Security in Software-Defined Wireless Sensor Networks: Threats, Challenges and Potential Solutions

Sean W. Pritchard
Department of Electrical, Electronic
and Computer Engineering
University of Pretoria
Pretoria, South Africa
Email: spritchard001@gmail.com

Gerhard P. Hancke
Department of Electrical, Electronic
and Computer Engineering
University of Pretoria
Pretoria, South Africa
Email: gerhard.hancke@up.ac.za

Adnan M. Abu-Mahfouz
Meraka Institute
Council for Scientific and Industrial
Research (CSIR)
Pretoria, South Africa
Email: a.abumahfouz@ieee.org

***Abstract*—A Software-Defined Wireless Sensor Network (SD-WSN) is a recently developed model which is expected to play a large role not only in the development of the Internet of Things (IoT) paradigm but also as a platform for other applications such as smart water management. This model makes use of a Software-Defined Networking (SDN) approach to manage a Wireless Sensor Network (WSN) in order to solve most of the inherent issues surrounding WSNs. One of the most important aspects of any network, is security. This is an area that has received little attention within the development of SDWSNs, as most research addresses security concerns within SDN and WSNs independently. There is a need for research into the security of SDWSN. Some concepts from both SDN and WSN security can be adjusted to suit the SDWSN model while others cannot. Further research is needed into consolidating SDN and WSN security measures to consider security in SDWSN. Threats, challenges and potential solutions to securing SDWSN are presented by considering both the WSN and SDN paradigms.**

***Index Terms*—IoT; WSN; security; security threats; SDN; SDWSN**

I. Introduction

A wireless sensor network (WSN) is a network of multiple small inexpensive smart sensor nodes that are capable of physical and environmental data acquisition and wireless communications [1]. The increasing development of the Internet of Things (IoT) paradigm has increased the demand for research in WSNs as a platform for IoT. WSNs are not only important in IoT but also in other applications such as smart water management. The scarcity of water has brought about a serious need for better water management due to the fact that traditional water management is inconvenient and wasteful. WSNs can be used as a platform for real-time smart water meters which can be monitored and controlled in order to reduce water loss [2].

When expanding WSNs, inherent problems due to resource constraints such as limited processing and storage, communication bandwidth and energy [3], [4] become increasingly significant due to the lack suitable network management and incapability of heterogeneous-node network realization [5]. Therefore, unless an appropriate solution to these inherent problems are found, WSNs as they are currently, are unable to fulfill the challenges of the IoT paradigm. However, there has been research into using software-defined networking (SDN) as a solution to the inherent problems of WSNs [6].

Studies have concluded that software-defined wireless sensor networking (SDWSN) is expected to play a large role in both the the IoT paradigm and also in smart water management systems, where SDWSNs can provide real-time monitoring and control of water grid components, resulting in an efficient water management system to mitigate water wastage [7].

One of the most important considerations in any network, however, is that of security within the network. For example, the smart water management system will not only consider real-time monitoring but also real-time control of various components of the water network. Ensuring the security of such a system is critical because it is not only about protecting the collected data or privacy of the consumers but also to prevent any malicious attack from compromising the system and be able to send incorrect control signals that may damage part of the water infrastructure [8].

This is a consideration that has been subject to research within the SDN and WSN paradigms individually; however, in terms of SDWSN this has not received much attention due to the focus on the SDWSN architecture itself [5]. Therefore security within the SDWSN itself needs to be considered if it is to truly play a critical role in the development of the IoT paradigm.

In order to identify solutions and challenges in the security of SDWSN, it is first important to identify any major security challenges and solutions within SDN and WSNs individually. These aspects of security must then be consolidated to consider the SDWSN model to find out which individual security aspects from SDN and WSN can be adapted to the SDWSN model.

II. Security in SDN

Software-defined networking is a networking approach that aims to simplify network management and configuration. This happens by decoupling the control plane which encompasses all the network intelligence from the data plane which is the packet forwarding engine of the network [9]. This essentially creates a global view of the network, allowing dynamic control and network programmability.

