Security Analysis of OpenRadio and SoftRAN with STRIDE Framework

Daniel Magin[1], Rahamatullah Khondoker[2], Kpatcha Bayarou[2]
[1] Department of Computer Science, TU Darmstadt, Germany
[2] Fraunhofer SIT, Rheinstr. 75, Darmstadt, Germany

***Abstract*—The field of mobile communication is a fast evolving area. New protocols and technologies are developed, for example, LTE or MIMO. However, problems arise in implementing and managing these technologies, for example, today a Radio Access Network (RAN) provider buys new hardware for each new standard. For speeding up this process and generating new optimization possibilities such as dynamic prioritizing of traffic, the use of SDN can become a handy. However, cellular networks are a very critical and complex infrastructure, therefore, they should be managed automatically. Several tools are summarized here that provide SDN features for wireless networks. Some of these tools are Odin, OpenWiFi, OpenRoads and SDNAN. Since none of these tools focus on cellular networks, we concentrate on OpenRadio and SoftRAN in this paper, specifically, the security of these technologies is evaluated and suggestions to improve their security are given. Both of these tools were developed to improve manageability of cellular networks. OpenRadio implements radio protocol stacks for GSM, LTE or Wi-Fi technologies in software and makes it easy to upgrade and optimize base stations without the need to change their hardware. SoftRAN abstracts several base stations in the same geographical area into one logical base station. This provides optimization over different base stations, including avoiding interference between them. The result of the survey of OpenRadio and SoftRAN is, that both tools do not consider any security aspects in their design proposals. However, by implementing standard security features, including encryption and message authentication, these gaps can be closed easily. The main difficulty when implementing security features in applications like OpenRadio or SoftRAN is to meet hard time constraints. These time constraint come from requirements of the physical layer like timeouts, which would lead to connection losses.**

I. Introduction

The importance as well as the complexity of cellular networks are growing and managing such networks becomes more and more difficult. Furthermore, the implementation of new standards like LTE is very expensive. A solution for such problems can be the use of Software Defined Network (SDN) based solutions, to abstract the logic in the network from the hardware.

In the past couple of years, several new technologies and applications in the area of SDN for wireless networks have been developed. A summary of the applications mentioned here can be found in Table I. Each application is described by its name, technology used, its functionality, and its maturity. The wireless technologies that are used by these applications are WLAN, Bluetooth, and cellular technologies such as LTE. Maturity is described the terminologies such as prototype (working example available), deployed (at least one running and used installation), proof-of-concept (principle demonstrated but no prototype), conceptual (no implementation at all), and commercial product.

TABLE I
Overview of SDN related applications for wireless networks.

SDN Application	Technology	Functionality	Maturity
Odin [1]	WLAN	Handover between APs	Prototype
OpenWifi [2]	WLAN	Separation of access, authentication and accounting	Prototype
OpenRoads [3]	WLAN	Testbed for wireless network experiments	Deployed
SDNAN [4]	WLAN, Bluetooth	Ad-hoc networks build by smart phones	Prototype
OpenRadio [5]	Cellular	Implement protocol stack in software	Proof of Concept
SoftRAN [6]	Cellular	Abstract several base stations into one logical base station	Conceptual
OpenRF [7]	WLAN	Interference management among WLAN MIMO devices	Deployed
HP Cloud Network Manager [8]	WLAN	Configuration of access points via cloud application	Commercial Product
HP FlexCampus [9]	WLAN	Converged Infrastructure	Commercial Product

As the focus of this paper lies only on wireless technologies which improve efficiency, manageability or maintainability of cellular networks, OpenRadio and SoftRAN will be evaluated here using the STRIDE framework. It is seen in Table I that other applications use technologies including WLAN, and Bluetooth.

The structure of this paper is as follows: In section II, the STRIDE framework will be briefly explained, as it is used later to evaluate possible security threats to OpenRadio and SoftRAN which are described in section III and section IV respectively. In these sections, the technologies are briefly described and then the evaluation with STRIDE is performed. Afterward, in section VI, some mitigation methods for the uncovered issues are discussed and an overall conclusion is drawn.

