain interference: Applications of SDN with chained execution may be interfered. For example, a malicious application can participate in a service chain and drop the control messages before the other SDN applications awaiting for them. Moreover, an interference can be seen when a malicious application falls in an infinite loop to stop the chained execution of applications.

Abuse of Northbound API: In SDN, the Northbound API defines the communication between control layer and application layer and also provides an interface to program the SDN network through the use of software. However, the Northbound API does not consider the security aspect while designed. As a result of this, a malicious application can manipulate the behavior of the other applications by abusing the Northbound API.

a) Event Listener Un-subscription: In order to make the target incapable of receiving any of the messages from other applications of SDN controller, a malicious application may arbitrarily unsubscribe the target application from the control message subscription list.

b) Application Eviction: A malicious application may force other applications of SDN to terminate arbitrarily.

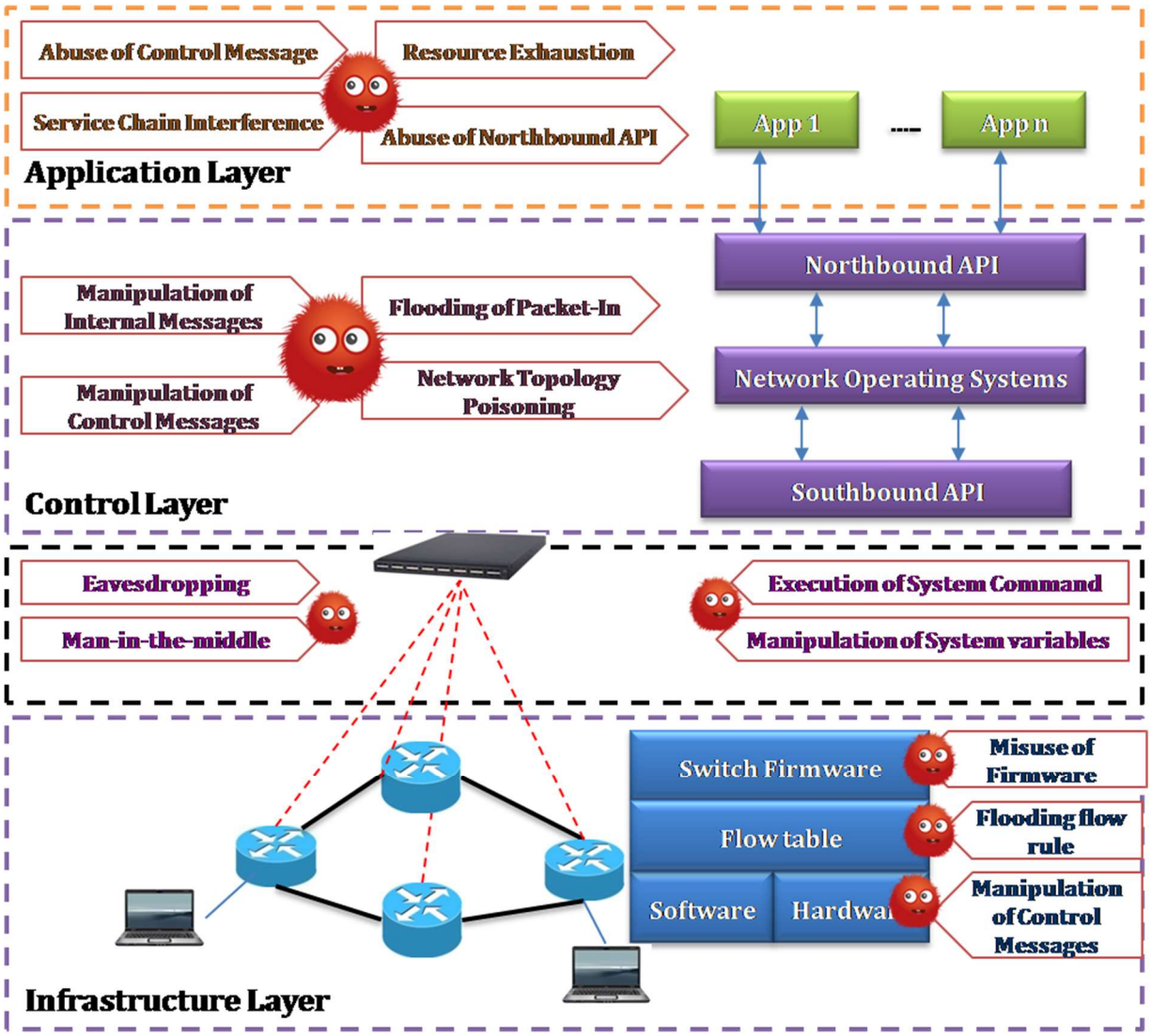

Figure 8: Attacks on SDN in different layers

3.1.2 Attacks at Control Layer

Attacks at the control layer has the most severe impact on SDN as they can even control and down the entire network. These attacks can be initiated from the system hosting the SDN controller by exploiting some software vulnerabilities in the system.

Network Topology Poisoning: An attacker can manipulate the network topology by exploiting the vulnerabilities exist in the Host Tracking Service and Link Discovery Service of various SDN controllers [12]:

a) Host Location Hijacking: Host tracking service (HTS) in SDN controller keeps track of locations of all hosts by monitoring Packet-In messages that received by the controller from the OpenFlow switches. For instance, if a host migrates to another location, HTS can detect such migration. However, HTS does not

verify authentication of hosts. As a result, an attacker can easily impersonate (spoof) a target host and subsequently hijack the network traffic.

b) Link Fabrication: SDN controller periodically sends LLDP packets to discover the links among OpenFlow switches at infrastructure layer. However, the link discovery procedure is vulnerable as an attacker can manipulate the network topology by sending a forged or relayed LLDP packet to the controller.

Flooding of Packet-In: A Packet-In essentially represents a packet that does not match any flow rules at infrastructure layer, and the OpenFlow protocol commands that such packets must be sent by the OpenFlow switch to the controller directly. The control layer has no built-in security mechanism to avoid the manipulation of packet-in messages even when the OpenFlow switches are enabled with transport layer security (TLS). For instance, an attacker can flood a several number Packet-In messages to place SDN controller in an unpredictable state.