Although the SDN paradigm results in many benefits to traditional networking such as centralized network programmability and control, it is also these benefits that introduced new threats and attack planes, compromising the security of the network. Two properties of SDN introduce appealing opportunities for malicious users, the centralization of network intelligence and control of the network using software [10]. Centralizing network intelligence results in access and control of the entire network through access to the controllers host server while the software used to control the network can be susceptible to bugs and other vulnerabilities.

A. Security Threats

Several security threats to SDNs have been identified [10]. These security threats were then reviewed and further categorized by the functional layers of the SDN planes [11], the application plane which houses SDN applications such as network management, the controller plane which contains the control logic framework and the data plane which contains the forwarding elements.

Within the application plane, authentication and authorization is identified as one of the important challenges [11]. The network could be compromised due to the fact that there are no methods for certification and trust between the controller and management applications [10]. The lack of trusted resources for forensics and remediation which allow the network to identify, and provide a fast secure network recovery, will result in useless data and slow or unnecessary recovery.

Traditional threats to the network such as vulnerable administrative stations are also a threat to SDNs and can be exploited to access the controller. The threat seen from a single machine is greatly increased in SDNs as a result of the network programmability that SDN brings.

The controller plane houses the control logic of the network and is therefore highly targeted by malicious users [11]. Denial of service (DoS) attacks and data theft could be carried out through attacks on control plane communications. After gaining access to the controller, an attacker may launch distributed DoS (DDoS) attacks using a large number of switches under the control of an attacker. The lack of trust in Transport Layer Security (TLS) and Secure Sockets layer (SSL) communications could also result in data leakage during regular network traffic.

Vulnerabilities in the controller may also lead to attacks on the controller itself. This may be the most severe threat to SDNs as it may result in compromising the entire network. Regular intrusion detection may not be able to identify and label malicious activity on the controller and once a malicious user has control of the controllers, they may be able to do as they please due to the fact that the controllers issue configuration commands.

Within the data plane, forged or fake traffic flows caused by both faulty devices and malicious users can compromise network switches and controllers [11]. OpenFlow switches (switches using or compatible with the OpenFlow protocol used in the SDN environment) and controller resources can be attacked using network elements and DoS attacks [10]. An attacker can also inject forged flows into the network after assuming control of the server that stores authentication details of users and using these details to authenticate these forged flows.

The next threat in the data plane is attacking vulnerabilities in switches. Due to the nature of SDN, packet forwarding is already one of the problems presented [12], however by assuming control of a single switch, these packets can be dropped or used to slow down the network, clone or divert traffic or inject forged requests and forged traffic to overload the network [10]. Thus, by exploiting vulnerabilities in the switch, the performance of the entire network is compromised.

B. Solutions

Ali *et al.* [9] propose that the SDN paradigm provides increased security for networks in terms of threat detection, remediation and network correctness as well as security as a service. However when looking at the security in SDN itself, most researchers agree [9], [11] that there is a need to include security into the design of SDNs from the ground up [10].

FRESCO [13] allows security applications to be implemented on any OpenFlow controller [11]. This enables security application development and policy implementation on NOX controllers [14]. NOX defines a network operating system for OpenFlow controllers that allows management and centralized control application development.

On the application layer, authentication and authorization is the main concern therefore most of the solutions to application layer security include validation and verification models [5]. PermOF [15], a permission control application, is proposed to ensure that applications have controlled access to network resources [11].

The network must also be aware any of changing conditions and verify these changes. VeriFlow [16] is an assertion based debugging and verification language used to debug any faulty rules before they cause harm to the network. This language has been proposed to enable verification and debugging of SDN applications using dynamic verification methods [17]. Flover [18] is a verification system for OpenFlow networks that is implemented on the controller to ensure that no new policies created on the controller conflict with specified properties.