II. Methodology

In this paper, STRIDE [10], a threat modeling framework developed by Microsoft, is used to evaluate the security of OpenRadio and SoftRAN. STRIDE is used because it is developed to analyze the architecture and design of a solution even without having any implementation. It does not build on a risk model which would highly depend on the actual usage environment and customer needs. So it fits perfect for analyzing solutions in a design or prototype state like OpenRadio and SoftRAN are.

STRIDE builds on two components: A threat model and Data Flow Diagrams (DFDs)[1]. STRIDE itself is an acronym for the parts of the threat model, which are defined as follows:

Spoofing is an impersonation of someone else.

Tampering is an unauthorized change of data (either in transit or when stored), which is a threat against integrity.

Repudiation mean that someone can deny an action after it is performed.

Information Disclosure is a leakage of confidential information to unauthorized persons, which is a threat against confidentiality.

Denial of Service (DoS) attacks prevent a system from operating with the necessary performance, which is a threat against availability.

Elevation of Privileges means that a user of the system gets higher privileges than he is intended to have.

To analyze a solution it is decomposed into components of a DFD and then each component is analyzed against the specific threats for its type. The five types of components of a DFD can be found in Table II. The threat matrix can be found in Table III.

As soon as the threats to the solution are identified a proper mitigation, based on the use cases of the threatened components, their type and external constraints, like time constraints, can be defined. The mitigation methods proposed in this paper are only suggestions and there is no guarantee that they are the best and/or the most secure solutions. Further testing must be performed before implementing any solutions in a productive environment.

In the next chapters the methodology of STRIDE will be applied to OpenRadio and then to SoftRAN

III. OpenRadio

OpenRadio “is capable of realizing modern wireless protocols (WiFi, LTE) on off-the-shelf DSP chips while providing flexibility to modify the PHY and MAC layers to implement protocol optimizations.” [5]

In OpenRadio the wireless protocol stack is divided into a processing and a decision plane, that can both be programmed independently. This makes it possible to implement new wireless protocols as well as optimizing existing ones without changing the underling hardware. Especially for large networks with a lot of base stations like RANs, this can accelerate the speed of adaption of new protocols. Further details about OpenRadio can be found in [5].

A. DFD of OpenRadio

To evaluate OpenRadio with STRIDE it is first necessary to build the according DFD which is shown in Figure 1.

Even while OpenRadio consists of different processes and hardware components, they are all contained in one single physical box with a single interface to the outer world. Therefore it can be modeled as one process in the context of STRIDE. This is shown in Figure 1 where both components are enclosed in a single process. The two actors in the DFD are the radio hardware itself which sends and receives the physical radio signals and the Operator, who programs OpenRadio. The data flows are therefore between the Operator and OpenRadio and between OpenRadio and the radio hardware. At last, there is a trust boundary between the radio hardware and the OpenRadio process on one side and the Operator on the other side. This reflects the physical constellation of the architecture. To minimize the costs of maintenance, it would be the best to enable a remote programmability of OpenRadio so that the Operators can change the behavior of a base station without the need to visit it physically.

B. Evaluation of OpenRadio using the STRIDE methodology

For the following evaluation one assumption is made: The radio hardware and the OpenRadio appliance are contained in the same physical box and are immune to physical manipulations. This implies that the only interface to the outer world goes over the OpenRadio process. Therefore it is not necessary to evaluate the radio hardware nor the data flow between the OpenRadio process and the radio hardware. Furthermore it is assumed that the solution is not vulnerable from client side. That means that it is not possible to exploit the system using the radio interface between mobile phones and the base station.

The threats according to Table III and mitigation methods for these threats are discussed in the following. A summary of the evaluation can be found in Table IV at the end of this section.

[1] The content of this chapter is a very brief summary of [10].

TABLE II
COMPONENTS OF DFDS. (TABLE AND DESCRIPTIONS TAKEN FROM [10])

Component	Symbol	Description
Data Flow	One way arrow	Data flows represent data in motion over network connections, named pipes, mail slots, RPC channels, and so on.
Data Store	Two parallel horizontal lines	Data stores represent files, databases, registry keys, and the like.
Process	Circle	Processes are computations or programs run by the computer.
Interactors	Rectangle	Interactors are the end points of your system: the people, Web services, and servers.
Trust Boundary	Dotted line	Trust boundaries [...] represent the border between trusted and untrusted elements.