Manipulation of Control Message: An attacker can manipulate the control messages to put the control layer of SDN in an unpredictable state:

a) Switch Table Flooding: An attacker can send a large number of forged control messages to flood the flow table of the OpenFlow switch. As a result of this, the switch may not be available and may cause to network partitions.

b) Switch Identification Spoofing: The switch identification field of a control message may be manipulated to poison in the network topology and subsequently put the control layer in an unpredictable state.

c) Malformed Control Message: An attacker can inject malformed control messages into the network and can cause malfunction of the control layer.

Manipulation of Internal Storage: The SDN controller shares internal storage among various SDN applications. Eventually, SDN applications can unrestrictedly access and manipulate the internal database of SDN controller. This internal database can further be misused for many subsequent attacks, such as manipulating the network topology.

Manipulation of System Variable: System variables may be manipulated to put SDN controller in an unpredictable state. For example, an attacker can change the timer of the hosting operating system to put SDN controller in an unpredictable state and disconnect from the OpenFlow switches of infrastructure layer.

Manipulation of System Command: In order to terminate the controller instance from the hosting operating system, a malicious application can execute a system exit command.

Eavesdropping: An attacker can sniff the control channel to steal sensitive information from SDN network. For example, an attacker sniffs the ongoing control messages on the control channel to learn the network topology.

Man-in-the-middle Attack: An attacker can actively intervene in the control channel. For example, an attacker modifies the flow rule message that is being transferred, and corrupt the behavior of the network.

3.1.3 Attacks at Infrastructure Layer

Infrastructure layer is located at the bottom of the SDN architecture, and contains several number of hosts and OpenFlow switches that are interconnected with each other. These switches are responsible for forwarding packets to the end host. An attacker can attack an OpenFlow switch by simply attaching a link to a port of the switch.

Misuse of Firmware: An attacker can misuse the characteristics of a certain switch model. For instance, the crafted flow rules installed by an attacker may not be processed in the hardware table of a certain switch model.

Flooding of Flow Rule: Infrastructure layers can be in an unpredictable state when a large number of flow rules are sent to an OpenFlow switch. An attacker can capture the control channel and install numerous flow rules to the target switch to fill up the flow table. As a result, the victim switch drops the flows for authorized hosts and leads to denial-of-service attack.

Manipulation of Control Message: A malformed control message may put the infrastructure layer in an unpredictable state: An attacker injects a malformed control message to the infrastructure layer to interrupt the connection between the control layer and the infrastructure layer.

3.2 Countermeasure of Attacks in Software-Defined Networking

It can be seen from the previous Section 3.1, the attackers can exploit all the layers in SDN due to its design issues. A number of techniques have been proposed to countermeasures [11] to the attacks as presented in Table III. In summary,

- Attacks on infrastructure layer due to limited authentication mechanisms. Message authentication code (MAC), digital signature or SSL/TLS can be used for authentication and encryption to avoid eavesdropping, sniffing and spoofing of traffic. Further, SDN controller can maintain a list of authorized switches in access control list (ACL) and run an IDS to mitigate attacks from infrastructure layer.
- Forged traffic flows, man-in-the-middle and false reply attacks as traffics are sent in clear text. In order mitigate these attacks, network traffic should be sent either with a message authentication code (MAC) or a digital signature. Timestamp can be included with the MAC or digital signature to avoid reply attacks. Further, SDN controller can ensure that security policy is implemented for all traffic flows.
- Attacks on control layer due to limited authentication mechanisms and open northbound and southbound APIs. Both northbound and southbound API can be protected by using cryptographic encryption (TLS or SSH). The control layer can replicate the controller periodically or can run multiple controllers to make the network more reliable and secure.
- Denial of service (DoS) or flooding attacks can be seen in any layers of SDN architecture. These attacks are possible due to OpenFlow switche have limited memory and flow table, and SDN controller is centralized. In order to mitigate DoS attacks in switches and controller, it needs to limit the flow rules in switches and use either replica of controller or multiple controllers respectively.

Table III: Attacks and Countermeasures in SDN Architecture

SDN Layers	Attacks	Caused by	Existing Techniques as Countermeasures
Infrastructure layer	Man-in-the-middle attack between OpenFlow switch and SDN controller	The control link between OpenFlow switch and SDN controller is not secure without SSL/TLS support	FlowChecker [13] ForNOX [14] VeriFlow [15] Controller replication [16]
	DoS attack-flooding of flow rules to overflow flow table and flow buffer	Limited authentication mechanism between OpenFlow switch and SDN Controller OpenFlow switch has limited storage capacity for flow table and flow buffer Large number of packets have to processed by OpenFlow switch in a short time	FlowVisor [17] Virtual source Address Validation Edge (VAVE) [18] Resonance [19]
Control layer	Compromised controller attack	Centralized controller	FloodGuard [20] DDoS Blocking Application [21]
	DoS attack	Centralized controller Controller has limited computing and storage resources Large number of flow requests have to processed by controller in a short time	DISCO [22] McNettle [23] HyperFlow [24]
	Attacks from application programs	Northbound interface is open for programming Malicious Applications	FRESCO [25]
Application layer	Illegal access of applications	No authentication mechanism	NICE [26] Verificare [27]

		Controller vulnerable as it relies on Operating System	VeriCon [28]
	Conflicts in security rules and configuration for application software	Difference of access control and accountability for various application software	Flover [29] Anteater [30] NetPlumber [31]