Ahmad *et al.* propose multiple approaches to securing the control plane [11] such as protection against malicious or faulty applications, protection against targeting scalability of the control plane, DoS/DDoS prevention and ensuring security through reliable controller placement.

Security against malicious or faulty applications is important due to the fact that network resources can be accessed through the control plane. A Security-enhanced (SE) Floodlight controller [19] is proposed to secure the control layer through separation of privileges, run-time verification models, assigning authorization roles, resolving conflicts between rules, as well as tracking security events within the control layer.

Distributed controllers result in a load on the controllers themselves which provide a platform for various controller based attacks. Various approaches are used to either reduce the load on the controllers using load balancing techniques [20], distribute functionalities using platforms such as DISCO [21] and HyperFlow [22], or increase controller processing power using McNettle controllers [23], which can scale up to 46 cores as opposed to NOX controllers which can scale up to 10 cores.

DoS or DDoS mitigation is carried out by observing flow statistics stored in OpenFlow switches [11]. Self-Organizing Maps (SOM) are proposed to detect lightweight DDoS attacks [24] by uncovering hidden flow relationships entering the network to classify traffic as normal or an attack.

Controller placement provides challenges to scalability and resilience in the network which greatly effects network performance in SDN. Thus multiple algorithms have been developed and tested for optimal controller placement [11] such as the Simulated Annealing (SA) algorithm and Pareto-based Optimal Controller-placement (POCO) framework.

In terms of securing the data plane, FortNOX [25] can be used as a platform to authorize applications that can change flow rules in real-time to protect the data plane from malicious applications which may install or modify flow rules. FlowChecker [26] is a tool used to verify inconsistency in rules to validate and enforce end-to-end configuration rules. VeriFlow [16] can also be used to find faulty rules to prevent harm to the network.

C. Summary

A summary of the existing platforms and solutions to the problems discussed above can be seen in TABLE I. Authentication and authorization concerns on the application plane have been addressed by various verification systems. The control plane has authorization, scalability, availability and DoS/DDoS mitigation platforms to address control plane and vulnerabilities. The data plane has protection against forged or faked traffic flows in the form of different verification and debugging tools.

It is important to note that these solutions do not address threats that could arise from the data forwarding functionality of SDN nodes. Although there are verification tools to prevent vulnerabilities in the data plane, the severity of the vulnerabilities increase due to the nature of the SDN paradigm which could mean that these tools will not be able to handle the load presented by the packet forwarding functionality. Another threat that had not been readily addressed is that of the lack of trust in TLS/SSL communications i.e. securing control plane communications and the lack of trusted forensics and remediation applications in the application plane.

III. Security in WSN

Due to the fact that WSNs use wireless means of communication, security within WSNs is important and must be addressed. A malicious user can easily carry out attacks to intercept data and waste network resources. Many authors attribute the inherent problems with WSNs such as processing and communication limitations, to the inability to implement security measures [1], [27] - [30]. Fundamental security requirements in WSNs have been identified as data authentication, data confidentiality, data integrity, and availability and redundancy [31], [32].

TABLE I
Existing Solutions for SDN planes [11]

SDN plane	Existing Platform	Threat Addressed	Approach
Application	FRESCO	Threats from applications	Framework for security application development
	PermOF	Access control	Permission control application
	Assertion	Flow rules contridection	Debugging framework
	Flover	Security Policy violation	Policy verification
Control	SE-Floodlight	Application authorization	Secure controller architecture
	DISCO	Controller scalability	Distributed controller architecture
	Placement Algorithms		Placement frameworks
	HyperFlow	Controller availability	Distributed control plane
	SOM	DoS/DDoS attack	Detection framework
Data	FortNOX	Flow rule contradiction	Controller framework
	FlowChecker	Faulty flow rules	Configuration verification tool
	VeriFlow		Network debugging tool

A. Security Threats

In order to achieve fundamental security goals, the threats to WSNs must first be identified. Attacks on WSNs are categorized into goal-orientated, performer-orientated and layer-oriented attacks [27], [33].