TABLE III
THREAT MATRIX (TAKEN FROM [10])

Component	Spoofing	Tampering	Repudiation	Information Disclosure	DoS	Elevation of Privilege
Data Flows		X		X	X	
Data Stores		X		X	X	
Processes	X	X	X	X	X	X
Interactors	X		X			

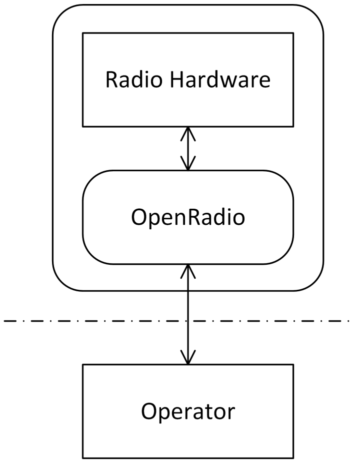

Fig. 1. DFD of OpenRadio

As first component, the OpenRadio process is discussed. The threats and mitigation methods for this component are:

Spoofing An attacker can impersonate an operator and is then able to change the programming of the process. This can be mitigated by the use of proper authentication mechanisms e.g. Kerberos [11] or the use of a proper certification system.

Tampering Either by an attacker or accidentally the programming of the process could be changed. One way to prevent this is the use of trusted computing as described in [12].

Repudiation This threat has to dimensions in the context of cellular networks. The one threat is, that it is possible to deny that someone has used the service provided by the base station. This would lead to problems with billing. The other threat is, that someone can make changes to the programming and then deny it. A proper (secured against tampering) logging appliance can prevent both threats.

Information Disclosure Competitors could gain an advantage if they could get knowledge of the programming of the OpenRadio implementation and implement the same optimizations into their own OpenRadio installations. This threat can be mitigated by denying the unauthorized download of any data from the OpenRadio appliance which implies the need of proper authentication (see Spoofing above). Encryption may also be a solution but in this case a trade-off between security, performance and complexity and therefore operational safety must be resolved.

Denial of Service A DoS attack would cause the whole base station to stop working. This would lead to unavailability of the cellular network in the area and therefore would cause significant monetary and reputation damage. By separating the authentication process form the rest of the OpenRadio process, a DoS attack could only affect the ability to change the programming of a base station but not its normal operation. This can be done by using an independent gateway for authentication. If the gateway is unavailable due to a DoS attack, the normal operation of OpenRadio is not affected.

The next part is the data flow from the Operator to the OpenRadio process. Threats and mitigation methods for this dataflow are:

Tampering An attacker can change a new program when in transit. This can be mitigated by using for example IPSec's

Authentication Header (AH) as described in [13].

Information Disclosure The threat here is the same as for the OpenRadio process. The mitigation here would be the use of IPSec's Encapsulating Security Payload (ESP) as mentioned in [13] or TLS [14].

Denial of Service A DoS attack would prevent an operator from changing the programming of the OpenRadio process. This is a minor problem, if a proper functional programming is already in place so that OpenRadio operates correctly. After the initial programming this should always be the case as every update should be tested before rolled out to operational equipment. Therefore one can accept that risk.

The last component that must be evaluated is the operator, which is vulnerable to the following threats:

Spoofing If an attacker can impersonate an OpenRadio appliance, he can sniff the programming of the appliance, when an operator sends an update. Furthermore an attacker would be able to present the operator forged health information of the appliance, which would lead to false decision. This would be prevented by the use of IPSec [13] as mentioned above.

Repudiation If an evil operator acting as an insider attacker can deny that he made changes to the programming of an OpenRadio appliance, he could hide malicious actions. A proper logging can mitigate this threat.

TABLE IV
STRIDE THREAT MATRIX FOR THE COMPONENTS OF OPENRADIO. "X" DENOTES THAT A THREAT CAN BE MITIGATED AND "O" MEANS THAT THE THREAT MAY BE IGNORED.

