Journal
Logo

00 (2019) 1–32

Secure Communication Channel Architecture for Software Defined Mobile Networks

Madhusanka Liyanage[a,], An Braeken[b], Anca Delia Jurcut[c], Mika Ylianttila[d], Andrei Gurtov[e]

[a]*Centre for Wireless Communications, University of Oulu, Finland.*
[b]*INDI and ETRO Department, Vrije Universiteit Brussel, Belgium.*
[c]*Department of Computer Science, University College Dublin, Ireland.*
[d]*Centre for Wireless Communications, University of Oulu, Finland.*
[e]*Department of Computer and Information Science, Linkping University, Sweden and ITMO University, Russia.*

Abstract

A Software-Defined Mobile Network (SDMN) architecture is proposed to enhance the performance, flexibility, and scalability of today's telecommunication networks. However, SDMN features such as centralized controlling, network programmability, and virtualization introduce new security challenges to telecommunication networks. In this article, we present security challenges related to SDMN communication channels (i.e., control and data channel) and propose a novel secure communication channel architecture based on Host Identity Protocol (HIP). IPsec tunneling and security gateways are widely utilized in present-day mobile networks to secure backhaul communication channels. However, the utilization of legacy IPsec mechanisms in SDMNs is challenging due to limitations such as distributed control, lack of visibility, and limited scalability. The proposed architecture also utilizes IPsec tunnels to secure the SDMN communication channels by eliminating these limitations. The proposed architecture is implemented in a testbed and we analyzed its security features. The performance penalty of security due to the proposed security mechanisms is measured on both control and data channels.

Keywords:
SDN, NFV, 5G, Telecommunication, Security, Mobile Networks, IPsec, HIP, OpenFlow

1. Introduction

The next generation of mobile networks should support a rich set of network services such as VoIP (Voice over IP), High-Definition (HD) video streaming, gigabit broadband connectivity, mobile cloud services and online gaming. As a result, the mobile traffic usage is drastically growing regardless of the limited radio bandwidth. However, the legacy mobile networks are inflexible, costly and complex to upgrade in order to satisfy this demand [1].

Therefore, mobile network operators adopt innovative technologies to overcome above limitations. On these grounds, Software-Defined Networking (SDN) and Network Function Virtualization (NFV) are identified as promising technologies to solve the existing limitations in legacy mobile networks. The adaptation of SDN concepts is directing the current mobile network towards a flow centric model that employs inexpensive hardware and a centralized controller. Basically, it offers three new features: logically centralized intelligence, programmability and abstraction [1]. Moreover, the adaptation of NFV allows decoupling the network functions from the proprietary hardware appliances, so they can run in software [2]. These features enhance the flexibility, scalability and performance of mobile networks [3].

However, added SDMN features such as centralized controlling, network programmability and network function virtualization introduce new security challenges to mobile networks [4]. Therefore, the security of SDMN is still an open issue and it is a timely research topic to discuss.

The main contributions of this article are 1). A short survey on security issues on SDMN communication channels, 2). Proposal of a new secure communication channel architecture based on Host Identity Protocol (HIP) and IPsec Tunnels, 3). Evaluate the performance of the proposed architecture in a testbed, 4). Evaluate the impact of IP based attacks on SDMN communication channels, 5). Compare the results with legacy security mechanisms.

The scope of this research is limited to address the security aspects of SDMN communication channels only. Thus, the initial survey presents the security challenges, threat vectors and security requirements of SDMN communication channels (i.e control and data channels). IPsec tunnelling and security gateways are widely used to secure backhaul communication channels in legacy mobile networks. According to the recent survey [5], 44% of mobile operators are relying on IPsec tunneling and security gateways mechanisms. Thus, we investigate the usability of these technologies to secure SDMN communication channels and highlight the limitations in legacy IPsec mechanisms. Then, we propose a novel secure communication channel architecture by using HIP not only to overcome the identified limitations but also to provide the required level of security for SDMN communication channels.

The proposed architecture is implemented in a testbed. Its security features are analyzed under different IP attacks: TCP SYN (Synchronization) DoS (Denial of Service), TCP reset and IP spoofing attacks. Moreover, we measure the performance penalty of security due to the proposed security mechanisms on throughput, jitter and latency. Finally, security properties are verified by using formal versification methods.

The rest of the paper is organized as follows. The SDMN architecture and its security issues are explained in Section 2. The proposed architecture is presented in Section 3. We describe our experiment testbed and present the performance analysis results in Section 4. Section 5 contains the security analysis of the proposed architecture. Section 6 and 7 respectively contain the discussion and conclusion of the research article.

2. Background

2.1. Software-Defined Mobile Network (SDMN)

The SDMN architecture is proposed as an extension of the SDN paradigm to incorporate mobile network specific functionalities [1, 2, 3]. The high-level SDMN architecture is illustrated in Figure 1.

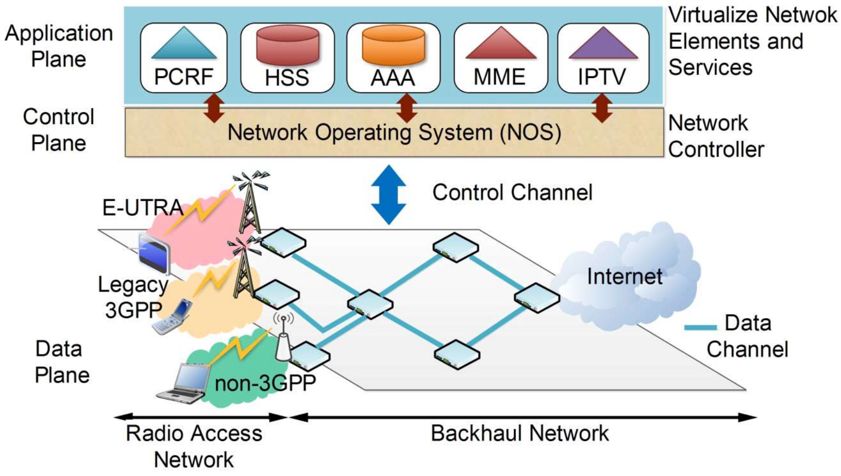


Figure 1: The SDMN Architecture

The SDMN architecture contains three planes, namely 1) Application plane, 2) Control plane and 3) Data plane. Open Application Programmable Interfaces (APIs) are defined to communicate between them.

- Data Plane (DP)

  SDN concepts separate the control plane from the data plane of the network. It pushes the network intelligence to a centralized controller. Thus, the data plane now consists of low-end switches and network links among them. Base stations, wireless access points and the Internet are connected to these DP switches (DPSs). The user traffic is transported through the data plane. This communication channel is called the **data channel**.

- Control Plane (CP)

  The control plane consists of a logically centralized controller which provides the consolidated control functionalities. Basically, the centralized controller supervises the packet forwarding functions of the network through an open interface. Moreover, it controls all the mobile backhaul functionalities such as routing, session initiations, session terminations and billing functions. The communication channel between the controller and DPS is called the **control channel**. This control channel is implemented by using control protocols. For instance, OpenFlow (OF) protocol is the widely used control protocol in the SDN domain [6].

- Application Plane

  The application plane consists of the end-user business applications and other control entities. Legacy mobile network control devices such as Policy and Charging Rules Function (PCRF), Home Subscriber Server (HSS), Mobility Management Entity (MME) and Authentication Authorization and Accounting (AAA) are now software applications which are running on top of the Network Operating System (NOS) [1] at the application plane. The boundary between the application and control layers is traversed by the northbound API.

  The SDMN architecture offers various benefits such as centralized controlling, improved flexibility, efficient segmentation, automatic network management, granular network control, reduced operation cost, low cost backhaul devices, on-demand provision and resource scaling [1]. Thus, SDMN is considered as the latest innovation in the telecommunication domain.

2.2. Communication Security of SDMN

SDMNs are vulnerable to security threats which can be originated at different sections of the network [7]. Therefore, security issues in SDMN backhaul network can be divided into four threat vectors as 1) Application Plane Security, 2) Control Plane Security, 3) Data Plane Security and 4) Communication Security [7][8].

The scope of this article is limited to the communication security aspects only. SDMNs contain two communication channels, i.e. control channel and data channel [1, 2, 3]. Therefore, the communication security threat vector can be further divided as 1) Security issues related to the control channel and 2) Security issues related to the data channel [7].

2.2.1. Security Issues related to the SDMN Control Channel

The main security issue of the control channel is the lack of IP level security. Existing SDMN control protocols rely on higher layer secure mechanism such as TLS (Transport Layer Security)/ SSL (Secure Sockets Layer) sessions. For instance, the widely used OF protocol utilizes TLS/SSL based control channels [8]. However, higher layer secure mechanisms are vulnerable to IP based attacks such as IP spoofing, TCP SYN DoS and TCP reset attacks [7] [9]. Therefore, the higher layer protection mechanisms are not sufficient enough to provide the required level of robustness and security for the control channel [8].

Moreover, a strong authentication mechanism is required between the controller and DPSs. Otherwise, intruders can impersonate as legitimate DPS and launch security attacks on the control channel. For instance, the attacker can inject fake flow requests to perform DoS attacks [7]. However, TLS/SSL sessions do not utilize a strong authentication procedure between controllers and switches. For example, the authentication mechanism of TLS/SSL sessions is vulnerable to IP spoofing and Compression Ratio Info-leak Made Easy (CRIME) attacks [9]. Table 1 contains known attacks on OF control channels.

The network controller is the key component of the SDMN network due to its centralized intelligence and controlling abilities. As a result, attacks on the network controller represent the most severe threats to the SDMN architecture. The control channel is the only interface which enables the communication between DPSs and the controller. Therefore, the security of the control channel is a key factor to ensure the proper communication with the controller [7]. A DoS attack on the SDN controller is demonstrated in [10] in which an attacker continuously sends IP packets with random headers to the controller via the control channel. This puts the controller in a non-responsive state and it will be unable to deploy flow rules in the DPS.

TLSv1 based communication is optional in the latest OpenFlow specifications due to its complexity of configuration [11]. TLS configuration requires to generate network site-specific certificates and corresponding signed device certificates with site-wide private keys for the controller and DPSs [12]. Therefore, many SDN equipment vendors have skipped the support for TLS in their DPS. It leaves the control channel vulnerable to security attacks. Thus, the control channel needs to be secured by using other mechanisms.

Table 1: Known attacks on OF control channel

Attack type	**Trigger and description**	**Impact on SDMN control channel**
TCP SYN (Synchronization) DDoS	A set of attackers sends a succession of TCP SYN requests to consume enough server resources in order to make the controller and/or DPSs unresponsive to legitimate traffic.	Ternary Content-Addressable Memory (TCAM) of DPS will overflow.
TCP reset attack	The attacker inserts a sequence of TCP reset requests to prematurely reset the communication session.	Unexpected termination and service quality decrement of communication channels.
RC4 biases in TLS	The attacker can recover the full plaintext when it is repeatedly encrypted in the same or several different sessions.	Exact information to perform future attacks and reveal the identity of the backhaul devices.
Browser Exploit Against SSL/TLS (BEAST) attack	The attacker mounts an adaptive chosen plaintext attack with predictable initialization vectors (IVs) using cipher block chaining (CBC).	Exact information to perform future attacks and reveal the identity of the backhaul devices.
Compression Ratio Info-leak Made Easy (CRIME) attack	The attacker discovers session tokens and other secret information to perform session hijacking on an authenticated communication session.	Exhaust controller resources by adding/modifying fake flow requests. Include fake flow rules to exhaust TCAM of OF switches. Jeopardize the data plane by destroying the in-flight flow rules.
LUCKY 13	The attacker performs a man-in-the-middle attack to recover the plaintext from a CBC (Ciper-block chaining) encrypted TLS session.	Exact information to perform future attacks and reveal the identity of the backhaul devices.
POODLE attack	The attacker forces to change TLS sessions to SSL3.0 sessions and uses design flaws in SSL 3.0 that allows changing padding data at the end of a block cipher. As a result, the encryption cipher becomes less secure each time it is passed.	Termination of the communication between the controller and DPS.

2.2.2. *Security Issues related to the Data Channel*

The SDMN architecture has an all-IP based backhaul network. In contrast to the prior 2G/3G telecommunication networks, the Radio Network Layer (RNL) encryptions are terminated at eNodeBs in the latest IP based telecommunication networks including SDMN [13]. Thus, current SDMN backhaul traffic is unencrypted and attackers can perform “SDN Scanner” mechanisms to collect network information [14]. Later, this information can be used to perform IP based attacks such as DoS, reset, replay and spoofing attacks [15].

Furthermore, current SDMN data channel does not contain any integrity protection mechanism. Thus, a flow modification attacker can alter or destroy the data without being noticed by the network operator. The alternations of data flows may result to decrease the Quality of Service (QoS) of communication sessions [7].

The SDMN architecture requires strong mutual authentication mechanisms for the data channel as well. Without such authentication mechanisms, intruders can impersonate as legitimate switches and inject forged traffic flows to the data plane [15]. Such a way, the attacker can exhaust the flow tables of DPS and reduce the available bandwidth for user traffic [8]. Moreover, it will also affect the control plane by inducing unnecessary flow requests to the controller [15].

2.3. *Impact of IP based attacks on Communication Channels*

Although IP based attacks are common in other IP networks such as Internet, campus networks and data center networks, most of these attacks are new to telecommunication networks. Here, we discuss how these common IP based attacks are now mounted and effect on SDMN communication channels.

2.3.1. *DoS/DDoS attacks*

There are different ways to perform IP based DoS Attacks. The most common IP based DoS attack is that an attacker sends an extensive amount of connection establishment requests (e.g. TCP SYN requests) to establish hanging connections with the controller or a DPS. Such a way, the attacker can consume the network resources which should be available for legitimate users [7]. In other cases, the attacker inserts a large amount of fake packets to the data plane by spoofing all or part of the header fields with random values [14]. These incoming packets will trigger table-misses and send lots of packet-in flow request messages to the network controller to saturate the controller resources. In some cases, an attacker who gains access to DPS can artificially generate lots of random packet-in flow request messages to saturate the control channel and the controller resources [15]. Moreover, the lack of diversity among DPSs fuels the fast propagation of such attacks [7].