Goal-orientated attacks consist of passive and active attacks [27]. Passive attacks are carried out when a malicious user monitors sensitive network information without disturbing network operations, so that it may be used in other attacks. These attacks result in the disclosure of sensitive or data files unbeknown to the network user. Active attacks is when the attacker uses this information to assume control over the network or disrupt network operation. Examples of active attacks include DoS, wormhole, hello flood, Sybil, blackhole/sinkhole, data modification and spoofing [5], [27], [28], [32], [33].

Performer-orientated attacks consist of outside and inside attacks [27]. Outside attacks allow for monitoring data transmissions as well as injecting false data into the network to consume resources resulting in DoS attacks. Inside attacks are when malicious nodes parade as legitimate nodes to damage the network. Once trusted by the network, the malicious node can then launch different attacks [27], the source of which is difficult to locate due to the fact that the malicious node can suppress important information from reaching the base station.

Layer-orientated attacks consist of different attacks targeting various layers of the network stack [27], [28]. Attacks on the physical layer vary from capturing nodes to jamming channels which can lead to DoS attacks [34]. Data link layer attacks target the functionality of link layer protocols [28] by violating established communication protocols. Network layer attacks are attacks that are aimed at disrupting routing protocols, for example DoS and sinkhole attacks. Transport layer attacks consume node resources by flooding it with connection requests. Application layer attacks consist of various attacks such as overwhelm, data corruption and malicious code [27].

B. Implementation Problems

The fundamental barrier to security implementation in WSNs is the inherent issues within WSNs, mainly the resource constraints in WSNs [1], [27] - [33]. Most research surrounding security solutions in WSNs have focused on low resource cryptography methods to secure the network [27] - [29], [33]. Cryptography methods are separated into symmetric cryptography and asymmetric cryptography.

While symmetric cryptography solutions are preferred due to low implementation cost and efficiency [5], they present many problems when managing large networks and attempts to improve this cryptography for WSNs [11] have resulted in the cost of resources. Symmetric cryptography is also difficult to implement in software and is resistant to scalability [32].

Asymmetric cryptography attempts to alleviate the issues of symmetric cryptography for example simplifying cryptographic key management [27]. However this attempt comes at a cost to resources as the methods are too computationally excessive for the nodes [29]. Optimal solutions regarding cryptographic security implementation remains a challenge in WSNs due to the fact that neither symmetric or asymmetric cryptography can provide the security needed in WSNs or the IoT as a whole [35].

IV. Security in SDWSN

Using a SDN approach to WSN results in a new paradigm called software-defined wireless sensor networking (SDWSN) [5]. In SDWSN the control logic of the WSN is separated and handled by a separate controller while the data transmission functionality is left behind and handled by the device. Therefore the sensor nodes only carry out data forwarding operations while any computationally intensive tasks are performed by the controller without affecting energy consumption [12]. This approach is said to solve most of the inherent problems with WSNs as well as promote interoperability with other networks and improve efficiency and sustainability in WSNs.

The SDWSN architecture currently applied is shown in Fig. 1. Using an SDN approach to WSNs results in the decoupling of the data and control plane, thus solving most of the inherent problems with WSNs [5], [12]. The nature of the SDN paradigm presents many advantages when considering security in SDWSN.