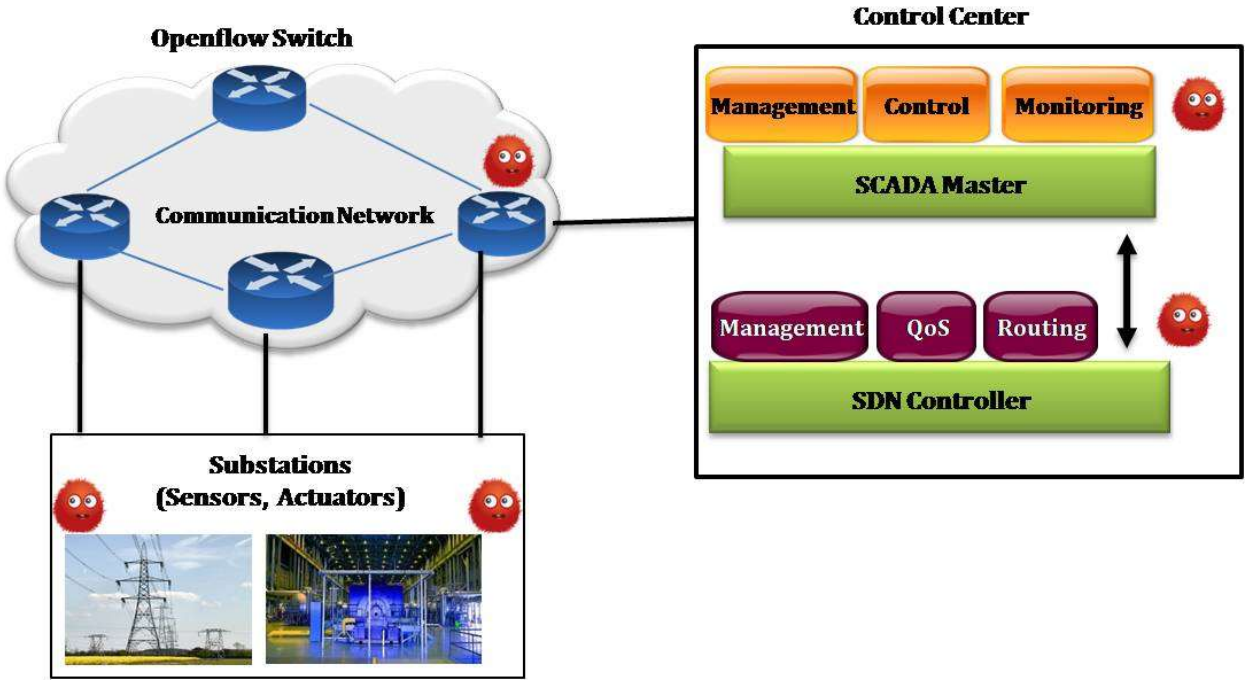

Figure 9: Attacks on SDN-based Smart Power Grid Networks

4 Attacks and Countermeasures in SDN-based Smart Power Grids

This Section discusses attacks and countermeasures in SDN-based smart power grids.

4.1 Attacks in SDN based Smart Power Grid

The attacks in SDN-based smart power grids are presented through case studies and then the attacks are categorized in following Subsection.

4.2 Attack Case Studies

As discussed earlier, the SDN-base smart grid architecture consists of three segments: (i) a control center; (ii) communication network, and (iii) substations. Attacks can be seen in any of these three segments as

illustrated in Figure 9. This Chapter considers three attack cases in SDN-based smart power grid as follows controller [2], [4]:

In *first case*, control center can be compromised by compromising SCADA master or SDN controller or even by compromising their applications. The compromised SDN controller can issue malicious control-commands (such as Add_Flow, Del_Flow, Mod_Flow) to the OpenFlow switches in order to degrade the performance of the network and subsequently the smart grid. Similarly, the compromised SCADA master can issue malicious control-commands (such as Read, Write, Execute) to the SCADA slaves and degrade the performance.

In *second case*, OF switches in the communication network segment may compromise and may drop or inject false packets and also delay the packets that carry measurement data/control-commands from SCADA slaves/master to SCADA master/slaves. For instance, a packet that carries a critical control-command like open a breaker of a relay, can be dropped or delayed by an intermediate compromised switch. This may cause a potential risk to physical infrastructure of a substation in SDN-based smart grid networks.

In *third and final case*, the SCADA slaves can be compromised and can inject malicious measurement data into the smart grid network. This is important to detect and identify bad data in measurements while estimating the state of the smart grid network correctly by the SCADA master. The failure communication between SCADA slaves and SCADA master and the poor calibration and false injection of malicious measurement data by SCADA slaves **[32]** are the main sources of bad data.