Type	Component	Threats					
		S	T	R	I	D	E
Interactors	Radio Hardware	X		X			
	Operator	X		X			
Process	OpenRadio	X	X	X	X	X	X
Data Flows	OpenRadio ↔ Radio Hardware		X		X	X	X
	Operator ↔ OpenRadio		X		X	O	X

IV. SoftRAN

The goal of SoftRAN is to abstract several micro- and femtocells of a RAN in the same geographical area into one logical macro-cell. The idea is, that the base station of each cell will be controlled centrally to achieve load balancing and to reduce interference between neighboring cells. An overview of the architecture of SoftRAN is given while building its DFD. Details of SoftRAN can be found in [6].

A. DFD of SoftRAN

In order to evaluate SoftRAN like OpenRadio it is necessary to build the according DFD, which is visualized in Figure 2 and described in more detail in the following.

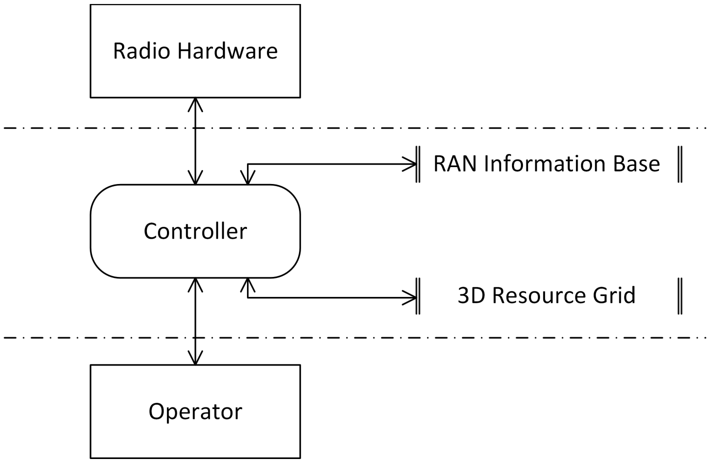

Fig. 2. DFD of SoftRAN

In the context of SoftRAN the radio hardware is not one single base station but all base stations manged by one controller. The controller is a logically centralized process that makes decisions affecting the behavior of the manged base stations like their transmission power and client handovers. These decisions are made based on the information from the RAN information base and the 3D resource grid. The first one includes information about neighboring cells (important to avoid interference), flow records[2], subscribers and preferences set by the network operator. The 3D network grid is a real-time representation of the manged network consisting of a base station identifier, its time slots and its transmission frequency.

The data flows in the DFD are set accordingly. The trust boundaries are between the radio hardware and the controller and between the controller and the operator. The first trust boundary is due to the fact, that one controller should manage several base stations in a centralized manner. Therefore the controller and the base stations are not located in the same physical nor logical box. The same holds true for the operator, who would supervise remotely more than one controller.

both data stores may logically be contained in the controller but could be physically installed on different servers (e.g. on a database server).

[2] For simplicity one can assume flows as clients. Actually one flow is one data flow from a client to a network, so if someone types a SMS and listens to music at the same time, this client would have (at least) two flows assigned.

B. Evaluation of SoftRAN using the STRIDE methodology

The evaluation of SoftRAN is performed in the same way as the one for OpenRadio. The ordering follows Figure 2 from top to bottom. A summary of the results can be found in Table V at the end of this section.

So the first component is the radio hardware:

Spoofing If an attacker is able to claim that he is a controller, he would take full control over the base station. To prevent this a proper authentication like the use of certificates should be implemented. It is important to ensure a bidirectional authentication because if an attacker can impersonate a base station, he is able to inject malicious information into the controller.

Repudiation If spoofing is covered, the threat of repudiation can be ignored, as the only component that can make changes to the base station's behavior is the controller and only base stations can send data to the controller.

The next component is the data flow between controller and radio hardware. This is a bidirectional data flow because while the controller sends commands to the base stations, it makes its decisions based on information from the base stations. The threats to this data flow are:

Tampering If the commands sent from the controller can be changed in transit, an attacker could sent arbitrary commands to base stations. Otherwise if the data sent from the radio hardware to the controller can be changed, the controller would make decisions based on wrong information. This could lead to an indirect control of base stations. However it is necessary to secure this data flow against tampering and this can be done by the use of proper Message Authentication Code (MAC) protocols like implemented in IPSec's AH [13].