Legacy mobile backhaul devices are inherently protected against the propagation of attacks due to complex and vendor specific equipment [16]. Moreover, legacy backhaul devices do not require frequent communication with core control devices in a manner similar to DPSs communicating with the centralized controller. These features minimize both the impact and propagation of DoS attacks. Moreover, the legacy backhaul devices are controlled as a joint effort of multiple network elements [13]. For instance, a single Long Term Evolution (LTE) eNodeB is connected up to 32 MMEs [13]. Therefore, DoS/DDoS attack on a single core element will not terminate the entire operation of a backhaul device or the network.

2.3.2. *Reset attacks*

The attacker inserts fake packets to the control or data channel by requesting to reset communication sessions. For instance, an attacker can set the reset bit in the TCP header to terminate a TCP session. Such incoming packets will reset the communication session between backhaul devices. These attacks are feasible in an SDMN architecture due to the unprotected TCP based control and data channels. For instance, OF uses TCP sessions [6] and about 75% of mobile data traffic are transported via TCP sessions [17]. On the other hand, legacy mobile networks utilized dedicated, complex and less ubiquitous control protocols than widely utilized SSL/TLS or TCP sessions. Thus, attackers need special knowledge on these complex control protocols to deploy any reset attack [13].

2.3.3. *IP spoofing attacks*

IP based SDMN communication channels are vulnerable to IP spoofing attacks without proper security features. An attacker impersonates as a legitimate device and performs IP based attacks such as DoS, reset and message modification attacks on both control and data channels [4].Prior to 2G/3G, mobile networks had non-IP backhaul networks.

Therefore, backhaul communication channels are inherently secure against IP spoofing attacks [16]. Moreover, IP based LTE backhaul network traffic is tunneled over GTP (GPRS Tunnelling Protocol). Therefore, it is not possible for outside devices to perform IP spoofing attacks on LTE backhaul [16]. However, the conversion of telecommunication network to Internet Service Provider (ISP)-style open architecture in SDMN will increase the vulnerability to IP spoofing attacks.

2.3.4. Eavesdropping attacks

The SDMN control channel relies on upper layer encryption mechanisms such as SSL/TLS. It leaves TCP/IP level network information such as IP addresses, ports, sequence numbers unencrypted [9]. Furthermore, current SDMN data channel traffic is unencrypted. Therefore, both control and data channels are vulnerable to eavesdropping attacks at different levels. Attackers can apply widely available IP sniffing tools to eavesdrop the network information, which is used to perform IP based attacks such as DDoS, reset and spoofing attacks. For instance, attackers can perform IP/TCP port scans on mobile backhaul elements to identify the active ports and exploit their vulnerabilities [15].

However, prior mobile networks utilized lower layer encryption mechanism (e.g. RNL encryption in 2G/3G) to encrypt both control and user traffic [16]. Moreover, LTE backhaul traffic is tunneled over GTP tunnels with IPsec encryption and it restricts eavesdropping opportunities [18].

2.3.5. Message modification attacks

Without a proper integrity protection mechanism, SDMN communication channels are vulnerable to data modification attacks. For instance, the SDMN control channel is lacking secure integrity protection mechanisms to protect the TCP/IP header information. Therefore, an attacker can overwhelm the controller resources by modifying flow requests in an undetectable manner [7]. Moreover, an attacker can randomly spoof the data channel packet header information to trigger table-misses in DPS. It will send a lot of packet-in flow request messages to the network controller [15].

Prior mobile backhaul networks used encryption and integrity protection mechanisms (e.g. RNL, IPsec encryption [18]) to protect the integrity of both control and user data.

To conclude the security discussion, Table 2 contains the widely available IP based attacks which are affecting the performance of SDMN communication channels.

Table 2: Impact of different IP based attacks on SDMN Communication Channel

Security Attack	**Affecting Channel**	**Impact on SDMN Communication Channel**	**Existing Mitigation Mechanism**	**Known Vulnerabilities of Existing Mitigation Mechanism**
DoS/DDoS Attacks	Control	Reduce/terminate the availability of the controller [7]	Ingress Filtering	Prevent the resource consuming DoS attacks. However, fail to prevent intelligent DoS attacks such as TCP SYN DoS attacks.
	Data	Reduce/terminate the availability of DPS and elements		
Reset Attacks	Control	Terminate the ongoing communication sessions with the controller		
	Data	Terminate the ongoing communication sessions between DPSs		
Spoofing and Impersonation Attacks	Control	Impersonate as a legitimate DPS and perform security attacks such as DoS, flow modification and eavesdropping attacks. Insert fake flow requests and overwhelm the controller resources.	Server-client certification based authentication	Vulnerable to IP spoofing and Compression Ratio Info-leak Made Easy (CRIME) attacks [9].
	Data	Destroy or modify the user traffic. Divert the traffic flows to wrong destinations [15]	Ingress Filtering and Network Address Translation (NAT)	Prevent Random spoofing attacks for some extend. Still vulnerable to subnet spoofing attacks.
Eavesdropping Attacks	Control	Steal IP level network parameters of the controller to perform IP based attacks such as DoS, reset and spoofing attacks [14]	Encrypt the data in the application layer	TCP/IP level network information such as IP addresses, ports, sequence numbers are still unencrypted.
	Data	Steal IP level network parameters and flow information to perform IP based attacks such as DoS, reset, spoofing and flow modification attacks [14] [15]		
Message Modification Attacks	Control	Overwhelm the controller resources by modifying flow requests [7]	Message Integrity check using a keyed MAC (Massage Authentication Code)	Secure the integrity of payload. However, network level information such as IP addresses, port numbers are still vulnerable.
	Data	Exhaust memory in DPS for flow tables with fake flow entries and reduces QoS of user services.		
Replay Attacks	Control	Replay the authentication requests to establish connection with the controller.	Protected against replay attacks using the MAC secret and the sequence number in SSL/TLS sessions.	
	Data	Replay connection establishment massages to establish connection with DPSs		

Consequently, it is clear that both control and data channels are vulnerable to IP based attacks and new security mechanisms are required to prevent such attacks.

2.4. Principles of Adequate Security for Communication Channels of SDMNs

SDMNs offer many advantages due to its centralized control, network programmability and network abstraction [1][3]. Security mechanisms for SDMN communication channels should not suppress these features. Therefore, SDMN communication channels require a security model which only covers the common Confidentiality, Integrity and Availability (CIA) features, but also extended with new security requirements such as centralized policy management and visibility (Figure 2).

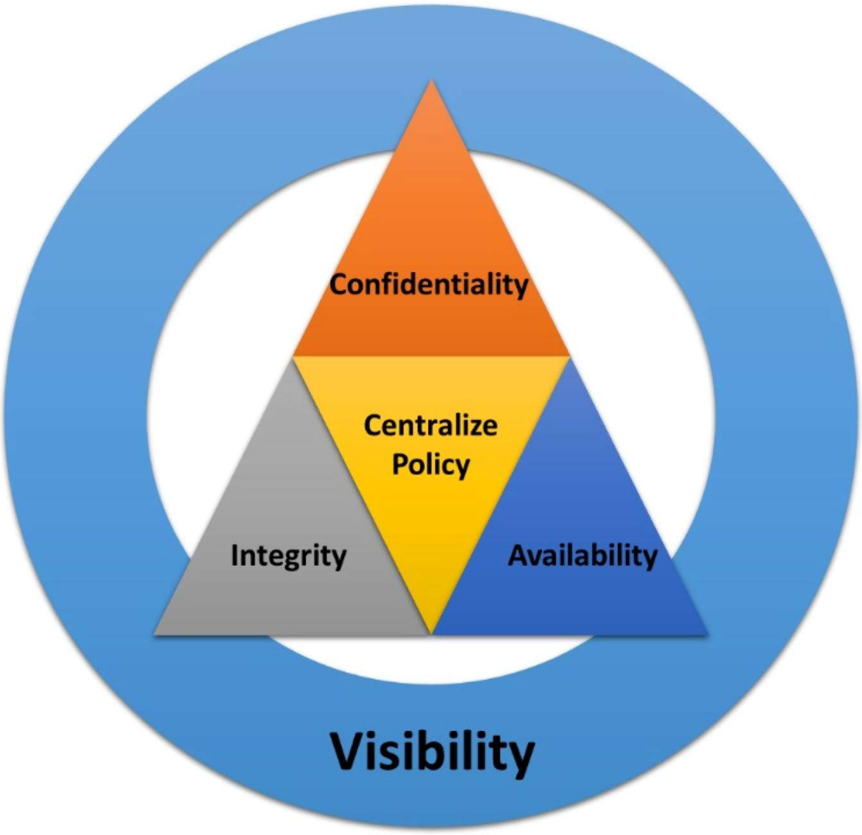


Figure 2: Security model for SDMN Communication Channels

2.4.1. Confidentiality

Confidentiality ensures that only the authorized backhaul devices can communicate with other backhaul elements by restricting unauthorized access to the communication channels. Key steps to ensure the confidentiality are the authentication, access control and encryption. Network elements need to be authenticated before initiating the communication. Authentication requires to proper identification of a user to provide necessary access to the system (Access control). Then, users encrypt the communication data and thus prevent unauthorized access to the data. The latest LTE and its predecessors (2G and 3G mobile networks) always used authentication, access control and encryption mechanisms to ensure confidentiality [13]. In SDMN networks, both control and data channel traffic should provide the same level of confidentiality as prior networks. The controller should authenticate DPSs and DPSs should be mutually authenticated before the communication session establishments. Moreover, both control and data traffic should be encrypted.

2.4.2. Integrity

Integrity ensures that the communication data is not tampered or removed in unauthorized or undetected manners during transfer between the different network elements. Usually, encryption methods are used to ensure the integrity of legacy telecommunication networks. RNL encryption is used in 2G/3G networks and IPsec encryption is used in LTE backhaul networks [13]. SDMNs also need to ensure the integrity of both control and data channels to avoid unauthorized modifications.

2.4.3. Availability

Availability ensures that communication channels are available when they are required. High availability is a key differentiation factor of telecommunication networks from other communication networks. Mobile networks offer Service Level Agreements (SLAs) with 99.999% availability [19]. Although legacy mobile networks do not have a specific mechanism to provide the availability, these networks are strongly safeguarded and pro-actively monitored from end-to-end manner [18]. Proposed centralized controller based SDMN architecture requires to maintain the high availability of SDMN controller for the smooth operation of the entire network. Therefore, the availability of the control channel should be the same as the controller since it is the only interface for DPSs to access the controller. On the other hand, data channel availability should be further improved since future mobile networks should offer higher availability than the current networks.

2.4.4. Centralized Policy Management

SDMN architecture utilizes a logically centralized controller to operate the entire network. Thus, the centralized policy management is one of the key advantages of SDMN which enables better control of network resources, services and applications than legacy networks. Therefore, the SDMN communication security mechanisms should be compatible with the centralized policy management. However, present mobile networks (e.g. 2G, 3G and LTE) use distributed security management mechanisms [18]. Most of the security procedures are designed to protect the network perimeter while leaving the inside network unprotected [13]. The lack of coordination between these security mechanisms leads to complex security systems with overlapping and redundant security services. It ultimately reduces the overall network performance.

2.4.5. Visibility

The visibility of the network helps to optimize the network capacity, detects the anomalous behaviors of the network traffic, ensures higher QoS and provides high availability. The SDMN architecture dramatically improves the visibility of network activities and provides global visibility across the mobile backhaul. Therefore, the security mechanisms for SDMN communication channels should not reduce the enhanced visibility of the network. However, present cellular networks are suffering from the lack visibility due to the complex networking protocols and the lack of orchestration between the network devices.

2.5. Reason to Fail Legacy IPsec Tunneling Mechanisms in SDMNs

IPsec is the most commonly used security protocol to secure the IP based communication sessions in a network. The latest IP based telecommunication networks (i.e. LTE/LTE-A) also use IPsec based mobile backhaul communication channels [5][13][18]. Several IPsec key exchange mechanisms were proposed to set up Security Associations (SAs) for IPsec tunnels, namely, Internet Key Exchange version 1 (IKEv1) [20], Internet Key Exchange version 2 (IKEv2) [21], IKEv2 Mobility and Multihoming Protocol (MOBIKE)[22], Host Identity Protocol (HIP) [23]. However, it is not possible to use legacy IPsec tunneling and key exchange mechanisms in SDMNs due to following limitations.

- Distributed tunnel establishment and lack of centralized controlling.

  Existing IPsec mechanisms establish end-to-end tunnels in distributed manner. They do not support the centralized policy management or centralized coordination during tunnel establishment.

- Point-to-Point tunnel establishment.

Existing IPsec mechanisms support point-to-point tunnels only. However, SDMN DPSs require multipoint-to-multipoint tunnels to enable network features such as load balancing, best path routing and automatic redundancy functions [1]. Moreover, point-to-multipoint tunnels are required to establish multiple control channel sessions with multiple controllers in multi-controller environment [24].

- Per tunnel encryption key negotiation.

  Legacy IPsec key exchange mechanisms negotiate a unique encryption key per tunnel. However, DPSs require to forward the same traffic to multiple nodes via multiple tunnels. For instance, a DPS might need to send the same encrypted control message to multiple controllers in a multi-controller architecture [24] or need to send the same user traffic to multiple switches in multicast and broadcast events. In these instances, the same data has to be encrypted several times with different encryption keys. Multiple encryptions of the same data with different keys waste the processing resources of DPSs and increase the latency.

- Limited security plane scalability.

  Per tunnel encryption key negotiation reduces the scalability of security mechanism as well. DPS needs hundreds of tunnel establishments to communicate with hundreds of other switches. Thus, DPSs need to store hundreds of keys and security agreements. Moreover, they need to use complex key management mechanisms. These attributes require high processing power and memory which ultimately increase the cost and the complexity of DPSs.

- Lack of visibility.

  In present IPsec mechanisms, the tunnel establishment and key negotiations are invisible to other network elements except the end devices [25]. Moreover, the encrypted IPsec traffic is visible only for the end devices. Thus, it reduces the visibility of end-to-end traffic transportation.

- Lack of access control.