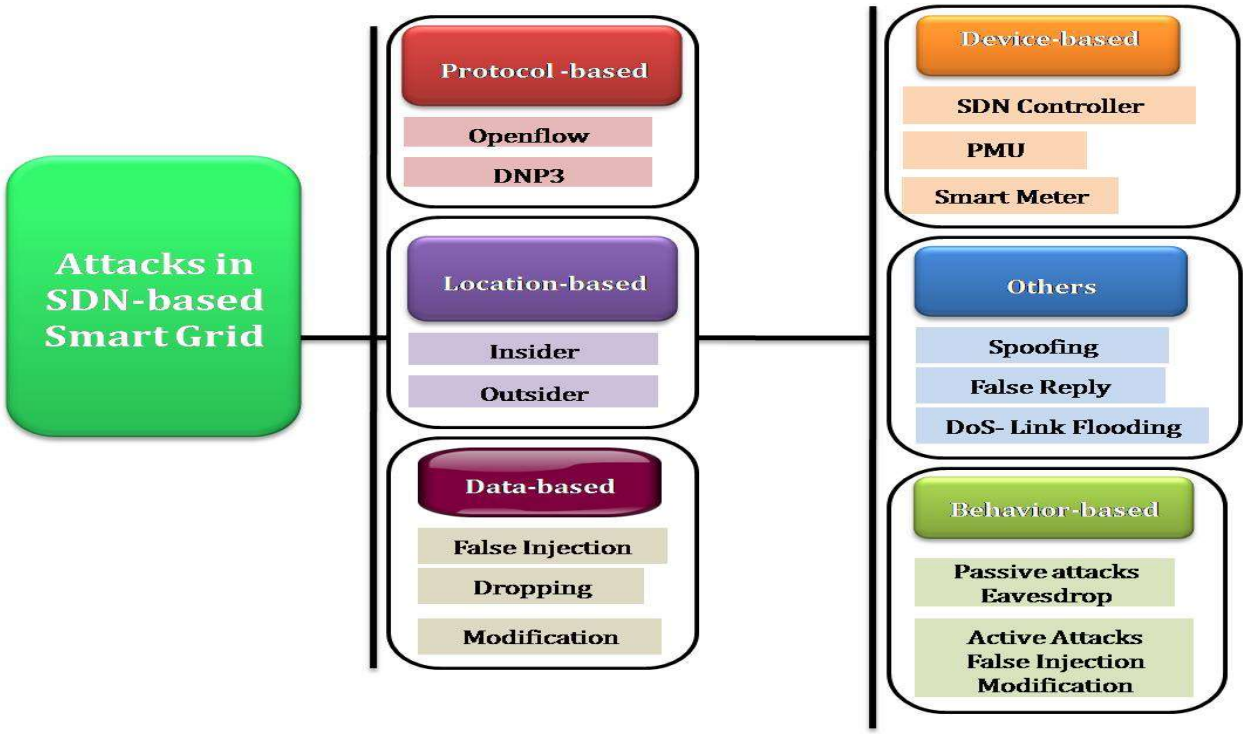

Figure 10: Attack Model for SDN-based Smart Power Grids

4.2.1 Categorization of attacks in SDN-based Smart Power Grid Networks

The attacks related to SDN-based smart grid can be classified into five different complimentary categories as shown in Figure 10: (1) Behavior-based; (2) Location-based; (3) Protocol-based; (4) Device-based; (5) Data-based.

Behavior-based attacks depend on the behavior of the attackers and their attacks execution. It can be either passive or active. In passive attack, the attacker can monitor (or eavesdrops) and analyze the grid data or communication data to gain meaningful information in the SDN-based smart grid network. This attack is easy to launch and it may lead to active attack. In active attack, the attacker can drop, modify the packets or even inject false packets to disrupt the normal operation of the SDN-based smart grid network. Passive attack is difficult to detect as the operation of network is not affected by this attack.

Whereas active attack is easier to detect as the normal operation of the network may be affected seriously. Location-based attacks rely on the location of the attacker and can be either external or internal. External attacks carried out by an attacker that does belong to the network. Whereas internal attacks carried by a compromised node which is actually a part of the network. Insider attackers are more dangerous compared to outside attackers as they have better knowledge about the secret information and internal architecture of the SDN-based smart grid network.

In protocol-based attacks, the attacker exploits the protocol (OpenFlow, DNP3, TCP/IP) vulnerabilities that run in the SDN-based smart grid network. These protocols can either be associated with SDN (OpenFlow) or smart grid (DNP3). In device-based attacks, the attacker targets a specific device for maximum gains potential malicious activities. These attacks can be either related to SDN devices (SDN controllers, OF switches) or smart grid devices (smart meters, PMUs, IEDs). Lastly, in data-base attacks, the attacker can inject, modify even drop the data packets. For example, the attacker can inject false meter data, prices and emergency event in the smart grid. The attacker can spread malware with the data to infect SDN controllers and SCADA devices (smart meters) so that it can steal sensitive information. These attacks may affect the smart grid financially on the electricity markets. Spoofing (IP/Hardware Address), false reply and denial of service (DoS) attacks are well known and can be seen in SDN-based smart grids too.

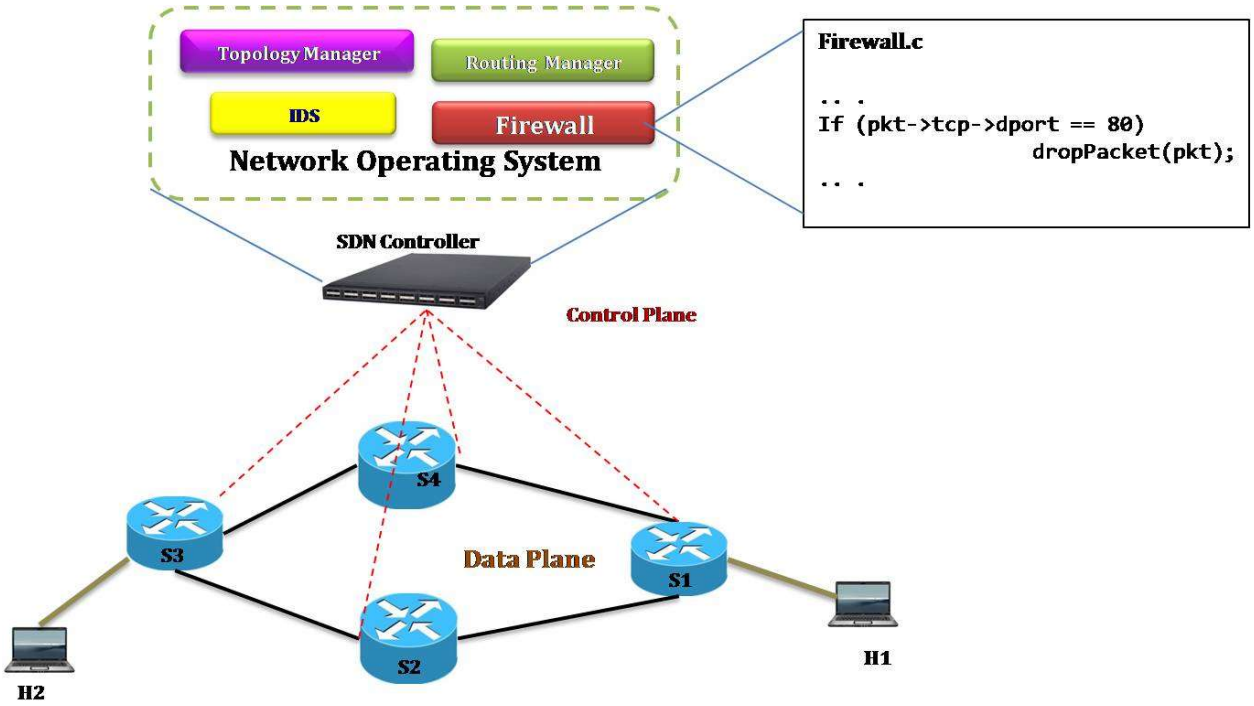

Figure 11: Security using SDN

4.3 Countermeasures of attacks in SDN-based Smart Power Grid Networks

SDN can provide security in smart power grid by providing consistent access control, applying efficient and effective security policies, and managing and controlling the network through the use of controller. Firewall can be developed as an application at the top of SDN controller where efficient and effective policies can be written to provide security against attacks from the outsider and even from the insider. Further, intrusion detection system (IDS) can be developed as an application to detect attackers from inside the network. SDN controller can perceive the entire network traffic periodically. Therefore IDS can easily detect abnormal behavior in network traffic caused by an attacker. Furthermore, SDN controller can timely deal with new attacks as the controller is programmable. These facts are illustrated in Figure 11.

