Enhancing SDN Security for IoT-related deployments through Blockchain

C. Tselios, I. Politis and S. Kotsopoulos
Wireless Telecommunications Laboratory
University of Patras
Email: {tselios, ilpolitis, kotsop}@ece.upatras.gr

***Abstract*—The majority of business activity of our integrated and connected world takes place in networks based on cloud computing infrastructure that cross national, geographic and jurisdictional boundaries. Such an efficient entity interconnection is made possible through an emerging networking paradigm, Software Defined Networking (SDN) that intends to vastly simplify policy enforcement and network reconfiguration in a dynamic manner. However, despite the obvious advantages this novel networking paradigm introduces, its increased attack surface compared to traditional networking deployments proved to be a thorny issue that creates skepticism when safety-critical applications are considered. Especially when SDN is used to support Internet-of-Things (IoT)-related networking elements, additional security concerns rise, due to the elevated vulnerability of such deployments to specific types of attacks and the necessity of inter-cloud communication any IoT application would require. The overall number of connected nodes makes the efficient monitoring of all entities a real challenge, that must be tackled to prevent system degradation and service outage. This position paper provides an overview of common security issues of SDN when linked to IoT clouds, describes the design principals of the recently introduced Blockchain paradigm and advocates the reasons that render Blockchain as a significant security factor for solutions where SDN and IoT are involved.**

***Index Terms*—Cloud, Virtualization, SDN, IoT, Blockchain.**

I. Introduction

In today's integrated and connected world most online activities take place in business networks based on cloud computing infrastructure and broad networking deployments that span national, geographic and mostly jurisdictional boundaries. Those networks are typically attached through dedicated interfaces which are designed to provide the highest possible degree of flexibility, scalability, expandability and security to all interconnected entities [1]. The efficient entity interconnection is made possible through an emerging networking paradigm, Software Defined Networking (SDN) that intends to disrupt vertical integration, thus separating the core networks control logic from the underlying routing and switching elements and hopefully create a logically centralized controller to further simplify policy enforcement and network reconfiguration in a dynamic manner [2]. In SDN, the aforementioned networking elements simply operate as packet forwarders according predefined policies, created or modified in a separate controller and then pushed to the edges of the network. This approach greatly helps network providers to dynamically change network configuration on-the-fly, in a centralized way via the controller, without the need of independently accessing and reconfiguring individual devices, scattered across the whole network. In this way, network upgrade along with the necessary adjustments can be completed within minutes, tackling potential issues in a near-real time fashion.

The ability of being able to rapidly change network configuration as provided by SDN proved to be of disruptive nature for operators handling vast cloud computing deployments with thousands of datacenters globally. However, despite the obvious advantages this novel networking paradigm introduces, there are several issues that somehow hold back its undisputed dominance over legacy solutions. One of the most important drawback of SDN is its increased attack surface compared to traditional networking deployments and the increased effect any successful attack will have, once the controller is compromised. As stated in [3], SDN brings a rather interesting dilemma, whether or not is a promising evolution of networking architectures preferable over a dangerous increase of the threat surface. One could argue that it is possible to preserve the benefits of the first by carefully securing the later but as we intend to show in this position paper, when new interesting applications and verticals such as the Internet of Things (IoT) rise, it is essential to step back and try to address the security issue as a whole, in a radical way.

The rest of this paper is organized as follows: Section II identifies the main security challenges of SDN networking while Section III describes the additional security considerations introduced by deployments following the IoT paradigm. Section IV presents the basic elements of Blockchain and how this technology can be used for enhancing network security through a holistic use case. Finally, Section V concludes the paper and discusses future research and implementation steps.

II. SDN Security Challenges

The design principles of SDN impose the necessity of such networks to be fully controlled by software and ensure the existence of a central network intelligence inside a dedicated node, the controller.