  Access control plays a major role to ensure the confidentiality of the network. Legacy IPsec tunnel mechanisms support only mutual authentication but not the access control. Thus, they fail to ensure the required confidentiality of the network.

- Static tunnel establishment.

  Legacy IPsec mechanisms support only static tunnel establishments. During the tunnel establishment procedure, both end devices are agreed on static network attributes such as tunnel duration, encryption keys and traffic separators. It is not possible to change these parameters according to the traffic demands and time of the day. Thus, static tunnel establishments over-utilize the network resources such as bandwidth, processing and memory [26].

2.6. Host Identity Protocol (HIP)

HIP is a novel mobility and security management protocol which is standardized by IETF (Internet Engineering Task Force) [27][23]. HIP separates the dual role of IP address as the locater and the host identity. Each HIP host has a public/private key pair and the public key is used as its Host Identity (HI). HIP utilizes a base protocol named HIP Base Exchange (HIP BEX) to mutually authenticate the end nodes and establish a Security Association (SA) for IPsec tunnels.

2.7. Related Work

Many recent research articles proposed security mechanisms to solve the related security threats of general SDNs [28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39]. In [28], authors proposed an OpenFlow security application development framework designed to facilitate the rapid design, and modular composition of OF-enabled detection and mitigation modules. In [40], an economical deployment of security monitoring systems called OpenSafe is proposed to manage the routing of traffic through network monitoring devices . In [29], authors designed a security layer between a software-defined networking controller and network devices that checks for network-wide invariant violations dynamically as each forwarding rule is inserted, modified or deleted. A tool to identify any intra-switch misconfiguration within a single FlowTable is presented in [30]. A lightweight method for DDoS attack detection based on traffic flow features is presented in [32]. A use case of SDN based anomaly detection system for improved detection in small scale networks was demonstrated in [41]. A deployment of security middle-boxes without requiring modifications in middle-boxes or the SDN architecture was presented in [42]. Table 3 contains the proposed security mechanism for general SDN networks.

Table 3 categorized the above described security mechanisms based on the ability to secure the Control Plane (CP), Data Plane (DP), Control Channel (CC) and Data Channel (DC).

Table 3: Proposed Security Mechanism for General SDNs

Security Type	**SDN Plane / Communication Channel**	**Reference**
Threat detection and mitigation	CP	[28][41]
Flow rules verification, Configuration verification	CP, DP	[29][30]
Conflict resolution, authorization, security audit system	CP, DP	[31][40]
DDoS detection, Controller resilience	CP, DP	[32][33]
Link monitoring	DP, CC	[34]
Find contradictions in flow rules, authorize applications	CP, DP	[35]
Controller availability, network monitoring	CP, CC	[36][37]
Access control and dynamic policy enforcement	CP, DP	[38][39]

Most of the security solutions listed above are proposed to secure the control and data planes, rather than the communication channels. Only a very few security mechanisms [34][37] contain security features that can be used to secure the SDN control channel. In [34], authors proposed a fast channel recovery mechanism based on link monitoring. Split architecture deployment for control plane and channel is proposed in [37] to increase the reliability and resilience. However, none of the security mechanisms have directly addressed the security issues related to SDMN communication channels.

OF specifications proposed to used SSL/TLSv1 sessions to protect control channel communication [6]. Above security proposals also use SSL/TLSv1 sessions to protect control channel communication [34][36][37]. However, SSL/TLSv1 based communication is not sufficient to protect control channel from many attacks [9]. On the other hand, none of the data channel security mechanisms are proposed for SDMN architecture yet. Therefore, our architecture fulfills the missing security features of the SDMN communication channels.

3. Proposed Secure Communication Channel Architecture

We propose a novel IPsec based SDMN backhaul traffic architecture to secure the communication channels. It is a “bump-in-the-wire” security architecture based on HIP. The proposed architecture is presented in Figure 3 [43].

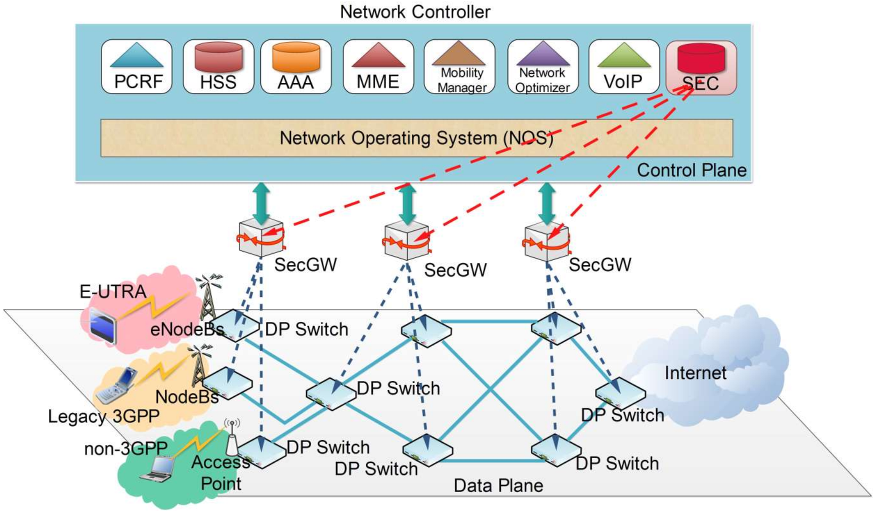


Figure 3: Proposed Secure Communication Channel Architecture

Our architecture proposes five main changes to the existing SDMN architecture. First, distributed Security Gateways (SecGWs) are utilized to secure the controller from the outside network. Second, a new Security Entity (SecE) is added as a control entity to control the SecGWs and other security functions. Third, a Local Security Agent (LSA) is installed in each DPS to handle security related functions in the switch. Fourth, IPsec Encapsulating Security Payload (ESP) Bounded-End-to-End-Tunnel (BEET) [44] mode tunnels are used to secure the control and data channels communication. Fifth, session based Traffic Encryption Keys (TEKs) are used to encrypt the control and data channel traffic. Note that the introduction of the three entities, SecGWs, SecE, and LSA, offers a modular and easy plug-in solution for the establishment of the security features in the current infrastructure.

We propose to place all the network control functionalities in a centralized location, enabling the creation of a trusted network zone for the important network elements in the SDMN. Consequently, a single administrative authority becomes responsible for the control of physical site locations, the ownership, and the operation of the network. As a result, the SecGWs are now the only interface between this trusted network zone and the outside world. All message exchanges between SecE and the SecGWs are trusted and secure, since it happens in the trusted zone.

Note that in prior 2G/3G telecommunication systems, cellular networks were considered as trusted networks. However, an LTE network is not a trusted network since all the sections of an LTE transport network are not physically secured as in the previous 2G/3G networks. Let us now discuss the three new entities into more detail.

- Security Gateway (SecGW): SecGW is the intermediate device between the controller and the data plane. The SecGWs hide the network controller from the outside world and reduce the security related work load of the controller. SecGWs are responsible for two functions. 1) Establish IPsec tunnels with DPSs, 2) Relay the messages between SecE and DPSs.

  Here, we propose to utilize distributed/multiple SecGWs to prevent a single point of failure. Moreover, it is possible to integrate various security functions such as Intrusion Detection Systems (IDS), Deep Packet Inspection (DPI) and Firewalls within SecGWs to provide extra protection.

- Security Entity (SecE) SecE is a new control entity which controls the SecGWs and other security functions. It authorizes DPSs (Figure 5 and 7) based on Access Control Lists (ACLs). The network operator uploads a set of ACLs which contain the identities of legitimate DPSs. SecE also generates Traffic Encryption Keys (TEKs) for both control and data channels. Furthermore, SecE cooperates with other control entities (e.g. Traffic Optimizer Entity (TOE)) to manage the tunnel establishments in the control and data channels.

- Local SEC Agent (LSA) LSA is a security entity which is implemented in each DPS. Figure 4 illustrates the position of LSA in a DPS. It is mainly responsible for HIP tunnel establishments with SecGWs and other DP switches.

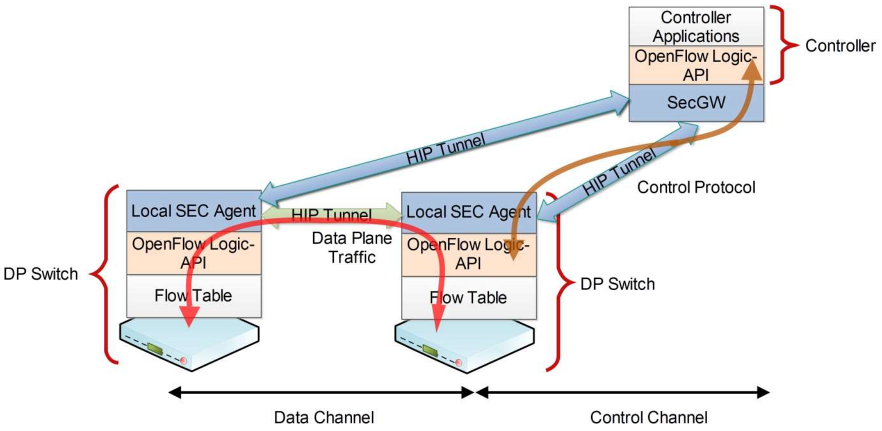


Figure 4: The Secure Control and Data Channel

3.1. Key Management

The proposed architecture uses three types of TEKs.

- Control Traffic Encryption Key (CTEK) CTEK is used to encrypt the control channel traffic. SecE periodically generates CTEKs and distributes them to the DPSs. CTEKs are encrypted by using the KEK (Key Encryption Key) of the DPS and delivered via the SecGWs.
- Data Traffic Encryption Key (DTEK) DTEK is used to encrypt the data channel traffic. SecE periodically generates DTEKs and distributes them to the DPSs. DTEKs are also encrypted by using the KEK of the DPS and delivered via the SecGWs.
- Key Encryption Key (KEK) The KEK is used to encrypt CTEKs and DTEKs during the delivery via the secure control channel. The KEK is unique to each DPS. SecGW and each DPS agreed on this KEK during the tunnel establishment procedure by using Diffie-Hellman (D-H) key exchange protocol. KEKs are periodically update by using D-H key exchange protocol.

3.2. Control Channel

We propose a HIP based control channel. A HIP (IPsec ESP BEET mode [44][45]) tunnel is established between SecGW and LSA (Figure 4). Thus, the controller and DPSs can communicate by using the traditional control protocol (e.g. OF protocol) without any modification. The proposed security mechanism is invisible to the existing control protocol.

3.2.1. Authentication and Registration Procedure of DPSs

The proposed architecture supports the dynamic addition of new DPSs and the automatic control channel establishment. Here, we propose a novel tunnel establishment procedure based on HIP BEX [46]. It supports two tasks. 1) Authenticate and register a new DPS. 2) Establish an IPsec (ESP BEET mode) tunnel between the DPS and SecGW.

In the proposed architecture, every DPS has its own public/private key pair and the public key is used as its host identity (HI). The public/private key pair is stored in each DPS before the installation in the network. At the same time, the network operator adds the HIs of the legitimate switches to the ACLs. In order to be able to register to the network, the DPS should be aware of the HI of the SecGW to which it wants to register.

The proposed tunnel establishment procedure is presented in Figure 5.

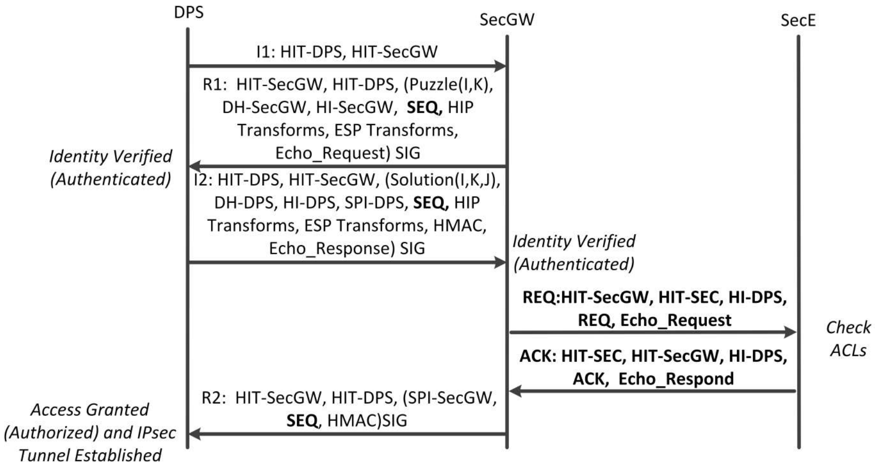

DPS
SecGW
SecE
I1: HIT-DPS, HIT-SecGW
R1: HIT-SecGW, HIT-DPS, (Puzzle(I,K), DH-SecGW, HI-SecGW, SEQ, HIP Transforms, ESP Transforms, Echo_Request) SIG
Identity Verified (Authenticated)
I2: HIT-DPS, HIT-SecGW, (Solution(I,K,J), DH-DPS, HI-DPS, SPI-DPS, SEQ, HIP Transforms, ESP Transforms, HMAC, Echo_Response) SIG
Identity Verified (Authenticated)
REQ:HIT-SecGW, HIT-SEC, HI-DPS, REQ, Echo_Request
Check ACLs
ACK: HIT-SEC, HIT-SecGW, HI-DPS, ACK, Echo_Respond
R2: HIT-SecGW, HIT-DPS, (SPI-SecGW, SEQ, HMAC)SIG
Access Granted (Authorized) and IPsec Tunnel Established

Figure 5: The Tunnel Establishment Procedure for Control Channel

The DPS initiates the tunnel establishment procedure by sending an I1 message. It contains the HITs (Host Identity Tags) of DPS and SecGW. Note that the HIT corresponds with the 128-bit hash of the HI. To prevent DoS attacks, the SecGW replies the I1 message with a pre-computed R1 message without allocating any resources. The main components of the R1 message are the cryptographic puzzle, D-H key parameters, the public key of SecGW, ESP transforms, HIP transforms, the echo request and the signature. D-H key parameters are exchanged between two nodes to generate a common symmetric key which is used as the KEK of DPS. D-H key parameters are used to generate a symmetric key for ESP payload encryption. The set of IPsec encryption and hashing algorithms supported by SecGW is included as ESP transform parameters. The set of encryption and integrity algorithms supported by SecGW is contained in HIP transforms section. These HIP transforms are used to protect the HI exchange. The echo request parameter contains an opaque blob of data which should be echoed back in the reply packet. It is used to check the integrity of the puzzle. Finally, a signature is generated over the R1 message by using the SecGW's private key. It verifies the integrity of the R1 message. The sequence number contains the monotonically increasing “R1 generation counter” value which is used to protect the initiator from R1 messages based replay attacks.