Information Disclosure The commands for the base stations are not threatened by information disclosure since this are strongly mutable commands only valid for milliseconds. But the data sent to the controller will contain user data which has to be protected against information disclosure. Therefore this direction must be secured by the use of proper encryption protocols like IPSec's ESP [13] or TLS [14].

Denial of Service A DoS attack on the data flow from the base station to the control would lead to an interruption of the availability of this base station. This could be covered by neighboring base stations. That could be performed by the controller if the link to the base station is lost. If only the data flow from the controller to the base station is unavailable the base station should be able to go into a fallback mode and operate as there were never be a centralized controller.

The two data stores can be evaluated together because threats and consequences are similar.

Tampering An unauthorized change to any of the data stores would lead to wrong decisions by the controller. Since both data stores change very often due to their function a file integrity monitoring is not usable. So it is necessary to perform proper authentication and authorization to anyone that wants to perform changes on this data stores.

Information Disclosure This threat can be ignored for the 3D resource grid, but in the RAN information base are flows saved which contain personal data. This should be covered already by the mitigation methods for tampering but personal data should always be stored encrypted.

Denial of Service If any of the data stores is unavailable the controller can not operate. So it is critical that the data stores are always available. This can be achieved by a high availability architecture using redundant hardware and hot backups. Since the data stores are only used internally an external attack is unlikely to happen and a hardware fail is the bigger threat.

The data flows between the two data stores and the controller can be assumed to be secure since data flows between servers at the same site should be secure. If this is not the case the same results as for the data flow between the base stations and the controller holds true.

The controller is the most critical component because if an attacker can get control over this process he has full control over the whole macro-cell. This would mean control for example over the RAN of a whole city. So a close evaluation of this component is important:

Spoofing If an attacker can spoof a base station, he can send wrong information to the controller which would lead to wrong decisions. The same holds true for the data stores. Much worse is it, if someone can spoof to be an operator. This would immediately lead to full control over the network. Mitigation methods here are the use of certificates as mentioned for the radio hardware or protocols like Kerberos [11] and in the sense of security in depth for example plausibility checks of IP addresses. This means that a data store will never have an external IP and the IP range of operators is very limited.

Tampering The consequences of tampering can be compared to the one of spoofing. Since the code controller process itself does not change very often, possible mitigation methods could be the use of Trusted Platform Modules (TPMs) [12] and file integrity monitoring of the code of the controller process.

Repudiation If the controller process gets corrupted in a malicious way it could manipulate the whole RAN. If it is possible for the controller process to deny that the change was made by itself, a debugging and fixing of the problems gets very hard. This can be avoided by the implementation of secure logging for any external communication.

Information Disclosure As the controller decides for every flow how to handle it, it also sees the personal information contained in that flows. Furthermore it has access to the personal information stored in the RAN information base. This shows that an information disclosure threat can cause significant consequences. As the controller is absolutely critical for SoftRAN to operate it should be kept as simple as possible. Therefore solutions like

RAM-Encryption are not an option. The simplest way to protect the information processed by the controller is to limit access to the server on which it is running to the very least possible. Also only the controller process should run on this server. This would reduce the risk of a crash of the server due to other services as well as it minimizes the attack surface.

Denial of Service If the controller is under an overload condition for example due to a Distributed DoS (DDoS) attack the complete RAN manged by this controller would become unavailable. This threat can not be ignored or mitigated in a simple way like for the other mentioned DoS scenarios. DDoS attacks and defenses are for example described in [15]. An other scenario is that only the controller process is unavailable due to a DoS attack, but the network itself just works fine. As mentioned above the base stations should go to a fallback mode and operate on their own so that the service for the end users can be delivered nevertheless. Besides from a non optimal mode of operation this could cause problems if the accounting and billing process is also integrated in the controller. Due to the principle of separation of concerns this should not be the case. Therefore this threat can be ignored **iff** the base stations are able to operate on their own if the controller is not available. Otherwise it is absolutely critical to keep the process running. So high availability features like redundant servers and hot backups must be in place.

