

Novel framework for enhancing security of SDN based VPLS architecture

Kuntal Gaur[@]

*Department of Computer Applications
Manipal University Jaipur
Jaipur
Rajasthan
India*

Umashankar Rawat[†]

*Department of Computer Science and Engineering
Manipal University Jaipur
Jaipur
Rajasthan
India*

Saket Acharya[§]

*Member-ACM
Jaipur
Rajasthan
India*

Pradeep Kumar^{*}

*Department of Computer Applications
Manipal University Jaipur
Jaipur
Rajasthan
India*

Anshuman Kalla[‡]

*Department of Computer Engineering
Chhotubhai Gopalbhai Patel Institute of Technology
Uka Tarsadia University
Bardoli
Gujarat
India*

[@] E-mail: kuntal.gaur@jaipur.manipal.edu

[†] E-mail: Umashankar.rawat@jaipur.manipal.edu

[§] E-mail: saket590@gmail.com

^{*} E-mail: pradeep.kumar@jaipur.manipal.edu (Corresponding Author)

[‡] E-mail: anshuman.kalla@ieee.org

Abstract

Software-Defined Networking (SDN) is an emerging technology that enables the extension of a single Ethernet broadcast domain over a wide area network. However, there are still several major network security threats that could lead to network and resource unavailability, man-in-the-middle attacks, cryptographic flaws, and other vulnerabilities. This paper presents a method that proposes the use of Trusted Platform Module (TPM)-based SDN Virtual Private LAN Services (VPLS), which is intended to provide a secure solution that can help mitigate various network attacks. Open VSwitch (OVS) is a virtual switch that can be used to implement SDN-based VPLS networks. Direct Anonymous Attestation (DAA) is used to verify the authenticity of hardware devices without revealing any identifying information about the devices. The proposed architecture is implemented in a testbed to analyze the performance. The results show that TPM can be utilized in an SDN network to enhance security by securely storing encryption keys and certificates.

Subject Classification: Primary 93A30, Secondary 49K15.

Keywords: SDN, VPLS, TPM, DDoS, Security.

1. Introduction

The extension and expansion of industries and enterprises has triggered an utmost requirement of securely connecting workplaces that are placed geographically apart. Virtual Private Network (VPN) has surfaced as a cost-efficient solution to this challenge. The VPN works on the concept of logically partitioning the shared networking infrastructure to form a closed virtual network which can be accessed by sites that are far apart. The VPNs are classified into three subcategories- Layer1(L1), Layer2(L2) and Layer3(L3), depending on which layer of the Open Systems Interconnection (OSI) it is implemented at. Conventionally, L2VPN incurs lesser operational cost and are more compatible than L3VPNs.

One of the convenient ways of provisioning an L2VPN is VPLS. The features like cost efficient operations and protocol independence works in favor of the VPLS. VPLS is an optimized traditional Transparent LAN Services (TLS). Many architectures for implementing VPLS had been proposed, but security remained one of the major concerns of VPLS. In the recent development in VPLS implementation SDN was used. SDN embraces the notion of network programmability. Hence, all the operations of the network need to be presented as software programs, algorithms, data structures and by the concepts that are available for the software development. Software Defined VPLS (SDVPLS) extends better security for VPLS implementation along with enhanced scalability. TPM is a hardware security module used to securely store the cryptographic keys,

passwords, and other sensitive data. TPM in SDVPLS can be used to securely store critical information. This can help in ensuring that VPLS is secure.

In summary, TPM can be used in an SDVPLS to enhance security by securely storing encryption keys, certificates, and other sensitive data. This can help ensure that the VPLS is secure, and that the data being transmitted over it is protected.

The rest of the research article is structured as follows:

Section II discusses the literature review of similar studies. In Section III, the proposed framework based on SDN-VPLS-TPM has been described. Section IV covers the implementation of the proposed framework. Section V shows the comparison between SDN-VPLS and SDN-VPLS-TPM. In the end, Section VI concludes the paper.