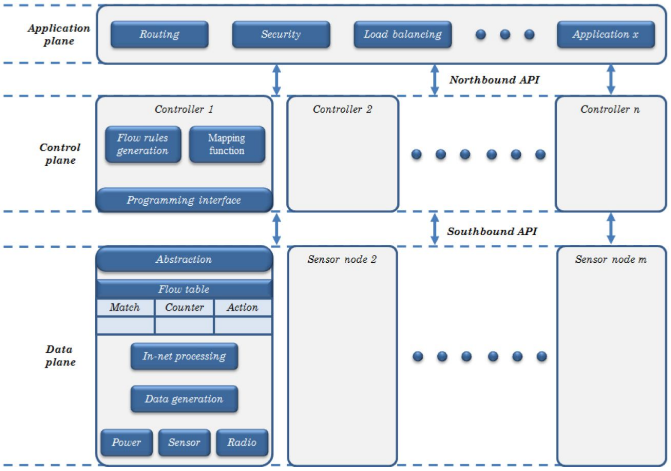

Fig. 1. Basic SDWSN architecture (from [5])

A. SDN Advantages in WSN Security

Although the cryptographic solutions for security in WSNs are based on a coupled architecture as opposed to the decoupled architecture as shown in Fig. 1, the security measures could still be implemented on the control or application plane [5]. The centralization of control results in the centralization of security management and coupled with the fact that the SDN approach partially frees up WSN resources, this allows for the implementation of more resource intense security measures in WSNs [5].

The global view of the network provided by SDN results in a platform for analysis which enables fast identification of malicious activities and therefore a fast response time. The challenges and solutions in SDN as discussed also provide some solutions to WSNs although not all solutions can be adapted due to WSNs unique properties.

One of the most important features that SDN brings to WSN is the fact that the sensor node can only accept control commands due to the separation of control logic [5]. Therefore, it becomes difficult to use the sensor nodes to conduct malicious attacks on the network, although it is still possible to use these nodes as gateways for different attacks.

The SDN approach allows for flexible security configuration as opposed to the error prone manual processes currently implemented [5]. However, the SDN approach also presents its own problems, as errors in the configuration network can also lead to more security vulnerabilities [36] - [37]. This is evident due to the fact that in general a poorly configured network results in more network vulnerabilities.

B. SDWSN Proposed Solutions

The SDN paradigm itself already presents some solutions to WSN security which can be directly incorporated into SDWSN security. As discussed above, the centralization of control simplifies security management and results in less resource constraints on the node itself, thus allowing security implementation. The SDN also provides a global view of the network for fast analysis and response.

In order to identify which security solutions can be adapted from SDN and WSN into the SDWSN paradigm, it is important to identify which security threats are present in both SDN and WSN. TABLE II shows the security threats associated with both the WSN network stack layers and the SDN planes [5], [9].

TABLE II
WSN LAYER AND SDN PLANE SECURITY THREATS (FROM [9])

WSN layer	SDN plane	Threat
Application	Application	Poor Authentication and Control
		Fraudulent flow rules insertion
		Poor access control and accountability
		Malicious Applications
		DoS
		Northbound Interface (API) attack
Transport	Control	Threats from applications
		DoS
Network		Unauthorized access
		Scalability and Unavailability
		Faulty or Malicious controller
Data Link	Data	Unauthorized access
		Fraudulent rules
		Forged/False traffic flows
		Flooding, Spoofing
Physical		Southbound Interface (API) attack
		Jamming, Tampering
		Sybil
		Compromised/hi-jacked controller
		Malicious node

The application plane of the SDWSN architecture as seen in Fig. 1 consists of the application layers from the WSN network stack and the application plane from the SDN model. This plane is mostly susceptible to SDN security vulnerabilities such as the authentication and authorization challenges [11], lack of trust mechanisms [10] and threats from network applications. These problems result in the threats shown in TABLE II and therefore applying SDN solutions in the framework of SDWSN application layer security would potentially solve most of the application layer threats. These solutions include implementing validation and verification models on platforms such as PermOF, VeriFlow and Flover. Cryptographic solutions shown in WSN security could also be implemented on the application layer to secure northbound interface communications.

The control plane of the SDWSN model shown in Fig. 1 consists of threats to the SDN control plane and WSN transport and network layer threats. Cryptography solutions in WSN can be applied to the controller to secure the control plane against transport layer attacks. SDN control plane solutions mitigate network layer attacks through the approaches proposed by Ahmad *et al.* [11] such as using SE Floodlight controllers to protect against malicious applications, increasing controller scalability using platforms such as DISCO and HyperFlow, using Self-Organizing Maps to mitigate DoS attacks and optimizing controller placement using algorithms such as Simulated Annealing algorithms.