Elevation of Privilege In [6] no user roles for SoftRAN are described but in a real world implementation it would make sense if there were at least two roles: One for supervisors, who can only monitor the process but must not take any changes to its programming, and one for administrators who can implement patches and updates. The control of SoftRAN in the normal operation is performed over rules in the data stores and not by changing the programming of the controller. By installing a malicious patch and attacker could get full control over the controller. This could be performed either by a supervisor who is able to elevate his privilege to the one of an administrator or by an administrator. The mitigation for this would be the use of the four eye principle so that it is necessary that a second administrator approves an installation before it is executed.

The last component to evaluate is the operator. This component is vulnerable to the following threats:

Spoofing If an attacker can claim to be the controller front-end he could cause the operator to make malicious changes to the rules for the controller by showing him forged information. A strong authentication like mentioned above would mitigate this threat.

Repudiation An evil operator can make malicious changes to the data stores or the controller itself and later deny it, what would make the investigation of this incident later on much harder. As mentioned above an appropriate logging can mitigate this threat.

TABLE V
STRIDE THREAT MATRIX FOR THE COMPONENTS OF SOFTRAN. ”X” DENOTES THAT A THREAT CAN BE MITIGATED.

Type	Component	Threats					
		S	T	R	I	D	E
Interactors	Radio Hardware	X		X			
	Operator	X		X			
Process	Controller	X	X	X	X	X	X
Data Flows	Controller ↔ Radio Hardware		X		X	X	X
	Operator ↔ Controller		X		X	X	X
Data Stores	RAN Information Base		X		X	X	
	3D Resource Grid		X		X	X	

V. Related and Future Work

Apart from the work mentioned in this paper, the STRIDE framework has also been used to accomplish security analyses of SDN SBI (South Bound Interface) protocols including OpenFlow, OF-Config, and OVSDB [16]; SDN architectures including PCE, 4D, and SANE [17]; SDN security applications including OpenFlow Random Host Mutation and Resonance [18]; and SDN applications for monitoring and measurement including sFlow and BigTap [19]. All of these work not only unveil potential security threats but also provide suggestions to tackle those threats using existing well-defined mechanisms. Future work will be to verify the threats using penetration testing with or without considering the suggestions. All of these work fall into the category of security-centric SDN, additionally, network security can be improved by using SDN such as OrchSec architecture [20].

VI. Discussion and Conclusion

From a technology point of view, OpenRadio and SoftRAN can complement each other. OpenRadio allows flexible programming of base stations, and can therefore, accelerate the adoption of new protocols or optimizations. Furthermore, it provides an Application Programming Interface (API) for base stations. This API could be used by SoftRAN to communicate with the base stations it manages. SoftRAN itself provides optimizations for clusters of base stations including avoiding of interference between base stations and load balancing.

There is one big constraint to consider when implementing security features into the technologies: Both have time critical parts with very hard time constraints. Therefore, the security features must not cost much execution time.

For OpenRadio, this is not a big issue as it is assumed that all hardware is contained in one Box with only two interface to the outside: the radio interface and the operator interface. If the operator is a human being, there are no hard time constraints in this direction. So strong cryptography and authentication techniques can be used in this scenario.

The same holds true for the operator interface of SoftRAN. But the communication between the SoftRAN controller and the base stations is very time critical in the scale of milliseconds. To solve this problem, it is first assumed that confidentiality is not an issue as it is handled on a higher layer where necessary. However, integrity is an issue which has been described in the evaluation. There are standard techniques to ensure the integrity of messages, for example, the use of MACs. Of course, using these technologies costs execution time. However, when implemented correctly this can be covered by upgrading the hardware so that the time constraints can be fulfilled.

Both OpenRadio and SoftRAN provide opportunities but security is not yet part of their design. This should be changed because security by design is better than fixing security issues afterward in a running installation. This paper showed that doing so is still possible since known security techniques can be used to avoid the most common threats. Furthermore, since OpenRadio or SoftRAN are not deployed in productive environments yet, it is not a problem to change their specification and no backward compatibilities must be considered. Therefore, by integrating suggested security methods in OpenRadio and SoftRAN, improved and secure manageability of cellular networks are expected.

References

[1] L. Suresch, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards Programmable Enterprise WLANs with Odin," in *HotSDN'12*, 2012, pp. 115–120.