DPS sends the I2 message after the arrival of the R1 message. The I2 message contains HMAC (Hash Message Authentication Code), the solution of the cryptographic puzzle, D-H key parameters, the public key of the DPS, SPIs, ESP transforms, HIP transforms and the signature. I2 has similar obligatory fields as R1, except the puzzle parameter contains the solution, HMAC and SPIs. HMAC is used to the faster verification of I2 than HIP signature check to avoid replay attacks. Thus, HMAC is checked by SecGW before the signature. Then, SecGW verifies the solution of the puzzle. The puzzle verification is a single fast hash computation. In a HIP (IPsec BEET mode) ESP packet, HIs are not transported and SPIs are used to locate a correct SA. Thus, the selected SPI value for the established SA is included in SPI parameter field.

Then, the SecGW sends the switch's credentials to SecE via REQ message. It contains the HIs of DPS and SecGW, the authentication request, and the echo request. Upon the arrival of REQ, SecE checks the received HIs against the ACLs and the network optimizer. Then, it replies the REQ message with an ACK message. The ACK message contains two HIs, the acknowledgment and the echo reply. Figure 6 illustrates the different steps in the protocol.

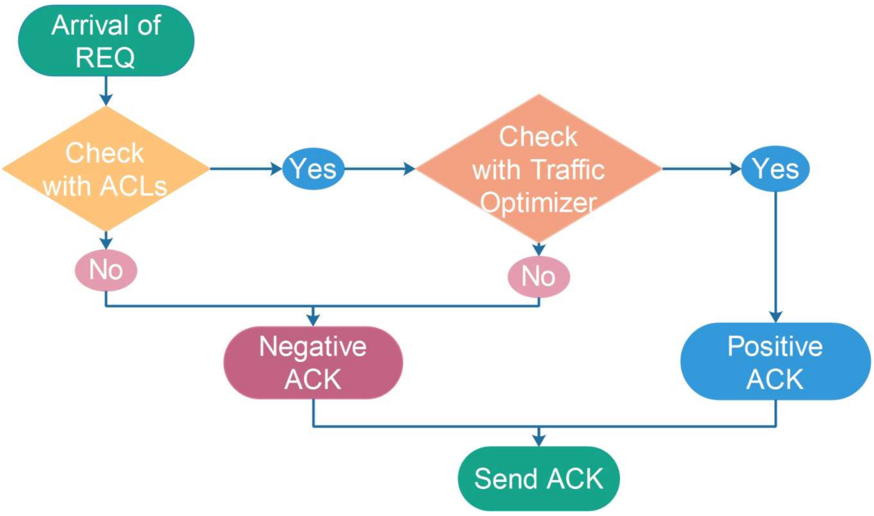


Figure 6: The algorithm to decide the content of acknowledgment

Here, the HI of the DPS is checked with the ACLs by SecE. This step prevents unauthorized access to the network. Moreover, SecE keeps a record of the different requests with a time stamp. It helps to identify replay attacks. If a DPS sends premature requests again and again, those requests will be dropped.

Then, the TOE checks the HIs of DPS and SecGW with the traffic optimization procedure. In our proposal, we illustrate the cooperation of TOE and SecE only. However, SecE can communicate with other control entities (e.g. Mobility manager, Topology manager, Load balancer) before generating the acknowledgment.

A positive acknowledgment is sent for a request from an authorized DPS and a negative acknowledgment is sent for other requests. In case of a negative acknowledgment, SecGW drops the connection request from the DPS. Otherwise, SecGW completes the tunnel establishment by sending the R2 message. It contains the HIs of DPS and SecGW, SPIs, HMAC and the signature.

It is possible for DPSs to establish HIP tunnels with multiple SecGWs to obtain the load balancing and redundancy features. In such cases, the DPS has to follow the above tunnel establishment procedure with each SecGW. SecE keeps track of the connected SecGWs for each DPS. Therefore, it can accept or reject the incoming connection requests to distribute the load equally on each SecGW. Moreover, SecE can stop the aggressive DPSs, who try to connect with too many SecGWs.

3.3. Data Channel

Similar to the control channel, we propose a HIP based data channel. HIP tunnels are established between LSAs to secure the data channel (Figure 4). The proposed security mechanism is invisible to the DP traffic. Thus, the traditional DP traffic is transported between DPSs without any modification.

Our architecture proposes three changes to the existing SDMN data channel. First, DPSs are mutually authenticated based on a PKI mechanism before any data exchange. Second, the communication session establishment between DPSs is authorized based on ACLs and other control entities (e.g. TOE). Third, IPsec (ESP BEET mode) tunnels are established between switches to secure the data channel communication.

3.3.1. Tunnel establishment procedure of data channel

We propose a novel tunnel establishment procedure for the data channel as well. The mechanism is presented in Figure 7.

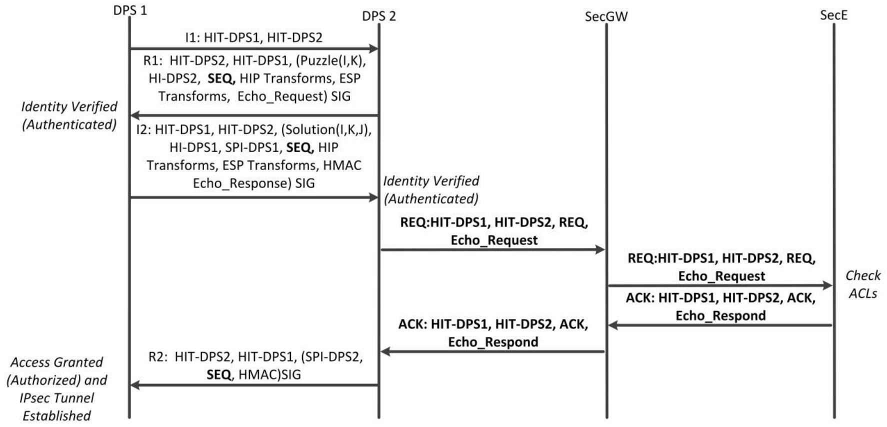

DPS 1
DPS 2
SecGW
SecE
I1: HIT-DPS1, HIT-DPS2
R1: HIT-DPS2, HIT-DPS1, (Puzzle(I,K), HI-DPS2, SEQ, HIP Transforms, ESP Transforms, Echo_Request) SIG
Identity Verified (Authenticated)
I2: HIT-DPS1, HIT-DPS2, (Solution(I,K,J), HI-DPS1, SPI-DPS1, SEQ, HIP Transforms, ESP Transforms, HMAC Echo_Response) SIG
Identity Verified (Authenticated)
REQ:HIT-DPS1, HIT-DPS2, REQ, Echo_Request
REQ:HIT-DPS1, HIT-DPS2, REQ, Echo_Request
Check ACLs
ACK: HIT-DPS1, HIT-DPS2, ACK, Echo_Respond
ACK: HIT-DPS1, HIT-DPS2, ACK, Echo_Respond
R2: HIT-DPS2, HIT-DPS1, (SPI-DPS2, SEQ, HMAC)SIG
Access Granted (Authorized) and IPsec Tunnel Established

Figure 7: The Tunnel Establishment Procedure for Data Channel

The tunnel establishment procedure is almost similar to the tunnel establishment procedure of the control channel. I1, R1, I2 and R2 messages have the same obligatory field as in the previous control channel tunnel establishment (Figure 5). However, DPS2 sends the REQ message to SecE via SecGW. It contains the HIs of DPSs and the echo request. Upon the arrival of REQ, SecE checks the received HIs with ACLs and TOE. Then, it replies the REQ message with an ACK message. The ACK message contains the HIs of the DPSs, the acknowledgment and the echo reply. Here, we use the same algorithm (Figure 6) to decide the content of the acknowledgment.

A positive acknowledgment is sent for a request from authorized DPSs and a negative acknowledgment is sent for other requests. If it is a negative acknowledgment, DPS2 drops the connection request from DPS1. Otherwise, DPS2 completes the tunnel establishment by sending the R2 message.

4. Performance Analysis

We implement the proposed architecture in a testbed and analyze the performance penalty of the added security on throughput, jitter and latency. We make the comparison with the performance of the OF protocol [6], which is the most widely utilized control protocol in SDN networks [47]. Figure 8 illustrates the experiment testbed.

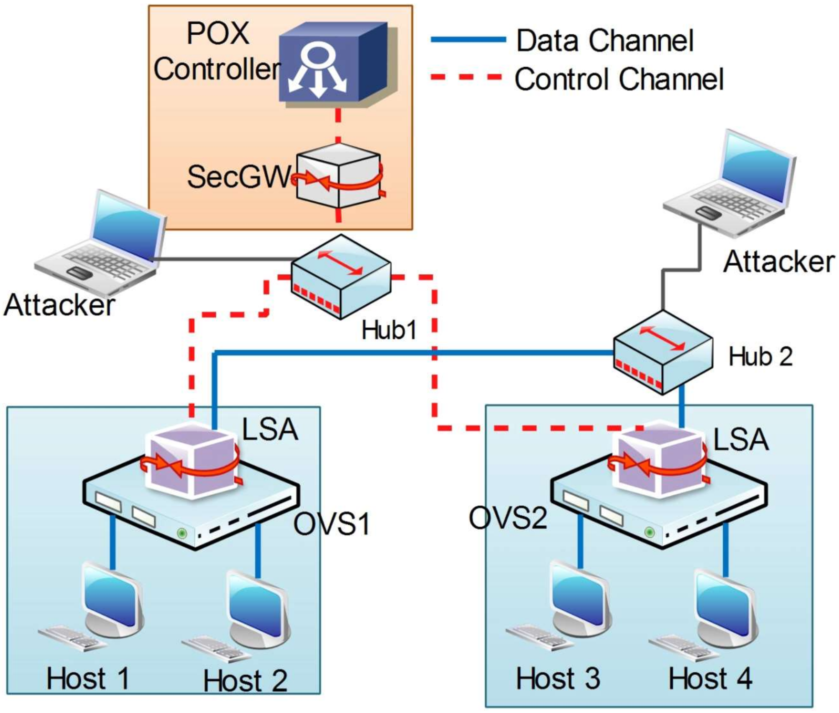


Figure 8: The layout of the experimental testbed

We use four laptops and two Ethernet hubs in the testbed. Two laptops with i5-3210M (2.5G Hz) CPUs are used as SDN switches. An OpenVswitch (OVS) version 1.10.0 [48] is installed in each laptop. Two virtual hosts (Host1 and Host2) are connected via OVS1 and they run Lubuntu 13.10 Operating System (OS). Similarly, two virtual hosts (Host3 and Host4) which run Lubuntu 13.10 OS, are connected via OVS2.

The third laptop with a L2400 CPU of 1.66GHz works as the SDN controller. The latest POX controller [49] runs on this laptop. All three laptops have Ubuntu 12.04 LTS OS. They are connected via two D-LINK DSR-250N routers. The link speed of this experiment is set to 100 Mbps.

The attacker is connected to each hub according to the experiment scenario. The attacker laptop also has a L2400 CPU of 1.66GHz and runs Ubuntu 12.04 LTS OS.

Finally, we use OpenHIP implementation [50] to model the SecGW and LSAs at the corresponding laptops. Here, the POX controller controls OVSs via OpenFlow version 1.1.0 [51]. Furthermore, OpenFlow version 1.1.0 uses TLSv1 to secure the control channel and we use it as our reference control channel.

In these experiments, the latency, throughput and jitter are measured by using the Internet Control Message Protocol (ICMP) ping requests and the IPERF network measurement tool [52]. Table 4 presents the simulation settings for IPERF testing tool.

Table 4: The simulation settings for the IPERF

Parameter	**Value**	**Value**
Protocol	UDP	TCP
Port	5004	5004
Buffer size	default (1470 kB)	default (1470 kB)
Packet size	default (1470 B)	default (1470 B)
TCP window size	-	21.0 KByte
Report interval	1 s	1 s

4.1. Performance Analysis of Control Channel

In the first set of experiments, we analyse the performance penalty of security on the SDMN control channel due to the proposed architecture.

4.1.1. Connection Establishment Delay

In the first experiment, we measure the connection establishment delay between OVS1 and the POX controller under different scenarios. Here, we try to send a ping request from Host1 to Host2 and measure the connection establishment delay.

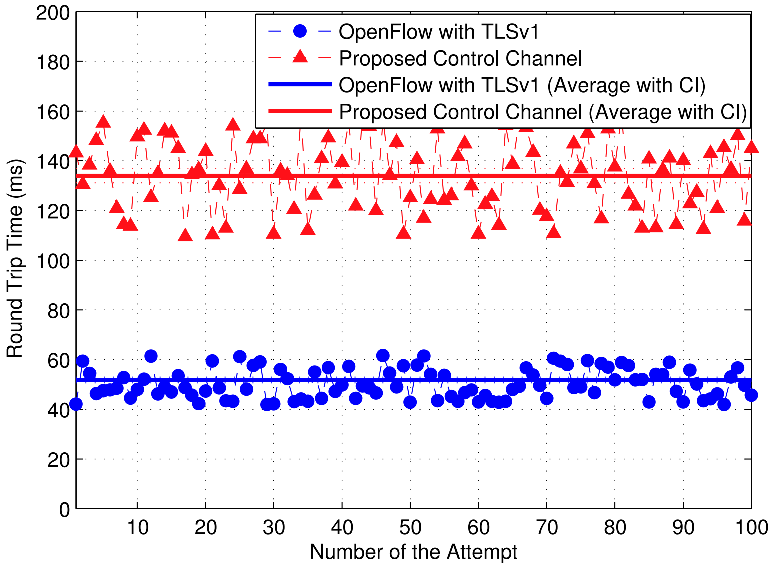


Figure 9: The connection establishment delay

Experiment results (Figure 9) reveal that the proposed secure architecture significantly increases (136%) the tunnel establishment delay. HIP tunnel establishments between LSA and SecGW adds extra delay to the tunnel establishment. However, the impact of this delay can be minimized by keeping the established HIP tunnels for a long period. It is possible to maintain established HIP tunnels for long periods (i.e. 15 mins) [46].