The data plane in Fig. 1 consists of the WSN physical and data link layers as well as the SDN data plane. SDWSN are susceptible to these threats mainly through WSN data link and physical layer security threats. However, by using the SDN approach, most of these data link and physical layer vulnerabilities are mitigated as the sensor node is only able to accept control commands and is therefore difficult to use as a malicious node. The main problem with security in the data plane comes from the packet forwarding problems associated with SDN [12]. This becomes an even greater problem when considering the SDWSN paradigm as this problem becomes a platform for the attacks mentioned in TABLE II. The only solution to these data plane threats is using validation platforms such as FortNOX, FlowChecker and VeriFlow to mitigate unauthorized access, fraudulent rules, forged flows and southbound interface attacks.

Although some of the platforms and solutions mentioned above may be able to secure SDWSNs to a certain degree, they are not ideal solutions as most rely on external security methods which focus on network devices such as switches instead of WSNs. In order to truly secure SDWSNs, security must be built into the SDWSN architecture itself [10] and be compatible with other design requirements. Using Network Function Virtualization (NFV), which visualizes network functionalities [5], most security solutions can be implemented which is a step towards incorporating security solutions into the SDWSN architecture itself.

C. Remaining Challenges

Although some of the solutions from SDN and WSN security that can be incorporated into the SDWSN model have been identified, further research is needed in adapting these solutions to consider the SDWSN model as opposed to SDN and WSN individually. The solutions mentioned above have been identified from existing research and have not been confirmed as valid security models in SDWSN.

When considering the SDN side of SDWSN, the control plane presents the biggest target for malicious attacks [11] because compromising the controller compromises the network. Therefore the control plane needs to be secure against all threats identified in TABLE II. The communication protocol used by the controller must be secure against any form of interception. Ali *et al.* states that using the SSL/TLS protocols for small devices such as sensor nodes is impractical [9] and therefore more research is needed in developing secure network communication.

Further study is needed to confirm whether sensor nodes used in SDWSN pose less of a threat to data plane as they are at the periphery of the network and generate traffic as opposed to traditional SDN switches which can be used as gateways for other attacks [5]. One of the most problematic features of the SDWSN paradigm is that of the data forwarding issues on the data plane [38]. These issues may result in more security vulnerabilities and must therefore be resolved in order to truly secure SDWSNs.

V. CONCLUSION

The SDWSN model uses a SDN approach to solve most of the inherent issues in WSNs however the SDWSN model still presents many challenges due to the fact that it incorporates two models which are not yet fully fledged. This is also apparent when considering security implementation in SDWSN. WSNs have many inherent issues such as resource limitations which result in the inability to implement security measures. SDN also has its own problems such as the trade-off between functionality and performance especially on the data plane which result in security vulnerabilities.

However, some of the solutions proposed in each individual paradigm can be consolidated to consider security in the SDWSN paradigm, although due to the nature of the two different paradigms, not all solutions can be combined and adapted to the SDWSN model. Current work on SDN, WSN and SDWSN has been reviewed in order to identify problems and solutions which could be adapted to the SDWSN model however some challenges remain. These challenges must be addressed in order to realize secure SDWSNs, which is important if SDWSN is to truly play a critical role in the development of the IoT paradigm.

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.

[2] M. Mudumbe and A. M. Abu-Mahfouz, "Smart water meter system for user-centric consumption measurement," in *Proceedings of the IEEE International Conference on Industrial Informatics*, Cambridge, UK, Jul. 2015, pp. 993–998.

[3] A. M. Abu-Mahfouz and G. P. Hancke, "Localised information fusion techniques for location discovery in wireless sensor networks," *International Journal of Sensor Networks (IJSNET), in press*, 2017.