Recently, a number architectures [2], [33], [34], [35], [36] have been proposed for SDN-based smart grid to provide security by utilizing the above features of SDN. In [33], the authors proposed a network based intrusion detection system architecture (NIDS) for SDN-based SCADA systems. One-class classification (OCC) algorithm is basically proposed in NIDS. The NIDS architecture consists of a main control center, 8 distribution substations, 4 intermediate control center, and several number of field devices. They evaluated the performance of the proposed architecture using OpenFlow SDN controller and demonstrated that the OCC algorithm can detect the intrusion in SCADA system with 98% accuracy.

The authors of [34] proposed a solution for power distribution subsystems. They used SDN to make the network auto-configurable, secure and reliable against possible system inappropriate configuration. They further developed a prototype using the Ryu OpenFlow controller and evaluated in a testbed with real

SCADA devices. In [36], the authors presented a SDN based architecture for a substation that follows the IEC 61850 standard. They developed automation techniques for performing a flow-based resource management that enable features such as routing, traffic filtering, QoS, load balancing, and security.

Table IV: Secure Architecture for SDN-based Smart Grid Networks

Framework/ Architecture	First Case		Second Case	Third Case	Security Tool / Mechanism Used
	SCADA Master Security	SDN Controller Security	OF Switch Security	SCADA Slave Security	
Silva et al. [33]	Not Secured	Not Secured	Secured	Secured	Network based IDS (NIDS)
Cahn et al. [34]	Not Secured	Not Secured	Secured	Not Secured	SDN controller policies
Molina et al. [36]	Not Secured	Not Secured	Secured	Not Secured	sFlow collector
Dong et al [35]	Secured	Not Secured	Secured	Not Secured	Control center runs centralized IDS
Ghosh et al controller [2]	Secured	Secured	Secured	Secured	Distributed IDS: Each substation runs IDS

In [35]., the authors investigated (i) how the resilience of Smart Grids can be enhanced against malicious attacks by SDN, (ii) additional risks due to SDN and how to manage them, and (iii) how to evaluate and validate solutions for SDN-based Smart Grids. They further discussed the concrete security issues and their possible countermeasures. Another security framework for SDN-based smart grid is proposed in [2]. In their study, the authors used a global SDN controller at control center and a local SDN controller at each substation along with security controllers to protect the smart grid networks. The framework runs a local IDS in each substation collect the measurement data periodically and to monitor the control-commands

that are executed on SCADA slaves. Whereas control center runs a global IDS and collects the measurement data from the substations and estimates the state of the smart grid system. The global IDS further verifies the consequences of control-commands issued by either the SDN controller or the SCADA master.

Table IV presents the existing secure frameworks for SDN-based smart grid networks. These frameworks mainly provide security in substations. The existing security frameworks/schemes can be further classified according to the area in which they are applied: substation, advanced metering infrastructure (AMI) and phasor measurement unit (PMU) networks, and different networks [9] as given in Figure 12. The schemes presented in [37] and [38] provide security in substations against eavesdropping and link flooding attacks respectively. The anti-eavesdropping scheme in [37] achieves secure communication by using multipath routing in a SCADA system. In [38], the authors proposed a security score model based on SDN to protect the substation against the link flooding attack. The OpenFlow controller can easily enforce QoS policies and identify heaviest flows and busiest communications links at real time.

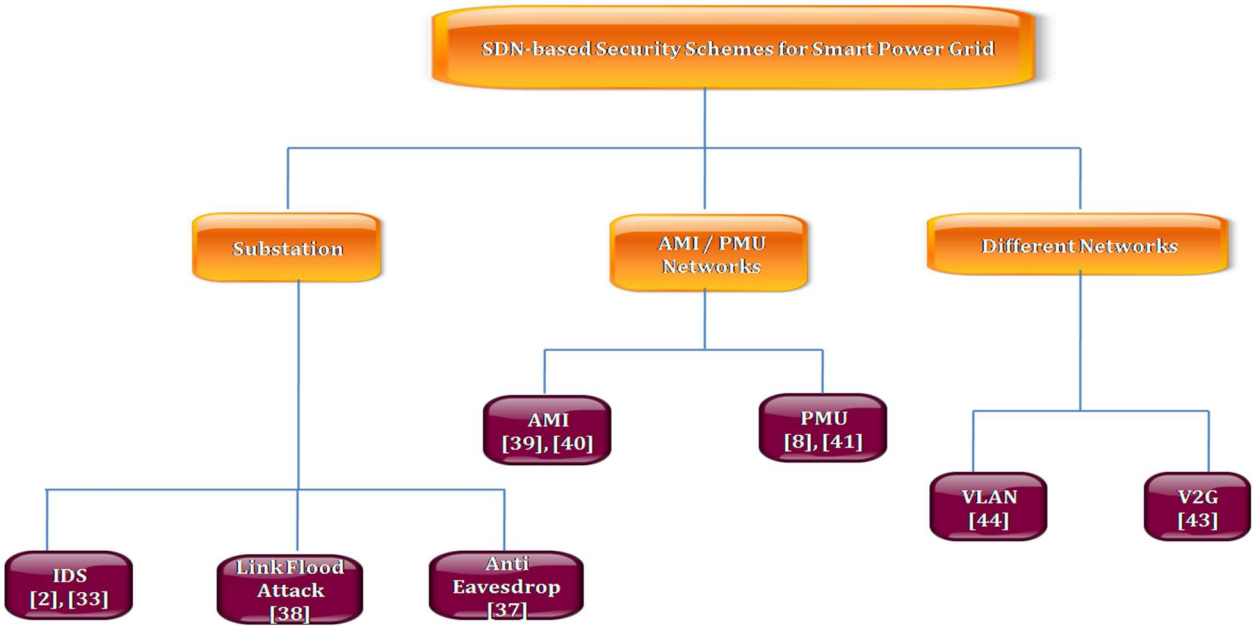

Figure 12: Classification of Security Schemes for SDN-based Smart Grids according to substation, AMI and PMU

A security architecture have been proposed in [39] for the protection of data of AMI network. The proposed security architecture used Flowvisor as a SDN controller. Flowvisor provides the virtualization and slices the networks and also helps to ensure authorization, authentication, and confidentiality. Furthermore, the smart meter sends data using long-term evaluation (LTE) is used for the sending of smart meter data which is then compared with AES-128 encrypted metering data sent by the SG controller. An efficient and privacy-aware power injection (EPPI) security scheme is proposed in [40] for SDN-based AMI networks. It uses message authentication code (MAC) for providing security to the customers. EPPI provides security against the replay attacks. In reply attack, the attacker captures the record of valid packets and replays it in future.