[2] K.-K. Yap, Y. Yiakoumis, M. Kobayashi, S. Katti, G. Parulkar, and N. McKeown, "Separating Authentication, Access and Accounting: A Case Study with OpenWiFi," in *OPENFLOW-TR-2011-1, Open Network Foundation*, 2011.

[3] K.-K. Yap, M. Kobayashi, D. Underhill, S. Seetharaman, P. Kazemian, and N. McKeown, "The Stanford OpenRoads Deployment," in *WiNTECH'09, ACM New York*, 2009, pp. 59–66.

[4] P. Baskett, Y. Shang, W. Zeng, and B. Guttersohn, "SDNAN: Software-defined networking in ad hoc networks of smartphones," in *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*, 2013, pp. 861–862.

[5] M. Bansal, J. Mehlman, S. Katti, and P. Levis, "OpenRadio: A Programmable Wireless Dataplane," in *Proc. HotSDN'12*, 2012.

[6] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "SoftRAN: Software Defined Radio Access Network," in *Proc. HotSDN'13*, 2013.

[7] S. Kumar, D. Cifuentes, S. Gollakota, and D. Katabi, "Bringing Cross-Layer MIMO to Todays Wireless LANs," in *ACM SIGCOMM*, 2013, pp. 387–398.

[8] Hewlett-Pakcard Development Company, L.P., "HP Cloud Network Manager Software Series," Available: http://h17007.www1.hp.com/us/en/networking/products/wireless/HP_Cloud_Network_Manager_Software_Series/index.aspx#.VN4FXy4yFKq, 2015, Accessed on 02/13/2015.

[9] HP, "HP Networking Campus Solutions for FlexCampus," Available: http://h17007.www1.hp.com/us/en/networking/solutions/campus-lan/index.aspx#.VN4GlS4yFKo, 2015, Accessed on 02/13/2015.

[10] S. Hernan, S. Lambert, T. Ostwald, and A. Shostack, "Uncover Security Design Flaws Using The STRIDE Approach," Available: http://msdn.microsoft.com/en-us/magazine/cc163519.aspx#S2, 2006, Accessed on 02/13/2015.

[11] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, *RFC 4120: The Kerberos Network Authentication Service (V5)*, Available: http://www.ietf.org/rfc/rfc4120.txt, 2005.

[12] Trusted Computing Group, "Trusted Computing," Available: http://www.trustedcomputinggroup.org/trusted_computing, 2014, Accessed on 02/13/2015.

[13] S. Kent and K. Seo, *RFC 4301: Security Architecture for the Internet Protocol*, Available: https://tools.ietf.org/html/rfc4301, 2005.

[14] T. Dierks and E. Rescorla, *RFC 5264: The Transport Layer Security (TLS) Protocol Version 1.2*, Available: http://www.ietf.org/rfc/rfc5246.txt, 2008.

[15] R. Kenig, D. Manor, Z. Gadot, and D. Trauner, *DDoS Survival Handbook*. Radware, 2013.

[16] M. Brandt, R. Khondoker, R. Marx, and K. Bayarou, "Security Analysis of Software Defined Networking Protocols – OpenFlow, OF-Config and OVSDB," in *IEEE ICCE 2014, Special Session on SDN*, July 2014.

[17] D. Klingel, R. Khondoker, R. Marx, and K. Bayarou, "Security Analysis of Software Defined Networking Architectures – PCE, 4D and SANE," in *ACM AINTEC*, November 2014.

[18] M. Tasch, R. Khondoker, R. Marx, and K. Bayarou, "Security Analysis of Security Applications for Software Defined Networks," in *ACM AINTEC*, November 2014.

[19] P. Dauer, R. Khondoker, R. Marx, and K. Bayarou, "(Accepted) Security Analysis of Software Defined Networking Applications for Monitoring and Measurement – sFlow and BigTap," in *The 10th International Conference on Future Internet Technologies (CFI)*, June 2015.

[20] A. Zaalouk, R. Khondoker, R. Marx, and K. Bayarou, "OrchSec: An Orchestrator-Based Architecture For Enhancing Network-Security Using Network Monitoring And SDN Control Functions," in *IEEE NOMS*, May 2014.