4.1.2. Flow Table Update Delay

In the second experiment, we measure the delay to update the flow tables for new packet flows in the steady state of operation. In the steady state of operation, HIP tunnels between LSAs and SecGW are already established and operational. Here, we ping from Host1 to Host2 and measure the Round Trip Time (RTT).

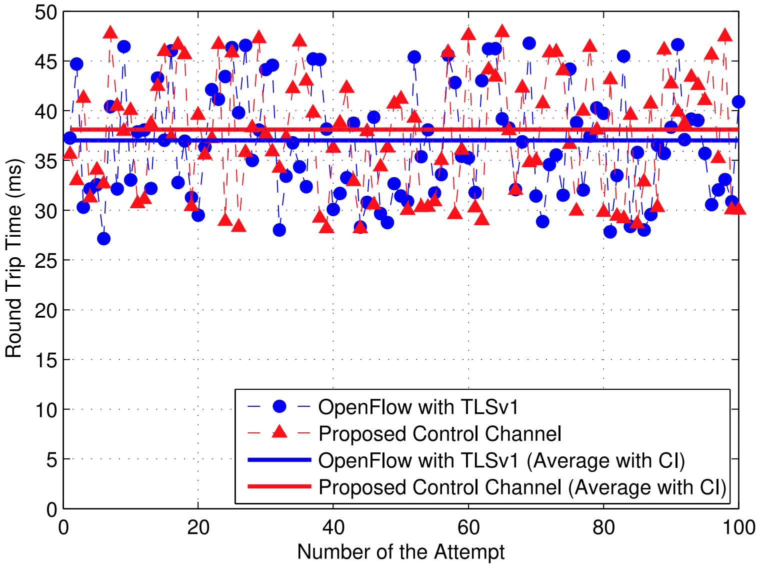


Figure 10: Flow Table Update Delay

Experiment results (Figure 10) reveals that the performance penalty of the proposed secure architecture is less significant in the steady state of operation. The extra IPsec encryption increases the flow update delay only by 2%. However, this delay can be further minimized by using IPsec accelerators. IPsec acceleration is possible by using external accelerators and/or using new AES (Advanced Encryption Standard) instruction sets for processors [53].

4.2. Performance Analysis of Data Channel

In the second set of experiments, we measure the TCP and UDP throughput performance of data channel under different scenarios.

4.2.1. Impact on TCP Throughput

In the third experiment, we establish a TCP connection between Host1 and Host3 to measure the TCP throughput performance of the data channel by using the IPERF tool[52].

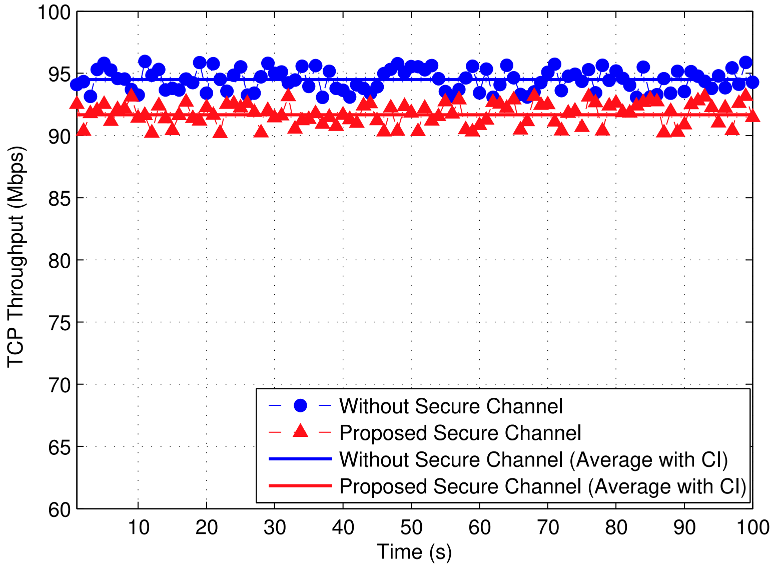


Figure 11: Performance penalty on TCP throughput

Experiment results (Figure 11) reveal that the proposed secure architecture decreases the TCP throughput only by 2.3%, compared with the non-secure data channel. The extra layer of encryption decreases the TCP Throughput.

4.2.2. *Impact on UDP Throughput*

In the fourth experiment, we establish a UDP connection between Host1 and Host3, to measure the UDP throughput performance of the data channel.

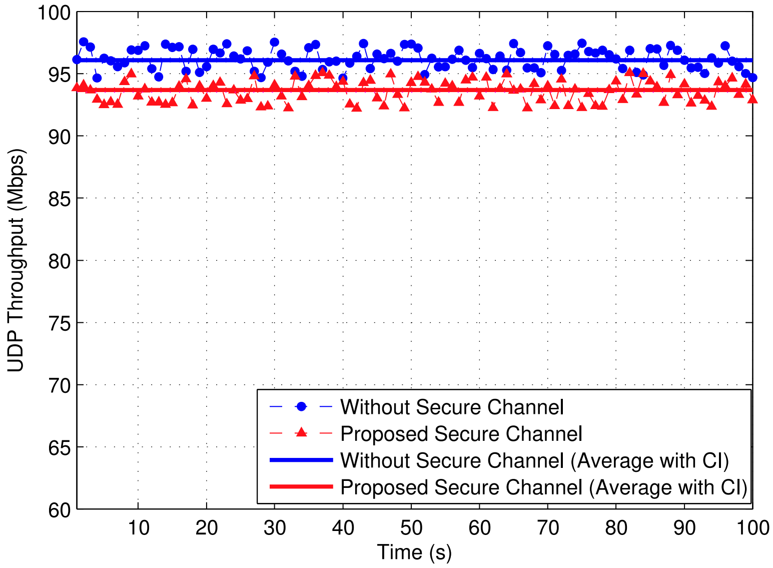


Figure 12: Performance penalty on UDP throughput

Experiment results (Figure 12) reveal that the proposed secure architecture decreases the UDP throughput only by

2.2% than the non-secure data channel. The extra layer of encryption decreases the UDP Throughput.

Moreover, the performance penalty of security on throughput is around 2% for both UDP and TCP sessions, compared with the non secure scenario. Thus, we can conclude that the performance penalty of security on throughput is independent of the transport layer protocol.

4.2.3. Impact on Jitter

In the fifth experiment, the jitter performance of a UDP session between Host1 and Host3 is measured by using the IPERF tool [52].

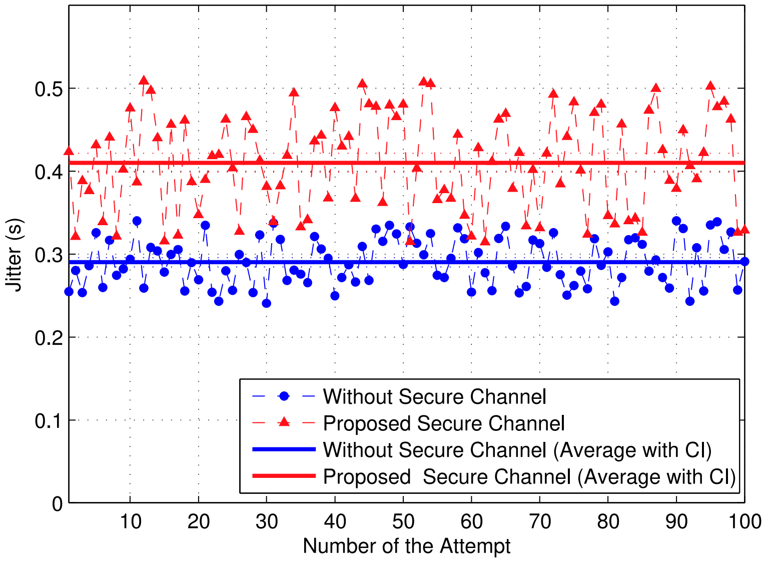


Figure 13: Performance penalty on Jitter

Experiment results (Figure 13) reveal that the performance penalty of secured architecture is 41% compared with the non secure data channel. However, the jitter is still far below $500\mu s$ (VoIP requires a jitter below 4 ms [54]) and the impact of jitter for real-time applications such as VoIP, video streaming is less significant in a short range network [19].

4.3. Overhead Analysis

The introduction of LSA, SecE and SecGWs will increase the overhead due to the additional signaling traffic and encryption headers. We analyzed the additional overhead on communication channels under two scenarios

1. Steady State : The tunnel is already established between LSA and SecGW for the control channel or between the LSAs for the data channels
2. Initial State: No tunnel is established.

We changed the file sizes from 5 KB to 1 MB for the control channel and from 5 KB to 1 MB for the data channel. The overhead percentage is calculated according to equation 1.

$$Overhead = \frac{Overhead}{FileSize} * 100\% \tag{1}$$

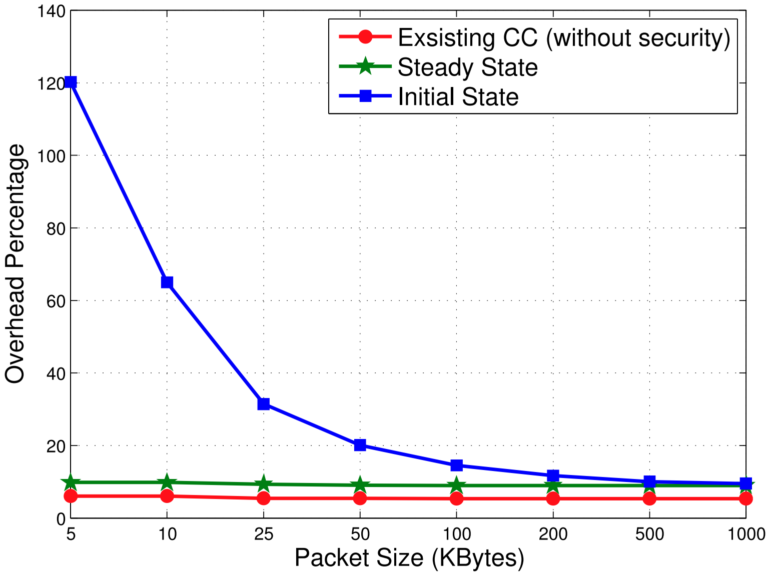


Figure 14: Overhead on Control Channel Traffic

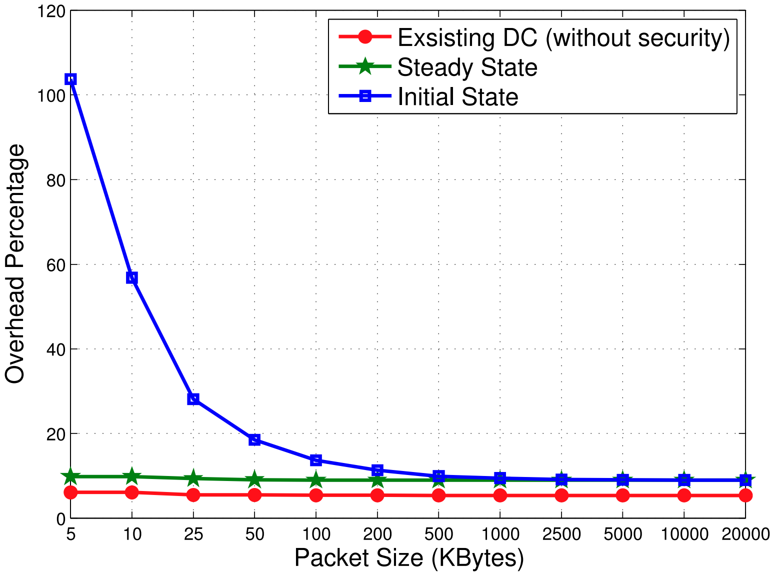


Figure 15: Overhead on Data Channel Traffic

Figure 14 shows the overhead penalty on the control channel and Figure 15 depicts the overhead penalty on the data channel. The proposed modifications increase the overhead only by 3% at steady state operation and it is less significant. The additional encryption headers are the main reason for this overhead. However, the overhead penalty is significant when we start the communication from the initial state. The additional tunnel establishment signaling adds extra overhead here. However the impact of signaling overhead is decreasing with the file size. Hence, this performance penalty can be minimized by keeping the established tunnels for a long period.

5. Security Analysis

In this section, we discuss the protection of the proposed architecture against various security attacks, cf. Table 2 .

5.1. Protection against DoS Attacks

There are three possible DoS attack categories. In the first category, the attacker sends an excessive amount of connection establishment requests (e.g. TCP SYN DoS attacks) to establish hanging connection with the controller or DPSs [7]. Such DoS attacks can be prevented by our architecture. If attackers send a series of I1 packets (Figure 5 and 7) to perform a DoS attack, the responder replies with a precomputed R1 packet for each I1 without allocating any resources such as memory space or server port. The responder allocates resources only after the arrival of a correct solution in the I2 message. Therefore, both the controller and DPSs are protected from DoS attacks.

In the second category, the attacker sends fake packets to the control and data channels by spoofing all or a part of the header fields with random values [14]. Such DoS attacks can also be prevented by our architecture. The proposed architecture use IPsec tunnels. Therefore, it is not possible to generate acceptable IPsec packets without knowing IPsec parameters such as encryption keys. Thus, end devices do not accept any random data packets from random users.

In the third category, the attacker dumps an excessive amount of junk traffic to overload the network bandwidth (e.g. UDP floods, ICMP floods [55]). The proposed architecture does not have an inbuilt mechanism to prevent the impact of such volume based DoS attacks. However, volume based DoS attacks can be prevented by implementing firewalls, ingress filtering and enforcing rate bounds [56]. Such mechanisms are used in most of the communication networks to prevent the impact of volume based DoS attacks. Thus, we recommend to implement them in edge DP switches and SecGWs.