A. Application Layer

Given the fact that most network functions can be implemented as or utilized by SDN applications [4], potential execution of malicious software may cause severe damage to

978-1-5386-3285-7/17/$31.00 ©2017 IEEE

the topology throughout its layers. The lack of standards and guidelines for software development is also possible to induce security policy collisions or even inherent interoperability limitations, since many players, from network providers to third-party personnel, may simultaneously operate inside the same SDN-controlled infrastructure. The dominant security challenges in the application layer are illustrated in Figure 1 by vectors (1) and (2) respectively and are also listed below:

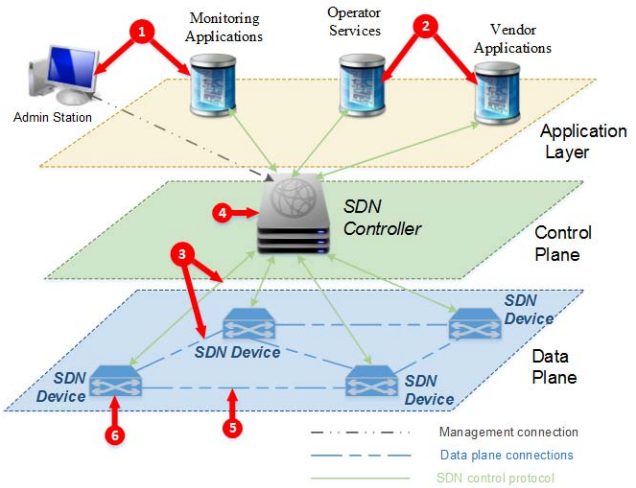


Fig. 1: Potentially Vulnerable Surfaces in SDN

1) Unauthorized/Unauthenticated Access Control: SDN administrative stations, poorly designed third party applications deployed in the top layer and all interconnected SDN entities are subdue to various types of attacks, which aim to gain access to their core functions, thus being able to parse the SDN controller in order to inflict the maximum amount of damage to the network. Practices such as brute force attacks [5] also existing in legacy network deployments are now enhanced further by the inherent privilege of all SDN applications to access the controller and gain direct access to network resources and configuration mechanisms. Since SDN networks are easily re-programmable from a single location, any security gap is considered potentially hazardous [6]. Although it is possible to use protocols requiring dual factor verification for preventing malicious nodes to automatically launch attacks, or certain recovery mechanisms to guarantee return to a reliable state after the attack is handled, dealing with access control and application accountability is a real security challenge.

2) Improper network rules insertion: Once a malicious application is deployed or a benevolent one is compromised, it is possible to start generating traffic, prevent proper signaling and packet delivery, or try to enforce false flow rules initially towards the neighboring nodes but sooner or later will attempt reaching the full extent of the SDN network. A valid check of whether an application is compromised is a difficult task, especially against third-party applications some of which with additional nested applications operating in a unified mode. Such a malicious application constantly consuming network resources could have the same inimical effect over the network portion it resides, to a compromised controller with a corrupted flow rule for the particular network chunk.

B. Control Plane

Fortifying the SDN Control plane against potential threads is considered a decisive strategy against actions that could possibly grant the attacker total control over the core networking infrastructure. For instance, an ill designed or malicious piece of software could become a potential security dent allowing controller, router or switch reprogramming, improper traffic generation and connectivity loss due to forwarding table eradication. The most common control layer attacks are presented by vectors (3) and (4) in Figure 1.

1) Denial-of-Service (DoS) attacks: DoS and Distributed DoS are extremely threatening for the SDN controller due to the control and data plane separation introduced to the topology by design. An attacker could exploit the communication channel between the two planes, insert forged or faked traffic flows that require additional actions by the controller or other network entities, rendering them unavailable for legitimate users. Such an attack is presented in [7], where a network scanner manipulates flow response times to identify underlying SDN nodes. Once an SDN network is discovered, specially formulated flow requests are transmitted towards the controller via the datapath. Increasing the number of flows in the datapath leads to an analogous inflation of flow setup requests towards the controller by the switches, eventually causing its service interruption. DoS attacks may also be deployed by continuously transmitting IP packets with random headers to set the controller in a non-responsive state [8] or by signal flooding of network nodes with limited resources, given the fact that a chain is as strong as is weakest link. One could argue in favor of redundant SDN controllers operating in the same network, however cascading failures of multiple controllers through DDoS attacks have been reported [9], therefore some auxiliary detection mechanism is rather necessary. For tackling DDoS attacks [3] proposes the use of oligarchic trust models with multiple trust-anchor certification authorities or a dynamic device association mechanism for trust establishment between the control and the data plane devices. Alas, such solutions might prove to be obsolete once the number of interconnected nodes rises above a certain threshold, or lightweight messaging protocols of different orientation and scope come into play.