[4] A. M. Abu-Mahfouz and G. P. Hancke, "ALWadHA localisation algorithm: Yet more energy efficient," *IEEE Access*, vol. 5, no. 5, pp. 6661–6667, Mar. 2017.

[5] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "A survey on software-defined wireless sensor networks: Challenges and design requirements," *IEEE Access*, vol. 5, pp. 1872–1899, Feb. 2017.

[6] M. Ndiaye, G. P. Hancke, and A. M. Abu-Mahfouz, "Software defined networking for improved wireless sensor network management: A survey," *Sensors*, vol. 17, no. 5:1031, pp. 1–32, 2017.

[7] N. Ntuli and A. Abu-Mahfouz, "A simple security architecture for smart water management system," in *Proc. 11th Int. Symp. on Intell. Tech. for Adhoc and Wirel. Sens. Netw.*, Spain, Madrid, May 2016, pp. 1164 – 1169.

[8] A. Abu-Mahfouz, Y. Hamam, P. R. Page, K. Djouani, and A. Kurien, "Real-time dynamic hydraulic model for potable water loss reduction," *Procedia Engineering*, vol. 154, no. 7, pp. 99–106, Aug. 2016.

[9] S. T. Ali, V. Sivaraman, A. Radford, and S. Jha, "A survey of securing networks using software defined networking," *IEEE Trans. Rel.*, vol. 64, no. 3, pp. 1086–1097, Sep. 2015.

[10] D. Kreutz, F. M. V. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in *Proc. Second ACM SIGCOMM Work. Hot Top. Softw. Defin. Netw. - HotSDN 13*, 2013, p. 55.

[11] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in software defined networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2317–2346, Aug. 2015.

[12] K. M. Modieginyane, B. B. Letswamotse, R. Malekian, and A. M. Abu-Mahfouz. (2017, Mar.) Software defined wireless sensor networks application opportunities for efficient network management: A survey. Computers and Electrical Engineering. [Online]. Available: http://dx.doi.org/10.1016/j.compeleceng.2017.02.026

[13] S. Shin, P. Porras, V. Yegneswaran, and M. Fong, "FRESCO: Modular composable security services for software-defined networks," *ISOC Network and Distributed System Security Symposium*, vol. 2, pp. 1–16, Feb. 2013.

[14] N. Gude, T. Koponen, J. Pettit, M. C. B. Pfaff, N. McKeown, and S. Shenker, "NOX: towards an operating system for networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 3, pp. 105–110, 2008.

[15] A. Aissioui, A. Ksentini, A. M. Gueroui, and T. Taleb, "Toward elastic distributed SDN/NFV controller for 5G mobile cloud management systems," *IEEE Access*, vol. 3, pp. 2055–2064, 2015.

[16] A. Khurshid, W. Zhou, M. Caesar, and P. B. Godfrey, "VeriFlow: Verifying network-wide invariants in real time," in *Proc. First Work. Hot Top. Softw. Defin. Netw. - HotSDN 12*, 2012, p. 49.

[17] R. Beckett, X. K. Zou, S. Zhang, S. Malik, J. Rexford, and D. Walker, "An assertion language for debugging SDN applications," in *Proc. Third Work. Hot Top. Softw. Defin. Netw. - HotSDN 14*, 2014, pp. 91–96.

[18] S. Son, S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "Model checking invariant security properties in openflow," *IEEE Int. Conf. Commun.*, pp. 1974–1979, 2013.

[19] Security-enhanced floodlight. SDx Centralz, Sunnyvale, CA, USA. [Online]. Available: http://www.sdncentral.com/education/towardsecure-sdn-control-layer/2013/10/

[20] I. Ahmad, S. N. Karunarathna, M. Ylianttila, and A. Gurtov, "Load balancing in software defined mobile networks," *Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture.*, pp. 225–245, 2015.