To prove the power of our protocol, we experimentally compare the impact of TCP SYN DoS attacks in our architecture with the non secured one. In the experiment, we attach an external attacker to Hub1 of the testbed. The attacker performs TCP SYN flooding attack on the POX controller. The attacker (TCP packet generator) sends TCP SYN packets by changing port numbers and source IP addresses. The controller allocates one port for every successfully arrived SYN packet. Likewise, the attacker occupies all ports (64000 per user) and IP address combinations[57].

Here, we try to send a ping request from Host1 to Host2 and measure the RTT. The attack is placed between connection establishment request attempts 10 to 40. Each attempt sends 10 ping requests and only averages are presented in Figure 16.

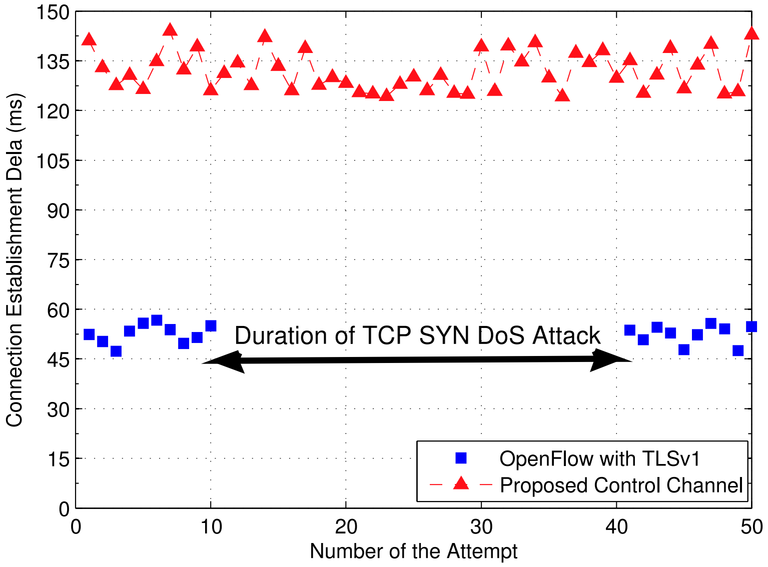


Figure 16: Impact of TCP SYN DoS attack

Experiment results (Figure 16) reveal that the proposed architecture protects the control channel from TCP SYN DoS attacks. However, TLSv1 based OF protocol is vulnerable to TCP SYN DoS attacks. It is not possible to establish a connection between OVS1 and POX controller during a TCP SYN DoS attack.

5.2. Protection against Reset Attacks

In reset attacks, the first step of the attacker is eavesdropping ongoing communication sessions to extract the session information. This session information is useful to make forged reset requests. For instance, the TCP attacker needs to match five packet header fields (source and destination IP addresses, source and destination ports and sequence number [58]. The attacker can successfully eavesdrop these header information from an unencrypted IP header.

The proposed architecture uses IPsec ESP mode messages in both control and data channel HIP tunnels. Therefore, all TCP/IP header and session information are encrypted. In that case, the attacker can not obtain enough information to reset the communication session. Therefore, the proposed architecture protects the communication channels from reset attacks.

We experimentally prove the resistance of our solution, both on control and data channel.

5.2.1. Impact of TCP Reset Attack on Control Channel

In this experiment, the attacker performs a TCP reset attack on both the POX controller and OVS1. The attacker sends fake TCP packets to both ends to reset the connection between them. However, the attacker must include correct IP addresses, port numbers and a valid sequence number in the header of the forged packet. Thus, the attacker eavesdrops the ongoing data and uses the eavesdropped information to generate convincing fake TCP packets [58].

Here, we ping from Host1 to Host2 and measure RTT. In this experiment, the attack is placed between ping attempt 10 to 40. Each attempt sends 10 ping requests and only averages are presented in the Figure 17.

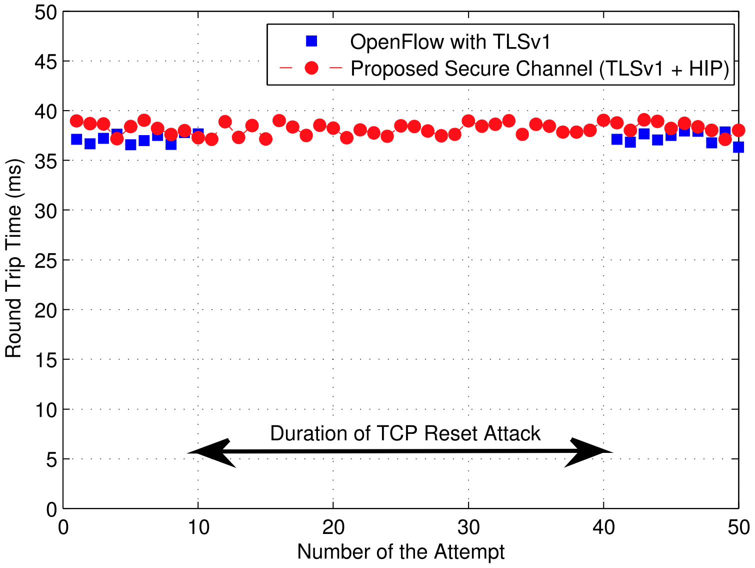


Figure 17: Impact of TCP Reset attack

Experiment results (Figure 17) reveal that the proposed architecture protects the control channel from TCP reset attack. However, TLSv1 based OpenFlow protocol is vulnerable to TCP reset DoS attack. It is not possible to update the flow tables since the reset attacker is always able to terminate the connection between OVS1 and POX controller.

5.2.2. Impact of TCP Reset Attack on Data Channel

In this experiment, the attacker performs a TCP reset attack on the data plane traffic which is routed via Hub2. Here, we establish a TCP connection between Host1 and Host3. The attacker sends fake TCP packets to both ends to reset the connection between hosts. In this experiment, the attack is placed between the time period from 20 s to 80 s. We measure the TCP throughput by using the IPERF tool[52].

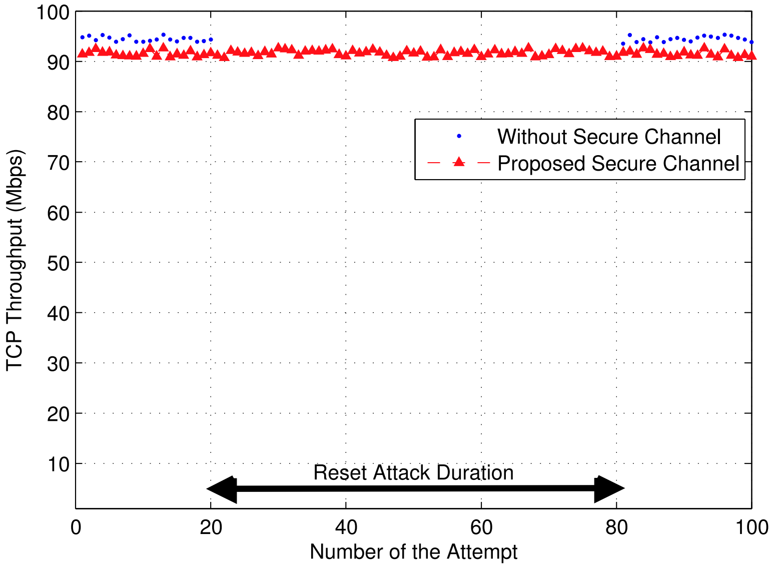


Figure 18: Impact of TCP Reset attack

Experiment results (Figure 17) reveal that the proposed architecture protects the data channel from TCP reset attacks. However, the non secure data channel is vulnerable to the TCP reset attacks. TCP throughput is dropping until zero during the attack. Therefore, the TCP attacker is able to reset the communication session between hosts in the existing SDMN.

5.3. Protection against IP Spoofing Attacks and Impersonation Attacks

The proposed mutual authentication mechanism uses Host Identity (a cryptographic key) to verify the identity of the node. Thus, the mutual authentication mechanism is capable of verifying the identity of the entity behind the IP address and it prevents IP spoofing attacks. Moreover, the proposed architecture uses an authorization mechanism based on ACLs (Section 3). It prevents the establishment of connections with random users.

Again, we experimentally compare the impact of an IP spoofing attack on the data channel between our architecture and the non secured one. In the experiment, we attach an external attacker to Hub2. The attacker performs IP spoofing attack on both DPSs. Here, the attacker impersonates as a DPS and broadcasts Internet Control Message Protocol (ICMP) request message with the intended victim's spoofed source IP. We implement ingress filters in both OVSs [56]. These filters allow each OVS to accept ICMP requests only from another OVS. Therefore, the IP spoofing attacker has to predict an IP address of another OVS to perform a successful attack.

We consider both random and subnet spoofing attacks. In a random spoofing attack, the attacker randomly generates an IP address within the whole IP address range. A subnet spoofing attacker randomly generates an IP address within the IP address range of the subnet (For instance, IP subnet of the mobile backhaul network). We change the subnet size (/16, /20 and /24) and perform the subnet spoofing attack.

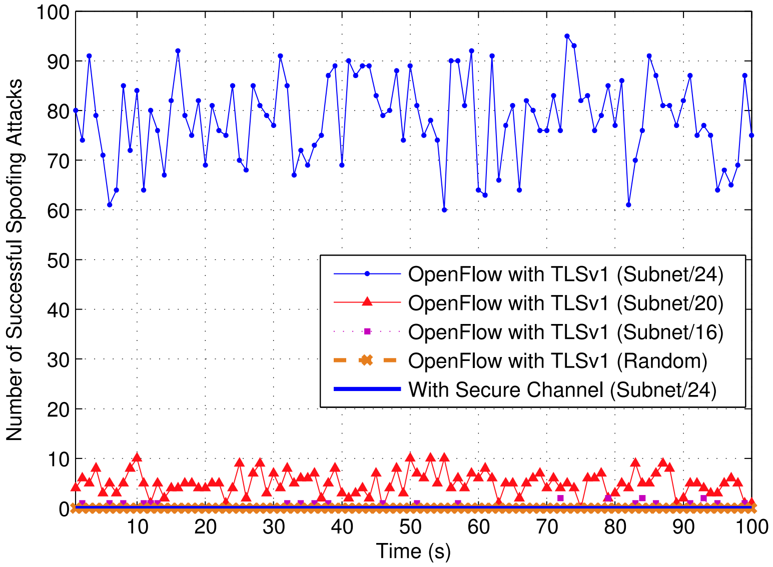


Figure 19: Impact of Spoofing attack

Experiment results (Figure 19) reveal that existing non secure SDMN data channel is vulnerable to subnet spoofing attacks. Moreover, the existing SDMN data channel is also vulnerable to random spoofing in few instances. The impact of random spoofing is negligible since the test bed contains only two DPSs. However, the impact of random spoofing can be significant in large networks with thousands of DP switches [59]. On the other hand, the proposed architecture protects the data channel from both subnet and random IP spoofing attacks.

5.4. Protection against Eavesdropping Attacks

Attackers eavesdrop the ongoing communication channels and use the eavesdropped information to perform various attacks such as IP spoofing, TCP reset and replay attacks. However, the proposed architecture uses HIP tunnels (IPsec BEET) in ESP mode for the data communication. Thus, the original IP headers, TCP headers and payload are always encrypted. It prevents possible eavesdropping attacks.

5.5. Protection against Message Modification Attacks

The proposed architecture uses HIP tunnels (IPsec BEET) in ESP mode for the communication channels. IPsec ESP mode provides connectionless integrity by using encrypted Integrity Check Value (ICV) field in the header [60]. Therefore, SDMN backhaul nodes can identify the modified messages in the communication channel and drop them without processing. It prevents possible message modification attacks. Moreover, the tunnel establishment messages (Figure 5 and 7) use HMAC (Hash Message Authentication Code) to ensure the integrity.

5.6. Protection against Replay Attacks

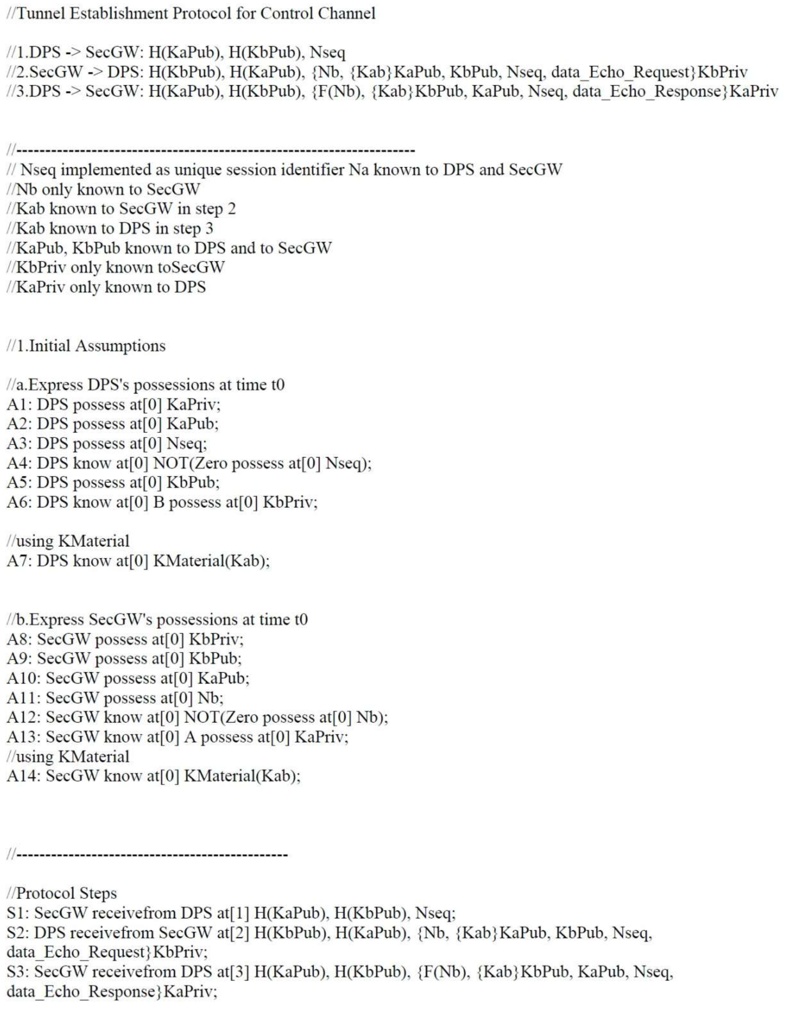
Most of the replay attacks are targeting connection establishment processes in both control and data channels. Attackers reply the captured connection establishment messages to establish an unauthorized connection. However, the proposed architecture uses the following mechanisms against replay attacks during the connection establishment phases (Figure 5 and 7). Virtue of the stateless response to I1 messages with pre-calculated R1 messages is used to protect responders against attacker's replays of I1 messages. A monotonically increasing "R1 generation counter" which is included in R1, is used to protect the initiator from R1 replays. Again, responders are protected against attacker's replays of I2 messages by using the puzzle mechanism and optional use of opaque data. Finally, a monotonically increasing "R2 generation counter" is used to protect the initiator from R2 replays.