2) SDN Controller attacks: The particular control layer attack is a characteristic of SDN networking era, since in legacy networking, there was not such a pivotal node that once tampered could pose a threat for all interconnected nodes. Given the fact that third parties are allowed to deploy applications on the controller, defective or virulent applications are rendered capable of re-configuring the entire network, since controllers only provide abstractions that translate into issuing configuration commands to the underlying infrastructure [3]. Countermeasures to such attacks include but are not limited to replication techniques able to detect, remove or isolate peculiar behavior, extensive component diversity eliminating the single

point of failure case through heterogeneous infrastructure and mechanisms for periodic system restoration to a stable and reliable state. However, as in the DoS case, network scale-out to a large number of interconnected nodes could result to content/data loss, in scenarios that the controller needs to constantly reboot itself. This renders such deployments improper for mission-critical applications or verticals that incorporate continuous data collection from remote sources.

C. Data Plane

Data and control plane decoupling leads to a consequent simplification of all forwarding devices (i.e. routers, routers etc), which now become simple packet handling elements, remotely operated via specialized interfaces. It is now possible to reconfigure any virtual switch, router or access point using a secure communication channel, in a flow-based forwarding manner. The data plane programmability is conducted in a granular level, enabling unprecedented flexibility limited only by the capabilities of the implemented flow tables [10]. Despite the increased elasticity data and control plane separation offers, there are still certain security issues in the otherwise robust and simplistic data plane, that need to be tackled as efficiently as possible. The most common data plane attacks along with their targets are presented by vectors (5) and (6) in Figure 1.

1) Forged switch flow rules and Flooding attacks: As described in [10] Openflow networks, the controller connects to the networks switches and installs flow rules in dedicated tables. These rules are installed either proactively, before a new host sends any ingress packets, or reactively, where the installation of flow rules occurs together with the initial packet handling request on behalf of the host. No matter the installation approach, a switch only incorporates a finite (and sometimes limited) number of flow tables where rules can be stored. Provided that in the SDN era switches lack certain intelligence and decision making capability, the dominant security challenge in the data plane is to actually identify genuine flow rules and discard the malicious ones. This inherent transformation of switches to simple forwarding nodes, renders them susceptible to counterfeit flow rules [11]. In addition, it is also possible that the attacker's target is the actual switch buffer and the total number of rules it can store. Due to the limited resources for unsolicited flow buffering, data plane nodes are prone to saturation attacks. Such types of attacks where a malicious node intercepts the proper communication channel and enforces its own policies in the network, bear many similarities to the legacy man-in-the-middle attack [12] and most of the times are categorized as such.

2) TLS and TCP related issues: The configuration complexity of Transport Layer Security (TLS) [13] rendered the specific security enhancement optional in the recent Openflow versions [11]. Originally, a site-wide certificate was generated, along with separate controller and switch certificates which after being signed with the site-wide one were forwarded to all nodes. This approach is no longer supported and switches are now more vulnerable. However, many advocate against using TLS for securing the network, since [14] demonstrated that its usage does not prevent TCP-level protection. This simply means that deployments based on Openflow are utterly exposed to man-in-the-middle attacks due to connectivity and lack of authentication in the plain-text TCP channel.

III. Security Considerations in IoT

The Internet of Things is a novel paradigm which gained significant momentum over the last few years. It relies on the notion of an ubiquitous network of seamlessly interconnected, yet vastly divergent devices, capable of collecting, aggregating and analyzing data despite the potential geographical diffuse of its nodes. This heterogeneous amalgam covers a diverse technological range, being able of sensing and communicating in an efficient end-to-end way. Unquestionably, the main strength of the IoT idea is the high impact it will have on several aspects of everyday-life and behavior of potential private and business users [15]. For many organizations IoT will be the cornerstone of their digital strategies, since its technologies and principles will affect a wide range of technical areas such as architecture and network design but also the core business strategy and risk management [16].