Table V: Attacks and Countermeasures in SDN-based Smart Grids

Attacks Category	Attacks	Existing Frameworks/Schemes	Possible Countermeasures
Location-based	Insider	[2], [33], [34], [35], [36]	• IDS as an SDN application • SDN controller policies
	Outsider	[2], [34], [35]	• Cryptography: Message Authentication Code (MAC), Digital Signature • SDN controller policies with list of authorized nodes
Behavior-based	Passive	[37]	• Symmetric /Asymmetric cryptography for confidentiality
	Active	[2], [34], [35], [7]	• IDS as an SDN application • Cryptography: Authentication & Confidentiality
Device-based	SDN Controller	[16] , [22], [23], [24]	• Controller Replication • Multiple Controllers
	PMUs	[8], [41]	Identity based cryptographic authentication : MAC, Digital Signature
	AMIs	[43], [44]	Identity based cryptographic authentication : MAC, Digital Signature
Protocol-based	OpenFlow	[35]	Cryptography: MAC, Digital Signature, Secure socket layer (SSL)
	DNP3	[8]	
Data-based	False Injection	[2], [33], [35]	• IDS as an SDN application • Data Authentication: MAC, Digital Signature • Data Confidentiality: Symmetric/ Asymmetric Encryption
	Dropping		
	Modification		
Others	Spoofing	[2], [8], [33], [39]	Identity based cryptographic authentication: MAC, Digital Signature
	False Reply		Cryptographic authentication: MAC with timestamp, Digital Signature with challenge-response scheme
	DoS		Using rate limiting and packet dropping techniques

Security schemes for SDN based PMU networks are proposed in [8], [41]. In [41], the authors extended the work [8] and integrated SDN technology with microgrid. They mainly focused on security and

reliability of microgrid which can be provided by SDN. The authors also developed a SDN testbed for microgrid evaluation which is DSSnet [42].

In [43], the authors proposed a software defined Vehicle-to-Grid (SDN-V2G) architecture. SDN-V2G architecture provides security against different types of attacks: (i) attacks on the utility server; (ii) attacks on the communication network of the utility; (iii) attacks on the SDN controller; (iv) attacks on the charging stations; and (v) attacks on the vehicles. SDN based virtual utility network (SVUNs) architecture is proposed in [44] for machine to-machine (M2M) applications in smart grids. IEEE 802.1Q is used to create virtual LANs (VLANs) in traditional networks. However, it has limitations: IEEE 802.1Q supports limited number of devices and only one time authentication for the M2M devices. Thus, SVUNs used SDN for creating virtual utility networks as SDN supports a large number of devices and provides security even after the first time authentication of M2M devices. Table V presents the attacks and countermeasures in SDN-based smart grids.

5 CONCLUSION AND FUTURE WORKS

This Chapter mainly studies different types of attacks and countermeasures in software-defined networking and SDN-based cyber-physical systems. The Chapter first presented a background architecture of SDN technology, smart grids and SDN-based smart grids. It discussed and classified different types of the attacks related to SDN and SDN-based smart grids. Existing security schemes along with their classification are then presented. The Chapter further discussed possible countermeasures of various attacks that can be seen in SDN and SDN-based smart grids.

In the future, we are interested to design and develop a secure and resilient SDN-based smart grid network. In order to make the framework secure, SDN controller can run applications like intrusion detection system (IDS), intrusion elimination system (IES), and key distribution system (KDS). IDS can periodically monitor the devices and their traffics in order to detect misbehavior nodes, whereas IES eliminates the malicious nodes from the SDN-based smart grids by using controller security policies. Key distribution system can distribute the cryptographic keys to the devices for proving authentication and confidentiality in SDN-based smart grids.

6 Bibliography

[1] M. Conti et al., "Looking ahead in pervasive computing: challenges and opportunities in the era of cyber-physical convergence," *Pervasive and Mobile Computing*, vol. 8, no. 1, pp. 2-21, 2012.

[2] U. Ghosh, P. Chatterjee, and S. Shetty, "A Security Framework for SDN-Enabled Smart Power Grids," in *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, Atlanta, USA, 2017, pp. 113-118.

[3] U. Ghosh et al., "A simulation study on smart grid resilience under softwaredefined networking controller failures," in *2nd ACM Workshop on CPSS*, 2016, pp. 52-58.

[4] U. Ghosh, P. Chatterjee, and S. Shetty, "Securing SDN-enabled Smart Power Grids: SDN-enabled Smart Grid Security," in *Securing SDN-enabled Smart Cyber-Physical Systems for Next Generation Networks*.: IGI Global, 2018.

[5] H. Lin, A. Slagell, Z. Kalbarczyk, P. Sauer, and R. Iyer, "Runtime semantic security analysis to detect and mitigate control-related attacks in power grids ," *IEEE Transactions on Smart Grid*, 2016.

[6] J. Zhao, E. Hammad, A. Farraj, and D. Kundur, "Network-Aware QoS Routing for Smart Grids Using Software Defined Networks," *Cham: Springer International Publishing*, pp. 384-394, 2016.

[7] N. Dorsch, F. Kurtz, F. Girke, and C. Wietfeld, "Enhanced fast failover for software-defined smart grid communication networks," in *IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1-6.

[8] H. Lin et al., "Self-healing attack-resilient PMU network for power system operation," *IEEE Transactions on Smart Grid*, vol. in Print, 2016.

[9] M. H. Rehmani, A. Davy, B. Jennings, and C. Assi, "Software Defined Networks based Smart Grid Communication: A Comprehensive Survey," *ArXiv e-prints*, January 2018.

[10] D. Kreutz, F. M. V. Ramos, and P. Verissimo, "Towards Secure and Dependable Software-Defined Networks," in *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, Hong Kong, China, 2013, pp. 55-60.