[21] K. Phemius, M. Bouet, and J. Leguay, "DISCO: Distributed multidomain SDN controllers," in *Proc. IEEE NOMS*, 2014, pp. 1–4.

[22] A. Tootoonchian and Y. Ganjali, "HyperFlow: A distributed control plane for openflow," in *Proc. Internet Netw. Manag. Conf. Res. Enterprise Netw. USENIX Assoc.*, 2010, p. 3.

[23] A. Voellmy and J. Wang, "Scalable software defined network controllers," in *Proc. ACM SIGCOMM Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, 2012, pp. 289–290.

[24] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *Proc. IEEE 35th Conf. LCN*, 2010, pp. 408–415.

[25] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, and G. Gu, "A security enforcement kernel for openFlow networks," in *Proc. First Work. Hot Top. Softw. Defin. Netw. - HotSDN 12*, 2012, p. 121.

[26] E. Al-Shaer and S. Al-Haj, "FlowChecker: Configuration analysis and verification of federated openflow infrastructures," in *Proc. 3rd ACM Workshop SafeConfig*, 2010, pp. 37–44.

[27] K. Chelli, "Security issues in wireless sensor networks: Attacks and countermeasures," in *Proc. World Congr. Eng.*, vol. 1, 2015.

[28] T. Zia and A. Zomaya, "Security issues in wireless sensor networks," in *2006 International Conference on Systems and Networks Communications (ICSNC06)*, 2006, p. 40.

[29] J. Louw, G. Niezen, T. D. Ramotsoela, and A. M. Abu-Mahfouz, "A key distribution scheme using elliptic curve cryptography in wireless sensor networks," in *Industrial Informatics (INDIN), 2016 IEEE 14th International Conference*, 2016, pp. 1166–1170.

[30] A. M. Abu-Mahfouz and G. P. Hancke, "Evaluating ALWadHA for providing secure localisation for wireless sensor networks," in *Proc. IEEE AFRICON 2013 conf.*, Mauritius, Sep. 2013, pp. 501–505.

[31] D. Christin, A. Reinhardt, P. S. Mogre, and R. Steinmetz, "Wireless sensor networks and the internet of things: Selected challenges," *Struct. Heal. Monit.*, vol. 5970, pp. 31–33, 2009.

[32] H. Modares, R. Salleh, and A. Moravejosharieh, "Overview of security issues in wireless sensor networks," in *2011 Third Int. Conf. Comput. Intell. Model. Simul.*, 2011, pp. 308–311.

[33] A. S. K. Pathan, H. W. Lee, and C. S. Hong, "Security in wireless sensor networks: issues and challenges," in *2006 8th Int. Conf. Adv. Comm. Tech.*, vol. 2, 2006, p. 1048.

[34] K. Adedeji, Y. Hamam, B. Abe, and A. M. Abu-Mahfouz, "Improving the physical layer security of wireless communication networks using spread spectrum coding and artificial noise approach," in *Southern Africa Telecommunication Networks and Applications Conference*, George, Western Cape, South Africa, Sep. 2016, pp. 80–81.

[35] C. Alcaraz, P. Najera, J. Lopez, and R. Roman, "Wireless sensor networks and the internet of things: Do we need a complete integration?" *1st Int. Work. Secur. Internet Things*, pp. 1–8, 2010.

[36] Z. Shu, J. Wan, D. Li, J. Lin, A. V. Vasilakos, and M. Imran, "Security in software-defined networking: Threats and countermeasures," *Mob. Networks Appl.*, vol. 21, pp. 764–776, Jan. 2016.

[37] I. Alsmadi and D. Xu, "Security of software defined networks: A survey," *Comput. Secur.*, vol. 53, pp. 79–108, 2015.

[38] B. B. Letswamotse, K. M. Modieginyane, and R. Malekian, "SDN based QoS provision in WSN technologies," in *Southern Africa Telecommunication Networks and Applications Conference (SATNAC)*, George, South Africa, Sep. 2016.