Up to this date, there is no commercial overall IoT platform or comprehensive solution. According to [16], IoT remains an immature domain where product and technology categories arent yet clearly established and despite the efforts of researchers towards developing applications based on ambiguous conventions, therefore organizations needing solutions in the short term appear unwilling to invest. In addition, IoT introduces a wide range of new security risks and challenges to the IoT devices, the underlying platforms and operating systems where traditional security solutions do not directly apply. This extended attack surface is also considered sinister for all the systems an IoT node is connected to and the possibility of an ominous action that could compromise the entire infrastructure through the aforementioned node is more than likely. These newly introduced security risks are summarized in the following paragraphs.

A. Protecting resource constrained, unattended devices

The majority of IoT nodes are designed to operate unattended and remain physically scattered throughout their lifespan. This renders them vulnerable to physical attacks, where none of the cyber-security solutions of todays cloud computing infrastructure does apply. Provided that most nodes were deployed for performing a specific task, following certain regulations for energy efficiency (leading to increased battery duration) and computational capacity, it is likely that they lack the resources to sufficiently protect themselves. However, security is of paramount importance for any device, regardless of its expected life duration, it is therefore necessary the existence of a dedicated interface or function that would enable software upgrade or a specific flow that would facilitate interventions in the actual hardware. As stated in [17], the fundamental question that remains unanswered is how is it possible to protect a very large number of resource-constrained

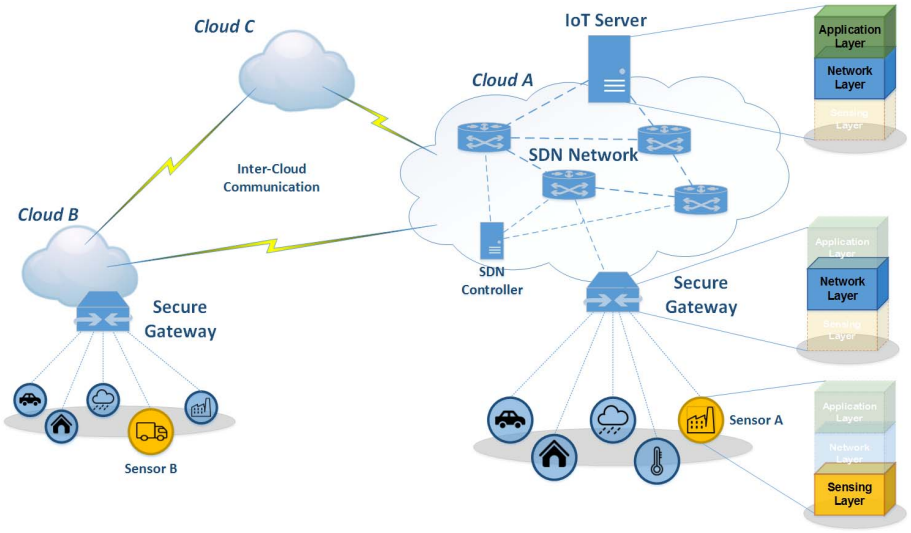


Fig. 2: Secure IoT Inter-cloud communication using Blockchain

devices from security attacks for their entire activity stage. The original notion of requiring each and every device connect to the cloud for security updates is rather impractical due to their highly increased estimated numbers and variety.

B. Trustworthy Security Status Assessment

The IoT paradigm was designed with intended support for large distributed systems. For instance, a smart city will have thousands of deployed devices within its premises, most likely categorized in dedicated networked subsystems for monitoring, data collection and content aggregation. It is therefore essential for any authority to have the ability to tell in a trustworthy manner whether or not such a large number of entities operate properly. Conventional cloud approaches such as authentication mechanisms that enforce nodes to attest their own trustworthiness to a remote verifier through cryptographic methods [18], will most likely have an implicit difficulty meeting the scalability and efficiency requirements simultaneously. In addition, many resource-constrained devices might not be able to support the processing-intensive act of ciphertext generation, or even if they do their sheer numbers will introduce a significant management complexity and prohibitively high cost to the cloud authority or SDN controller that handles such tasks. Last but not least, in real-world scenarios one should be able to ensure that devices that were deployed several years in the past (potentially even by other vendors) are not already compromised.