[11] Z. Shu et al., "Security in Software-Defined Networking: Threats and Countermeasures," *Springer Mobile Netw Appl*, 2016.

[12] S. Hong, L. Xu, H. Wang, and G. Gu, "Poisoning network visibility in software-defined networks: New attacks and countermeasures," in *Network and Distributed System Security (NDSS) Symposium (2015), USENIX*, 2015.

[13] E. Al-Shaer and S. Al-Haj, "FlowChecker: configuration analysis and verification of federated OpenFlow infrastructures," in *Proceedings of the 3rd ACM Workshop on Assurable and Usable Security Configuration*, 2010, pp. 37-44.

[14] Porras P et al., "A security enforcement kernel for OpenFlow networks," in *Proceedings of the First Workshop on Hot Topics in Software*, 2012.

[15] A. Khurshid, W. Zhou, M. Caesar, and P. B.Godfrey, "Veriflow: verifying network-wide invariants in real time," in *ACM Proceedings of the first workshop on Hot topics in software defined*, NY, USA, 2012, pp. 49-54.

[16] P. Fonseca, R. Bennesby, E. Mota, and A. Passito, "A replication component for resilient OpenFlow-based networking," in *IEEE Network Operations and Management Symposium*, Maui, HI, 2012, pp. 933-939.

[17] R. Sherwood et al., "FlowVisor: A Network Virtualization," in *Deutsche Telekom Inc. R&D Lab,* , Stanford, Nicira Networks, 2009, p. Tech. Report.

[18] G. Yao, J. Bi, and P. Xiao, "Source address validation solution with OpenFlow/NOX architecture," in *19th IEEE International Conference on Network Protocols (ICNP)*, 2011, pp. 7-12.

[19] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *IEEE Local Computer Network Conference*, Denver, CO, 2010, pp. 408-415.

[20] H. Wang, L. Xu, and G. Gu, "FloodGuard: A DoS Attack Prevention Extension in Software-Defined Networks," in *45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2015, pp. 239-250.

[21] S. Lim, J. Ha, H. Kim, Y. Kim, and S. Yang, "A SDN-oriented DDoS blocking scheme for botnet-based attacks," in *2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN)*, Shanghai, China, 2014, pp. 63-68.

[22] K. Phemius, M. Bouet, and J. Leguay, "DISCO: Distributed multi-domain SDN controllers," in *IEEE Network Operations and Management Symposium (NOMS)*, Krakow, 2014, pp. 1-4.

[23] A. Voellmy and J. Wang, "Scalable software defined network controllers," in *Proceedings of the ACM SIGCOMM*, NY, USA, 2012.

[24] A. Tootoonchian and Y. Ganjali, "HyperFlow: A Distributed Control Plane for OpenFlow," in *In Proceedings of the 2010 internet network management conference on Research on enterprise networking*, 2010, pp. 3-3.

[25] Porras P, Yegneswaran V, Fong M, Gu G, Tyson M Shin S, "FRESCO: Modular Composable Security Services for Software-Defined Networks," in *NDSS*, 2013, pp. 1-16.

[26] Venzano D, Peresini P, Kostic D, Rexford J Canini M, "A NICEway to test OpenFlow applications," in *9th USENIX Conference on Networked Systems Design and Implementation*, 2012.

[27] Lapets A, Bestavros A, Kfoury A Skowyra R, "Verifiablysafe software defined network for CPS," in *2nd ACM International Conference on High Confidence Networked Systems*, 2013, pp. 101-110.

[28] Bjmer N, Gember A, Itzhaky S, Karbyshev A, Sagiv M, Valadarsky A Ball T, "Vericon: towards verifying controller programs in software-defined networks.," vol. 49, no. 6, pp. 282-293, 2014.

[29] Shin S, Yegneswaran V, Porras P, Gu G Son S, "Model checking invariant security properties in OpenFlow," in *International Conference on Communications ICC*, 2013, pp. 1974-1979.

[30] Khurshid A, Agarwal R, Caesar M, Godfrey P, King S Mai H, "Debugging the data plane with anteater," *ACM SIGCOMM Computer Communication*, vol. 41, no. 4, pp. 290-301, 2011.

[31] ChanM, Zeng H, Varghese G,McKeown N, Whyte S Kazemian P, "Real time network policy checking using header space analysis.," in *USENIX Symposium on Networked Systems Design and Implementation*, pp. 99-111.

[32] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grid ," in *16th ACM Conference on Computer and Communications Security ccs*, 2009.

[33] E. G. d. Silva et al., "A One-Class NIDS for SDN-Based SCADA Systems," in *IEEE COMPSAC*, Atlanta, 2016, pp. 303-312.

[34] A. Cahn, J. Hoyos, M. Hulse, and E. Keller, "Software-defined energy communication networks: From substation automation to future smart grids ," in *IEEE SmartGridComm* , 2013, pp. 558–563.

[35] X. Dong, H. Lin, R. Tan, R. K. Iyer, and Z. Kalbarczyk, "Software- defined networking for smart grid resilience: Opportunities and challenges ," in *1st ACM Workshop on CPSS 2015*.

[36] E. Molina, E. Jacob, J. Matias, N. Moreira, and A. Astarloa, "Using software defined networking to manage and control iec 61850-based systems," *Comput. Electr. Eng vol. 43*, pp. 142–154, 2015.

[37] E. G. da Silva et al., "Capitalizing on SDNbased SCADA systems: An anti-eavesdropping case-study," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 165-173.

[38] H. Maziku and S. Shetty, "Software defined networking enabled resilience for IEC 61850-based substation communication systems," in *International Conference on Computing, Networking and Communications (ICNC)*, 2017, pp. 690-694.

[39] A. Irfan, N. Taj, and S. A. Mahmud, "A novel secure SDN/LTE based architecture for smart grid security," in *IEEE PICOM*, Liverpool, 2015, pp. 762-769.

[40] Y. Z. J. Zhao and D. Zheng, "Efficient and privacy-aware power injection over AMI and smart grid slice in future 5G networks," *Mobile Information Systems*, vol. 17, 2017.

[41] D. Jin et al., "Toward a cyber resilient and secure microgrid using software-defined networking," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2494-2504, 2017.