On the other hand, replay attacks are possible during the flow table update phase. However, IPsec ESP (Encapsulating Security Payload) mode messages are used for flow table updates in the proposed architecture. IPsec ESP mode utilizes sequence numbers to protect the messages against replay attacks [60]. Thus, the attacker's replays of an IPSec encrypted packet will be rejected due to the sequence number mismatch at the end users.

5.6.1. Formal Analysis against of replay and parallel session attacks

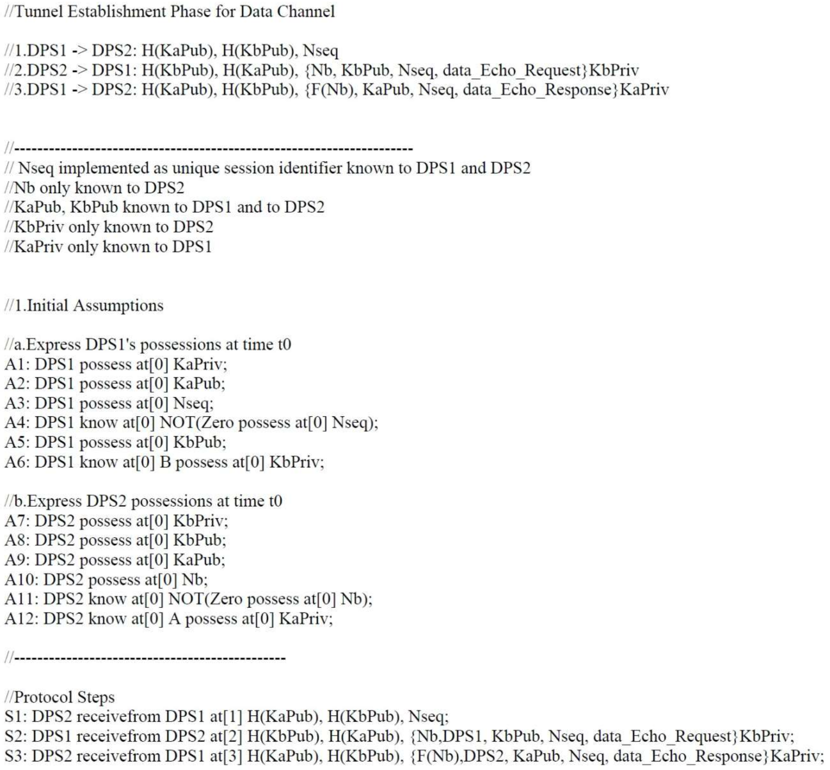
To formally prove the protection against replay attacks in our proposed architecture, we analysed the connection establishment phases presented in Figure 5 and 7. We used the CDVD/AD logic-based verification tool with attack detection capabilities against replay and parallel session attacks [61], in order to establish the correctness of the authentication session in the Tunnel Establishment for Control Channel and respectively, for the Data Channel.

Prior to verification, both authentication sessions (presented in Fig 5 and 7) were formalised (see Figure 20 and 21), i.e. translated into the language of the tool (i.e initial assumptions- conditions that hold before the session starts; and the phase steps- the messages exchanges between the principals).



```
//Tunnel Establishment Protocol for Control Channel

//1.DPS -> SecGW: H(KaPub), H(KbPub), Nseq
//2.SecGW -> DPS: H(KbPub), H(KaPub), {Nb, {Kab}KaPub, KbPub, Nseq, data_Echo_Request}KbPriv
//3.DPS -> SecGW: H(KaPub), H(KbPub), {F(Nb), {Kab}KbPub, KaPub, Nseq, data_Echo_Response}KaPriv


//--------------------------------------------------------------------
// Nseq implemented as unique session identifier Na known to DPS and SecGW
//Nb only known to SecGW
//Kab known to SecGW in step 2
//Kab known to DPS in step 3
//KaPub, KbPub known to DPS and to SecGW
//KbPriv only known toSecGW
//KaPriv only known to DPS


//1.Initial Assumptions

//a.Express DPS's possessions at time t0
A1: DPS possess at[0] KaPriv;
A2: DPS possess at[0] KaPub;
A3: DPS possess at[0] Nseq;
A4: DPS know at[0] NOT(Zero possess at[0] Nseq);
A5: DPS possess at[0] KbPub;
A6: DPS know at[0] B possess at[0] KbPriv;

//using KMaterial
A7: DPS know at[0] KMaterial(Kab);


//b.Express SecGW's possessions at time t0
A8: SecGW possess at[0] KbPriv;
A9: SecGW possess at[0] KbPub;
A10: SecGW possess at[0] KaPub;
A11: SecGW possess at[0] Nb;
A12: SecGW know at[0] NOT(Zero possess at[0] Nb);
A13: SecGW know at[0] A possess at[0] KaPriv;
//using KMaterial
A14: SecGW know at[0] KMaterial(Kab);



//-----------------------------------------------

//Protocol Steps
S1: SecGW receivefrom DPS at[1] H(KaPub), H(KbPub), Nseq;
S2: DPS receivefrom SecGW at[2] H(KbPub), H(KaPub), {Nb, {Kab}KaPub, KbPub, Nseq,
data_Echo_Request}KbPriv;
S3: SecGW receivefrom DPS at[3] H(KaPub), H(KbPub), {F(Nb), {Kab}KbPub, KaPub, Nseq,
data_Echo_Response}KaPriv;
```

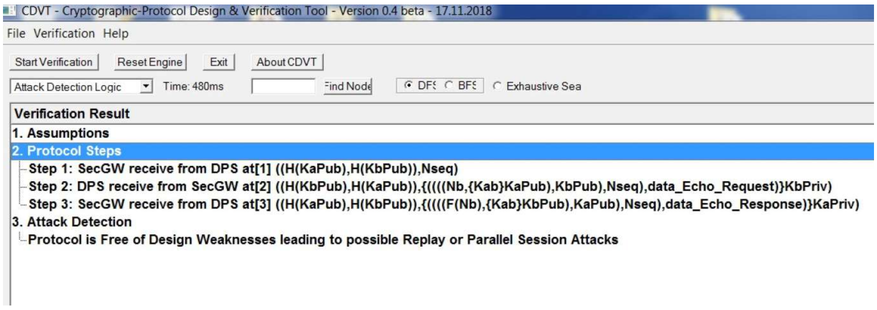
Figure 20: Formal Spec of authentication session for Control Channel



```
//Tunnel Establishment Phase for Data Channel

//1.DPS1 -> DPS2: H(KaPub), H(KbPub), Nseq
//2.DPS2 -> DPS1: H(KbPub), H(KaPub), {Nb, KbPub, Nseq, data_Echo_Request}KbPriv
//3.DPS1 -> DPS2: H(KaPub), H(KbPub), {F(Nb), KaPub, Nseq, data_Echo_Response}KaPriv


//----------------------------------------------------------------------
// Nseq implemented as unique session identifier known to DPS1 and DPS2
//Nb only known to DPS2
//KaPub, KbPub known to DPS1 and to DPS2
//KbPriv only known to DPS2
//KaPriv only known to DPS1


//1.Initial Assumptions

//a.Express DPS1's possessions at time t0
A1: DPS1 possess at[0] KaPriv;
A2: DPS1 possess at[0] KaPub;
A3: DPS1 possess at[0] Nseq;
A4: DPS1 know at[0] NOT(Zero possess at[0] Nseq);
A5: DPS1 possess at[0] KbPub;
A6: DPS1 know at[0] B possess at[0] KbPriv;

//b.Express DPS2 possessions at time t0
A7: DPS2 possess at[0] KbPriv;
A8: DPS2 possess at[0] KbPub;
A9: DPS2 possess at[0] KaPub;
A10: DPS2 possess at[0] Nb;
A11: DPS2 know at[0] NOT(Zero possess at[0] Nb);
A12: DPS2 know at[0] A possess at[0] KaPriv;

//-------------------------------------------------

//Protocol Steps
S1: DPS2 receivefrom DPS1 at[1] H(KaPub), H(KbPub), Nseq;
S2: DPS1 receivefrom DPS2 at[2] H(KbPub), H(KaPub), {Nb,DPS1, KbPub, Nseq, data_Echo_Request}KbPriv;
S3: DPS2 receivefrom DPS1 at[3] H(KaPub), H(KbPub), {F(Nb),DPS2, KaPub, Nseq, data_Echo_Response}KaPriv;
```

Figure 21: Formal Spec of authentication session for Data Channel

The results of the automated verification for the control channel session and for the data channel are shown in Figure 22 and 23. As can be seen, the outcome for the attack detection verification is free of any message indicating a weakness in the protocols design that can be exploited by mountable replay or parallel session attacks. This provides confidence in the correctness and effectiveness of the presented phases.



Verification Result

1. Assumptions

2. Protocol Steps

Step 1: SecGW receive from DPS at[1] ((H(KaPub),H(KbPub)),Nseq)

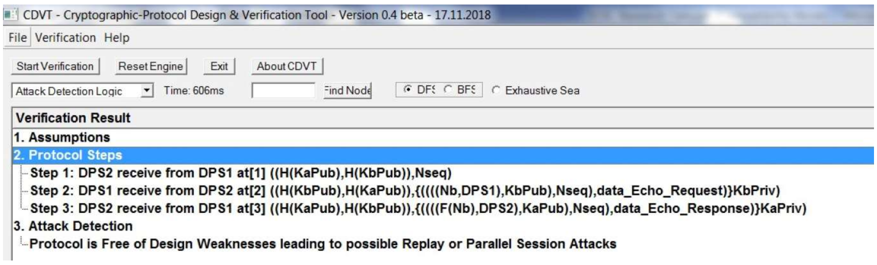
Step 2: DPS receive from SecGW at[2] ((H(KbPub),H(KaPub)),{((((Nb,{Kab}KaPub),KbPub),Nseq),data_Echo_Request)}KbPriv)

Step 3: SecGW receive from DPS at[3] ((H(KaPub),H(KbPub)),{((((F(Nb),{Kab}KbPub),KaPub),Nseq),data_Echo_Response)}KaPriv)

3. Attack Detection

Protocol is Free of Design Weaknesses leading to possible Replay or Parallel Session Attacks

Figure 22: Test Results for Authentication Session of the Control Channel



CDVT - Cryptographic-Protocol Design & Verification Tool - Version 0.4 beta - 17.11.2018
File
Verification
Help
Start Verification
Reset Engine
Exit
About CDVT
Find Node
Exhaustive Sea

Attack Detection Logic   Time: 606ms   ● DFS ○ BFS

```
Verification Result
1. Assumptions
2. Protocol Steps
 ├Step 1: DPS2 receive from DPS1 at[1] ((H(KaPub),H(KbPub)),Nseq)
 ├Step 2: DPS1 receive from DPS2 at[2] ((H(KbPub),H(KaPub)),{((((Nb,DPS1),KbPub),Nseq),data_Echo_Request)}KbPriv)
 └Step 3: DPS2 receive from DPS1 at[3] ((H(KaPub),H(KbPub)),{((((F(Nb),DPS2),KaPub),Nseq),data_Echo_Response)}KaPriv)
3. Attack Detection
 └Protocol is Free of Design Weaknesses leading to possible Replay or Parallel Session Attacks
```

Figure 23: Test Results for Authentication Session for the Data Channel Results

6. Discussion

6.1. Expected Benefits of Proposed Architecture

6.1.1. Security Gateway (SecGW)

The utilization of SecGWs provides two main benefits. First, the security mechanism is independent of the controller and the control protocol. Therefore, the controller or the control protocol can be upgraded or changed without modifying the security mechanism and vice versa. Second, it reduces cost of the controller and the security related workload of the controller. Without SecGWs, the controller should consist of integrated security specific hardware such as firewalls, IPsec accelerators, IDS to support high speed security functions. However, the integration of such hardware to the controller increases the complexity and the cost of the controller. The proposed architecture separates the security functions from the controller. It helps to develop low cost controllers and high performing SecGWs.

6.1.2. Distributed SecGWs

There are three main reasons to utilize a distributed SecGW mechanism. First, it avoids a single point of failure. Second, distributed SecGWs split SDMN backhaul into different independent slices. The different DPSs in a backhaul network face a different set of security threats and they require different levels of security. By separating them into different slices, it is possible to implement extra security mechanisms for such highly vulnerable devices. For instance, security keys of gateway switches which are connected to the Internet or roaming networks can be updated more frequently than other switches. Third, the security related functions use higher processing power than other network functions. The distributed security mechanism distributes the security workload among multiple SecGWs.

6.1.3. Security Entity (SecE)

The use of a separate security entity in SDMN architecture provides three main advantages. First, it reduces the security related overhead on other control entities in the controller. Second, it logically centralizes the security mechanisms. Thus, SecE can optimize the security resources by eliminating overlapping and redundant security services. Third, SecE can cooperate with other control entities to provide required security services. For instance, secure mobility can be enabled with the cooperation of mobility management entity.

6.1.4. IPsec Tunnelling for Communication Channels

IPsec tunnel establishment offers several security features such as confidentiality, data-origin authentication, connectionless integrity, anti-replay service and limited traffic-flow confidentiality for both the control and data channels.