C. Proper response to security compromises

Current incident response mechanisms are mostly relying on brute-force procedures that cease operations of a potentially compromised system and roll-back its software to a secure state through a reboot process that has a direct functionality effect in every interconnected subsystem. Unfortunately, such highly disruptive responses without any consideration of how fierce the consequences may prove to be, are simply intolerable for mission-critical systems, in which IoT aims to play a vital role. In applications, such as smart cities and grids, intelligent transportation and delivery systems, manufacturing plants, e-Health and vehicle-to-vehicle (V2V) communications, where uninterrupted operation is a primary concern, rebooting is simply not an option. Shutting down an industrial control system delaying the production chain, a critical sensor inside a moving vehicle endangering its passengers or a large power generator, causes more harm than allowing a containerized operations continuity and commence all necessary countermeasures during scheduled maintenance periods.

Addressing IoT security will be a huge step towards widespread adoption of such solutions by large corporations. Filling the technology gaps between current cloud computing and the emerging IoT paradigm will require a new architecture, heavily relying on SDN for handling its complex networking infrastructure, while distributing computing, control and storage functions closer to end users equipment. Such an architecture is presented in Fig. 2, which emphasizes not only to the attributes of each subsystem but also to the inter-cloud communication that must be supported between separate deployments.

Any cloud deployment with native IoT support consists of three main elements (i) the IoT Application Server (or simply IoT Server) where the core IoT application is executed, (ii) the SDN Network and (iii) the newly proposed Secure Gateway node. The later is connected to the sensors that may reside anywhere from topological view, provided that a communication channel is operational on any given time, allowing robust and continuous data transfer. The sensors, the Secure Gateway and the IoT Server may optionally support all three application, networking and sensing layers, however as shown in Fig. 2, certain layers are obligatory for minimum node and subsystem functionality. All communication between the sensors and the IoT Server traverses first through the Secure Gateway and then is redirected to the SDN Network nodes. The Secure

Gateway handles ingress traffic, is therefore responsible for package validity and acts as the *first line of defence* of the entire platform. Message flows that go through the Secure Gateway should be considered benevolent and must treated as such. Communication with neighbouring clouds is handled by the SDN Controller and the regulations that normally apply. The notion of having a Secure Gateway for checking sensor-oriented packages will prove efficient only if such a node can somehow be capable of confirming the authenticity of ingress traffic. In a way, most IoT-related security concerns remain valid but this is where the new Blockchain technology comes into play.

IV. Blockchain

Blockchains have recently gained significant momentum as an emerging method for instantaneous transaction verification among businesses, public or private organizations and industries. However, the potential use of this disruptive technology spawn to each and every application that need to evolve from a centralized authorization entity acting as a trusted intermediary or sometimes a third-party verifiable trust anchor, towards a purely distributed authentication model.

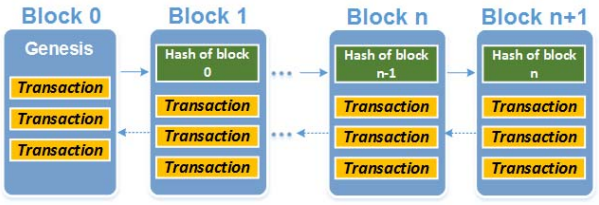
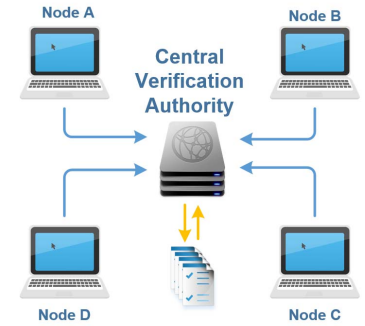