[42] C. Hannon, J. Yan, and D. Jin, "DSSnet: A smart grid modeling platform combining electrical power distribution system simulation and software defined networking emulation," in *ACM SIGSIM-PADS*, NY, USA, 2016, pp. 131-142.

[43] S. Zhang, J. Wu Q. Li, J. Li, and G. Li, "A security mechanism for software-defined networking based communications in vehicle-to-grid," in *IEEE Smart Energy Grid Engineering (SEGE)*, 2016, pp. 386-391.

[44] Y. J. Kim, K. He, M. Thottan, and J. G. Deshpande, "Virtualized and self-configurable utility communications enabled by softwaredefined networks," in *IEEE SmartGridComm*, 2014, pp. 416-421.

[45] U. Ghosh and R. Datta, "A secure addressing scheme for large-scale managed manets," *IEEE Transactions on Network and Service Management vol. 12*, pp. 483–495, 2015.

7 Author Biography

Uttam Ghosh (ORCID ID: 0000-0003-1698-8888) is an Assistant Professor of the Practice of Electrical Engineering and Computer Science, Vanderbilt University. Dr. Ghosh obtained his PhD in Electronics and Electrical Engineering from the Indian Institute of Technology Kharagpur, India in 2013, and has Post-doctoral experience at the University of Illinois in Urbana-Champaign, Fordham University, and Tennessee State University. His main research interests include Cybersecurity, Computer Networks, Wireless Networks, Information Centric Networking and Software-Defined Networking. He is actively working with VECTOR Vanderbilt University – TennSMART consortium is designed to accelerate "Intelligent Mobility" in Tennessee. Dr. Ghosh is selected for Junior Faculty Teaching Fellow for 2018-19 in Vanderbilt

University. He is also serving as Associate Editor and Reviewers of reputed journals and conferences. He is a member of the IEEE and ACM.

Pushpita Chatterjee (ORCID ID: 0000-0002-0775-5540) Pushpita Chatterjee is a Research Consultant at Old Dominion University, VA. She received her PhD from Indian Institute of Technology Kharagpur, India in 2012. Pushpita has a good number of publications to her credit in international journals, conferences and book chapters. Her research interests include mobile computing, distributed and trust computing, wireless ad hoc and sensor networks, information-centric networking and software-defined networking. She is a member of IEEE.

Sachin Shetty (ORCID ID: 0000-0002-8789-0610) is an Associate Professor in the Virginia Modeling, Analysis and Simulation Center at Old Dominion University. Sachin Shetty received his Ph.D. in Modeling and Simulation from the Old Dominion University in 2007. His research interests lie at the intersection of computer networking, network security, and machine learning. His laboratory conducts cloud and mobile security research and has received over $10 million in funding from National Science Foundation, Air Office of Scientific Research, Air Force Research Lab, Office of Naval Research, Department of Homeland Security, and Boeing. He has authored and coauthored over 140 research articles in journals and conference proceedings and two books. He is recipient of DHS Scientific Leadership Award and has been inducted in Tennessee State University's million dollar club.

Charles A. Kamhoua (ORCID ID: 0000-0003-2169-5975) is a researcher at the Network Security Branch of the U.S. Army Research Laboratory (ARL) in Adelphi, MD, where he is responsible for conducting and directing basic research in the area of game theory applied to cyber security. Prior to joining the Army Research Laboratory, he was a researcher at the U.S. Air Force Research Laboratory (AFRL), Rome, New York for 6 years and an educator in different academic institutions for more than 10 years. He has held visiting research positions at the University of Oxford and Harvard University. He has co-authored more than 150 peer-reviewed journal and conference papers. He is a co-inventor of a patent, 6 patent application, and co-editor of 3 books at IEEE Press. He has presented over 40 invited keynote and distinguished speeches and has co-organized over 10 conferences and workshops. He has mentored more than 60 young scholars, including students, postdocs, and Summer Faculty Fellow. He has been recognized for his scholarship and leadership with numerous prestigious awards, including the 2018 ARL Achievement Award for leadership and outstanding contribution to the ARL Cyber Camo (cyber deception) project, the 2018 Fulbright Senior Specialist Fellowship, the 2017 AFRL Information Directorate Basic Research Award "For Outstanding Achievements in Basic Research," the 2017 Fred I. Diamond Award for the best paper published at AFRL's Information Directorate, 40 Air Force Notable Achievement Awards, the 2016 FIU Charles E. Perry Young Alumni Visionary Award, the 2015 Black Engineer of the Year Award (BEYA), the 2015 NSBE Golden Torch Award—Pioneer of the Year, and selection to the 2015 Heidelberg Laureate Forum, to name a few. He has been congratulated by the White House and US Congress for those achievements. He received a B.S. in electronics from the University of Douala (ENSET), Cameroon, in 1999, an M.S. in Telecommunication and Networking from Florida International University (FIU) in 2008, and a Ph.D. in Electrical Engineering from FIU in 2011. He is currently an advisor for the National Research Council postdoc program, a member of the FIU alumni association and ACM, and a senior member of IEEE.

Laurent L. Njilla (ORCID ID: 0000-0001-8902-7418) received his B.S. in Computer Science from the University of Yaoundé 1 in Cameroon, the M.S. in Computer Engineering from the University of Central Florida (UCF) in 2005 and Ph.D. in Electrical Engineering from Florida International University (FIU) in 2015. He joined the Cyber Assurance Branch of the U.S. Air Force Research Laboratory (AFRL), Rome, New York, as a Research Electronics Engineer in 2015. Prior to joining the AFRL, he was a Senior Systems Analyst in the industry sector for more than 10 years. He is responsible for conducting basic research in the

areas of hardware design, game theory applied to cyber security and cyber survivability, hardware Security, online social network, cyber threat information sharing, category theory, and blockchain technology. He is the Program Manager for the Cyber Security Center of Excellence (CoE) for the HBCU/MI and the Disruptive Information Technology Program at AFRL/RI. Dr. Njilla's research has resulted in more than 60 peer-reviewed journal and conference papers and multiple awards including Air Force Notable Achievement Awards, the 2015 FIU World Ahead Graduate award and etc. He is a reviewer of multiple journals and serves on the technical program committees of several international conferences. He is a member of the National Society of Black Engineer **(NSBE).**