6.1.5. Session based Key Management for IPsec Tunnels

Session based key management provides three main advantages. First, SDMN controller (SecE) obtains the global visibility of IPsec traffic transportation. Therefore, the controller can dynamically change the traffic routing path to optimize the network capacity. For instance, the controller can offload the user traffic at the earliest point possible even at the access or aggregation networks. Second, it helps DPSs to send the same control request to multiple SecGWs without wasting DPS resources. Since a common key used to encrypt the traffic in the control channel, DPS does not need to encrypt the same control message with different keys. It just needs to encrypt the control message once and replicate it as required. It also reduces the packet processing delay at DPS. Third, DPSs can send the same DP traffic via any DP switches according to the flow rules installed by the controller. Moreover, it is possible to replicate DP packets at any backhaul device or modify the traffic flow routes without disconnecting established IPsec tunnels. Table 5 contains the comparison of the proposed architecture with existing IPsec security mechanisms.

Table 5: Comparison of proposed architecture with existing IPSec security mechanisms

Property	**TLS/SSL Communication**	**IPSec Tunnelling with IKEv2**	**IPSec Tunnelling with Mobike**	**IPSec Tunnelling with HIP**	**Proposed Architecture**
Vulnerability of mutual authentication mechanism	Medium	Medium	Medium	Low	Low
DoS attack prevention	No	No	No	Yes	Yes
Support for seamless mobility of backhaul nodes	No	No	Yes:Limited	Yes	Yes
Multihomed Support	No	No	No	Yes:Limited	Yes
Centralized Controlling	No	No	No	No	Yes
Point-to-Multipoint/ Multipoint-to-Multipoint	No	No	No	No	Yes
Visibility of traffic transportation	No	No	No	No	Yes
Access Control	No	No	No	No	Yes
Collaboration with other control entities (e.g. TOE)	No	No	No	No	Yes

6.2. Limitations of proposed architecture

The proposed architecture has two limitations. First limitation is the introduction of new elements such as LSA, SecE and SecGWs. The newly introduced LSAs will be added to each DPS. Thus, it will increase the cost of DPSs. Moreover, DPSs are required to have additional processing power to support encryption functions. However, it is possible to develop LSAs as separate boxes and route the DPS's traffic via this box. SecGWs are added to the controller. It can be developed as an external plug and play device. Thus, it will not increase the cost of the controller. The introduction of SecE has the least impact. It will be a software application which runs on top of the controller. It has the same behavior as other control entities.

The introduction of LSA, SecE and SecGWs will increase the overhead due to the additional signaling traffic and encryption headers. However, this performance penalty can be minimized by keeping the established tunnels for a long period.

Second limitation is the vulnerability to volume based DoS attacks. Our architecture is unable to prevent volume based DoS attacks. However, volume based DoS attacks can be prevented by implementing firewalls, ingress filtering and enforcing rate bounds [56].

7. Conclusion

We studied the security challenges of the communication channels in SDMNs and the applicability of IPsec tunneling mechanisms to secure it. We proposed a novel IPsec based secure communication channel architecture by using HIP. We presented the implementation of IPsec tunnels to secure SDMN communication channels. Finally, we analyzed the security features and the performance of the proposed architecture in a real testbed. Experiment results revealed that the proposed architecture protects the communication channels against IP based attacks such as DoS, reset, spoofing, replay and eavesdropping attacks. However, there is a performance penalty of security on throughput, latency and jitter due to the extra IPsec tunnel establishment. This drawback can be minimized by using security specific hardware and maintaining established HIP tunnels for a longer period.

In future, we will focus on how to utilize cloud resources to enhance the performance of the proposed IPsec tunneling architecture.

Acknowledgment

This work has been performed in the framework of the CELTIC project CP2012 SIGMONA. The authors would like to acknowledge the contributions of their colleagues. This information reflects the consortium's view, but the consortium is not liable for any use that may be made of any of the information contained therein. This research is funded by TEKES, Finland.

References

[1] C. Kolias, S. Ahlawat, C. Ashton, , et al., OpenFlow-Enabled Mobile and Wireless Networks, *White Paper* (September 2013).

[2] H. Hawilo, A. Shami, M. Mirahmadi, R. Asal, NFV: State of the Art, Challenges and Implementation in Next Generation Mobile Networks (vEPC), arXiv preprint arXiv:1409.4149.

[3] K. Pentikousis, Y. Wang, W. Hu, Mobileflow: Toward software-defined mobile networks, Communications Magazine, IEEE 51 (7).

[4] A. Y. Ding, J. Crowcroft, S. Tarkoma, H. Flinck, Software defined networking for security enhancement in wireless mobile networks, Computer Networks 66 (2014) 94–101.

[5] Worldwide Infrastructure Security Report (WISR) 2012, *Survey Report* (2013).

[6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, OpenFlow: enabling innovation in campus networks, ACM SIGCOMM Computer Communication Review 38 (2) (2008) 69–74.

[7] D. Kreutz, F. Ramos, P. Verissimo, Towards secure and dependable software-defined networks, in: Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, ACM, 2013, pp. 55–60.

[8] M. McBride, M. Cohn, S. Deshpande, , et al., SDN Security Considerations in the Data Center, *White Paper* (October 2013).

[9] C. Meyer, J. Schwenk, Lessons Learned From Previous SSL/TLS Attacks-A Brief Chronology Of Attacks And Weaknesses, IACR Cryptology ePrint Archive 2013 (2013) 49.

[10] P. Fonseca, R. Bennesby, E. Mota, A. Passito, A replication component for resilient OpenFlow-based networking, in: Network Operations and Management Symposium (NOMS), 2012 IEEE, IEEE, 2012, pp. 933–939.

[11] OpenFlow Switch Specification Version 1.4.0.
URL `https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf`

[12] K. Benton, L. J. Camp, C. Small, Openflow vulnerability assessment, in: Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, ACM, 2013, pp. 151–152.

[13] A. N. Bikos, N. Sklavos, LTE/SAE security issues on 4G wireless networks, Security & Privacy, IEEE 11 (2) (2013) 55–62.

[14] S. Shin, 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, ACM, 2013, pp. 165–166.

[15] S. Scott-Hayward, G. O'Callaghan, S. Sezer, SDN security: A survey, in: IEEE SDN for Future Networks and Services (SDN4FNS) conference, IEEE, 2013, pp. 1–7.

[16] D. Forsberg, G. Horn, W.-D. Moeller, V. Niemi, LTE Security, Vol. 1, John Wiley & Sons, 2012.

[17] M. Ilyas, S. A. Ahson, Handbook of wireless local area networks: Applications, Technology, Security, and Standards, CRC Press, 2005.

[18] J. Cao, M. Ma, H. Li, Y. Zhang, Z. Luo, A Survey on Security Aspects for LTE and LTE-A Networks, Communications Surveys & Tutorials, IEEE 16 (1) (2014) 283–302.

[19] 3GPP (2013) Policy and charging control architecture (Release 12), TS 23.203.
URL `\url{http://www.3gpp.org/DynaReport/23203.htm}`

[20] D. Harkins, D. Carrel, The Internet Key Exchange (IKE), RFC 2409 (November 1998).

[21] C. Kaufman, Internet Key Exchange (IKEv2) Protocol, RFC 4306 (December 2005).

[22] P. Eronen, IKEv2 Mobility and Multihoming Protocol (MOBIKE), RFC 4555 (June 2006).

[23] R. Moskowitz, T. Heer, P. Nikander, P. Jokela, Host Identity Protocol Version 2 (HIPv2), RFC 7401 (2015).

[24] A. Khurshid, W. Zhou, M. Caesar, P. Godfrey, Veriflow: Verifying network-wide invariants in real time, ACM SIGCOMM Computer Communication Review 42 (4) (2012) 467–472.

[25] Z. Dai, F. Wang, H. Deng, A Survey on Real-Time Secure Transport System, in: Intelligent Networks and Intelligent Systems (ICINIS), 2013 6th International Conference on, IEEE, 2013, pp. 328–331.

[26] E. E. Chinedum, S. Zhang, et al., Prevalent Network Threats and Telecommunication Security Challenges and Countermeasures in VoIP Networks, Network and Complex Systems 3 (3) (2013) 49–55.

[27] R. Moskowitz, P. Nikander, P. Jokela, Host Identity Protocol, RFC 5201 (2008).

[28] S. Shin, P. A. Porras, V. Yegneswaran, M. W. Fong, G. Gu, M. Tyson, FRESCO: Modular Composable Security Services for Software-Defined Networks., in: NDSS, 2013.

[29] A. Khurshid, W. Zhou, M. Caesar, P. Godfrey, Veriflow: verifying network-wide invariants in real time, ACM SIGCOMM Computer Communication Review 42 (4) (2012) 467–472.

[30] E. Al-Shaer, S. Al-Haj, FlowChecker: Configuration analysis and verification of federated OpenFlow infrastructures, in: Proceedings of the 3rd ACM workshop on Assurable and usable security configuration, ACM, 2010, pp. 37–44.

[31] Security-enhanced floodlight.
URL `http://www.sdncentral.com/education/toward-secure-sdn-controllayer/,2013/10/`

[32] R. Braga, E. Mota, A. Passito, Lightweight DDoS flooding attack detection using NOX/OpenFlow, in: Local Computer Networks (LCN), 2010 IEEE 35th Conference on, IEEE, 2010, pp. 408–415.

[33] P. Fonseca, R. Bennesby, E. Mota, A. Passito, A replication component for resilient OpenFlow-based networking, in: Network Operations and Management Symposium (NOMS), 2012 IEEE, IEEE, 2012, pp. 933–939.

[34] J. Kempf, E. Bellagamba, A. Kern, D. Jocha, A. Takács, P. Skoldstrom, Scalable fault management for OpenFlow, in: Communications (ICC), 2012 IEEE International Conference on, IEEE, 2012, pp. 6606–6610.

[35] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, G. Gu, A security enforcement kernel for OpenFlow networks, in: Proceedings of the first workshop on Hot topics in software defined networks, ACM, 2012, pp. 121–126.

[36] K. Phemius, M. Bouet, J. Leguay, Disco: Distributed multi-domain sdn controllers, in: Network Operations and Management Symposium (NOMS), 2014 IEEE, IEEE, 2014, pp. 1–4.

[37] Y. Zhang, N. Beheshti, M. Tatipamula, On resilience of split-architecture networks, in: Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE, IEEE, 2011, pp. 1–6.

[38] A. K. Nayak, A. Reimers, N. Feamster, R. Clark, Resonance: dynamic access control for enterprise networks, in: Proceedings of the 1st ACM workshop on Research on enterprise networking, ACM, 2009, pp. 11–18.

[39] H. Hu, W. Han, G.-J. Ahn, Z. Zhao, FLOWGUARD: building robust firewalls for software-defined networks, in: Proceedings of the third workshop on Hot topics in software defined networking, ACM, 2014, pp. 97–102.

[40] J. R. Ballard, I. Rae, A. Akella, Extensible and scalable network monitoring using opensafe, Proc. INM/WREN.

[41] S. A. Mehdi, J. Khalid, S. A. Khayam, Revisiting traffic anomaly detection using software defined networking, in: Recent Advances in Intrusion Detection, Springer, 2011, pp. 161–180.

[42] Z. A. Qazi, C.-C. Tu, L. Chiang, R. Miao, V. Sekar, M. Yu, SIMPLE-fying middlebox policy enforcement using SDN, in: ACM SIGCOMM Computer Communication Review, Vol. 43, ACM, 2013, pp. 27–38.

[43] J. Costa-Requena, J. L. Santos, V. F. Guasch, K. Ahokas, G. Premsankar, S. Luukkainen, I. Ahmed, M. Liyanage, M. Ylianttila, O. L. Prez, M. U. Itzazelaia, E. M. de Oca, SDN and NFV Integration in Generalized Mobile Network Architecture, in: European Conference on Networks and Communications (EUCNC), IEEE, 2015, pp. 1–5.

[44] P. Jokela, R. Moskowitz, J. Melen, Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP), RFC 7402 (2015).

[45] P. Jokela, R. Moskowitz, P. Nikander, Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP), RFC 5202 (2008).

[46] A. Gurtov, Host Identity Protocol (HIP): Towards the Secure Mobile Internet, Wiley, 2008.

[47] A. Lara, A. Kolasani, B. Ramamurthy, Network innovation using openflow: A survey, Communications Surveys & Tutorials.

[48] Open vSwitch: An Open Virtual Switch.
URL `\url{http://openvswitch.org/}`

[49] About POX.
URL `http://www.noxrepo.org/pox/about-pox/`

[50] The OpenHIP project.
URL `\url{http://www.openhip.org/}`

[51] OpenFlow Switch Specification Version 1.1.0.
URL `http://archive.openflow.org/documents/openflow-spec-v1.1.0.pdf`

[52] Iperf.
URL `http://iperf.sourceforge.net/`

[53] Carrier Cloud Telecoms - Exploring the Challenges of Deploying Virtualisation and SDN in Telecom Networks, Tech. rep., Intel Cooperation (2013).

[54] M. Poikselkä, G. Mayer, The IMS: IP multimedia concepts and services, John Wiley & Sons, 2013.

[55] N. Panwar, P. Kumar, C. Raj, Data Communication: Overview of TCP/IP Protocol Suite in Security Complication and their Safeguard, International Journal of Research 1 (9) (2014) 1417–1424.

[56] R. K. Chang, Defending against flooding-based distributed denial-of-service attacks: A tutorial, Communications Magazine, IEEE 40 (10) (2002) 42–51.

[57] W. Eddy, TCP SYN flooding attacks and common mitigations, RFC 4987 (2007).

[58] P. A. Watson, Slipping in the Window: TCP Reset attacks, Tech. rep. (2004).

[59] W. Chen, D.-Y. Yeung, Throttling spoofed SYN flooding traffic at the source, Telecommunication Systems 33 (1-3) (2006) 47–65.

[60] S. Kent, IP Encapsulating Security Payload (ESP), RFC 4303 (December 2005).

[61] A. D. Jurcut, T. Coffey, R. Dojen, On the Prevention and Detection of Replay Attacks Using a Logic-Based Verification Tool, Computer Networks (2014) 128–137.