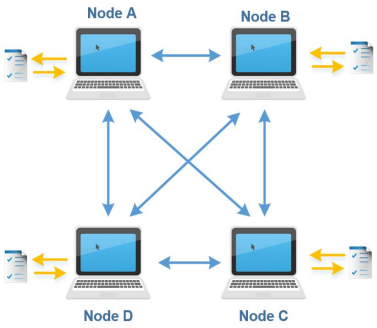
Fig. 3: Blockchain

A blockchain is a tamper-proof, distributed data structure that is replicated and shared among the members of a network. This data structure acts as a log whose elements (or blocks) are batched into timestamped entries, uniquely identified by a specific cryptographic hash. This hash is generated either on the block's content or its header, contains a subset of the overall transactions record made by all interconnected nodes with proper access to the system, and includes a reference to the preceding blocks hash. This method forms a link between blocks that connects them to a form of chain, the **blockchain** [19] as shown in Figure 3. The only slightly different block is the first block of the chain, called Genesis, which is distributed to all clients having access to the blockchain network and can be used as a "key" to the encrypted content of the blockchain. Through this process and upon certain requests towards neighboring nodes with complementary snapshots of the blockchain, each node becomes capable of parsing the entire information set stored in the overall data structure. This provides an accurate impression of the whole network on any given time [19].

In order to fully understand the value of blockchain and its significance to the IoT ecosystem, one must analyze the core difference between centralized and distributed ledgers. In the first approach, shown in Fig. 4(a), the existence of a Central Verification Authority is mandatory for settling transactions, while many intermediaries are involved in validating the integrity of any transaction. The ledger can be changed by any party with access to it and transaction data are prone to attacks since a single security breach may be considered inconsistency of the entire system. On the second approach, the distributed ledger transaction settles automatically while no intermediaries are needed to verify its validity. All transactions are transparent and visible to all parties in real-time, with each block being time- and data-stamped, therefore immutable once recorded in the blockchain.



(a) Centralized Ledger



(b) Distributed Ledger

Fig. 4: Ledger Types

Using blockchain as the main distributed archive of a certain system, allows authorized nodes to instantly track and verify data that is generated by IoT devices, once recorded in this data structure, regardless their volume or the overall number of sources. A distributed, encrypted database is inherently resilient with no single point of failure and since the system is regularly updated with the latest block, accessing any of the active nodes ensures properly updated information. Once created, blockchains are immutable and incorruptible. Mathematically provable algorithms enable continuous verification and calibration of the validity of a blockchain while attempted modifications are immediately flagged. This characteristic nullifies data tampering attacks, since malicious nodes must obtain the Genesis block to decipher any information from the blockchain, a rather impossible task.

Introducing a blockchain-based security layer to the proposed architecture of Fig. 2 now makes much more sense. A

cloud deployment where all sensors and interconnected nodes have access to the Genesis block, thus being able to create immutable blocks that update the existing blockchain, makes authentication tasks conducted by the Secure Gateway straightforward. Benevolent messages could be instantly categorized as part of the affiliated blockchain by simply checking its most recent blocks. No extensive packet inspection is needed and the overall process will be rapidly concluded. Inter-cloud communication will also become much more efficient, since access to a neighbouring cloud deployment will be granted through a copy of the necessary Genesis block. Such level of transparency even allows the IoT Server of Cloud A to obtain information originating from sensor B of Cloud B (Fig. 2), since blockchain enables *trustless* networks, as defined in [19].

V. Conclusion

SDN is without a doubt the networking backbone of cloud deployments around the globe. The original idea of physically separating control and forwarding plane, radically changed networking and rapidly shifted science to a new direction. However, certain security issues are still present and especially when other disruptive platforms and technologies emerge, trying to exploit the advances of the former, an interesting technological melting pot with non-deterministic results appears. IoT is one of those upcoming ecosystems that will have an enormous effect on every aspect of human activity. Its verticals in terms of applicability seem to be directly compatible to the upcoming 5G Networking era [20], [21] and the unprecedented telecommunication merge it proclaims [22].

The work in this position paper acts as the basis for further in-depth investigation of the possibility of utilizing blockchain, a distributed data structure that is used to create a digital transaction ledger and potentially a historical transaction record of massive proportions. This solution will allow encrypted data transfer between interconnected nodes regardless of the network size or its geographical distribution. This mechanism is currently under investigation by industry and academia [23], [24], [25], [26], while other parties [27] evaluate complementary solutions for addressing the thorny issue of IoT security on top of SDN infrastructure deployments.

References

[1] M. Tsagkaropoulos, I. Politis, C. Tselios, T. Dagiuklas, and S. Kotsopoulos, "Service continuity over intertechnology rats," in *2011 IEEE 16th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, June 2011, pp. 117–121.

[2] D. Kreutz, F. M. V. Ramos, P. E. Verssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.

[3] D. Kreutz, F. M. 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*, ser. HotSDN '13. New York, NY, USA: ACM, 2013, pp. 55–60. [Online]. Available: http://doi.acm.org/10.1145/2491185.2491199

[4] D. Athanasopoulos, I. Politis, A. Lykourgiotis, C. Tselios, and T. Dagiuklas, "End-to-end quality aware optimization for multimedia clouds," in *2016 International Conference on Telecommunications and Multimedia (TEMU)*, July 2016, pp. 1–5.

[5] C. Tselios, K. Birkos, P. Galiotos, S. Kotsopoulos, and T. Dagiuklas, "Malicious threats and novel security extensions in p2psip," in *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*, March 2012, pp. 746–751.

[6] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 623–654, Firstquarter 2016.

[7] S. Shin and G. Gu, "Attacking software-defined networks: A first feasibility study," in *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '13. New York, NY, USA: ACM, 2013, pp. 165–166. [Online]. Available: http://doi.acm.org/10.1145/2491185.2491220

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

[9] G. Yao, J. Bi, and L. Guo, "On the cascading failures of multi-controllers in software defined networks," in *2013 21st IEEE International Conference on Network Protocols (ICNP)*, Oct 2013, pp. 1–2.

[10] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008. [Online]. Available: http://doi.acm.org/10.1145/1355734.1355746

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

[12] F. Callegati, W. Cerroni, and M. Ramilli, "Man-in-the-middle attack to the https protocol," *IEEE Security Privacy*, vol. 7, no. 1, pp. 78–81, Jan 2009.

[13] T. Dierks, "The Transport Layer Security (TLS) protocol version 1.2," https://tools.ietf.org/rfc/rfc5246.txt, [Online].

[14] M. Liyanage and A. Gurtov, "Secured vpn models for lte backhaul networks," in *2012 IEEE Vehicular Technology Conference (VTC Fall)*, Sept 2012, pp. 1–5.

[15] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2010.05.010

[16] Gartner Inc., "Gartner Identifies the Top 10 IoT Technologies for 2017 and 2018," http://www.gartner.com/newsroom/id/3221818, [Online].

[17] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, Dec 2016.

[18] A. Francillon, Q. Nguyen, K. B. Rasmussen, and G. Tsudik, "A minimalist approach to remote attestation," in *Proceedings of the Conference on Design, Automation & Test in Europe*, ser. DATE '14. 3001 Leuven, Belgium, Belgium: European Design and Automation Association, 2014, pp. 244:1–244:6. [Online]. Available: http://dl.acm.org/citation.cfm?id=2616606.2616905

[19] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[20] C. Tselios and G. Tsolis, "On QoE-awareness through Virtualized Probes in 5G Networks," in *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2016 IEEE 21st International Workshop on*, 2016, pp. 1–5.

[21] I. Politis, C. Tselios, A. Lykourgiotis, and S. Kotsopoulos, "On optimizing scalable video delivery over media aware mobile clouds," in *IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[22] G. Bianchi, E. Biton, N. Blefari-Melazzi, I. Borges, L. Chiaraviglio, P. Cruz Ramos, P. Eardley, F. Fontes, M. J. McGrath, L. Natarianni *et al.*, "Superfluidity: a flexible functional architecture for 5g networks," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 9, pp. 1178–1186, 2016.

[23] IBM Corp., "Blockchain benefits for electronics - White Paper," https://public.dhe.ibm.com/common/ssi/ecm/gb/en/gbe03809usen/GBE03809USEN.PDF, [Online].

[24] Microsoft Corp., "Blockchain as a Service," https://azure.microsoft.com/en-us/solutions/blockchain/?, [Online].

[25] The Linux Foundation, "Hyperledger project," https://www.hyperledger.org/, [Online].

[26] Ericsson, "Data-centric security," http://cloudpages.ericsson.com/data-centric-security-ebook, [Online].

[27] Citrix Systems Inc., "Netscaler: Secure Event Delivery Controller," [Online].