2. Literature Review

Over the years, the field of SDN and SDVPLS have been the subject of extensive research. In order to enhance the security of SDVPLS, researchers have put forward numerous models and techniques. Some of the most notable works in this area are briefly discussed below.

Behzad et.al. [1] suggested SDxVPN, an SDN-based technique that decreases the complexity of VPN service management. However, the suggested methodology is not secure against several attacks, including DoS and cryptography attacks.

Using OpenVirteX as the foundation, Hu et.al. [2] suggested L2OVX to use the VPLS service ad-hoc SDN. Although the method is compatible with OpenFlow switches and demonstrates good TCP throughput, it is susceptible to network security attacks.

In their research work, Liyanage et.al. [3] presented a Virtual Private LAN Service (VPLS) framework that is scalable, secure, and consists of two layers. Furthermore, they introduced a SoftVPLS, which is an SDN-based VPLS that addresses tunnel handling issues faced by earlier VPLS models.

Tupakula et.al [4] presented a comprehensive study on the different types of attacks that can be launched on SDN switches. To address this issue, they developed a Switch Security Application (SSA) based on a trusted computing platform, which can identify potential attacks on switches. However, implementing their methodology can be complex and impractical for all devices.

In research conducted by Kenchaiah et.al. [[5], a novel method was proposed to address the issue of trust during communication between switches and SDN controllers that use the OpenFlow protocol. The authors employed the use of the Trusted Platform Module (TPM) and blockchain technologies to establish trust and prevent network threats

Previous research studies have delved into the subject of SDN and SDVPLS, and have provided valuable insights. However, despite these efforts, there are still several major network security threats that could lead to network and resource unavailability [6], man-in-the-middle attacks, cryptographic flaws, and other vulnerabilities. In order to address these research gaps, this paper presents a method that proposes the use of Trusted Platform Module (TPM)-based SDVPLS, which is intended to provide a secure solution that can help mitigate various network attacks.

3. Proposed Framework

OVS is an open-source virtual switch that can be used to implement SDN-based VPLS networks. DAA is a cryptographic protocol used to verify the authenticity of hardware devices without revealing any information about the devices. Here are the general steps to implement an SDVPLS network using OVS and DAA:

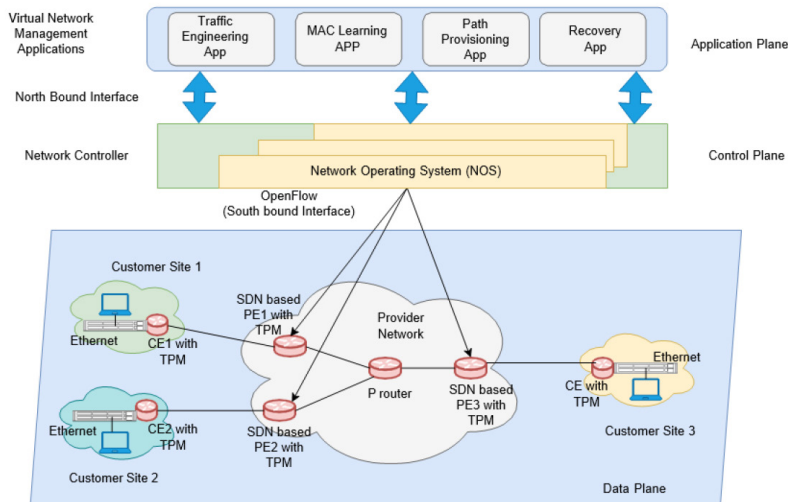


Figure 1

The Proposed SDN-VPLS-TPM Architecture

1. Install OVS: First, install OVS on the devices that will be used in the network, such as servers or switches.
2. Set up the SDN controller: Next, set up an SDN controller to manage the network. There are several open-source SDN controllers available, such as OpenDaylight and ONOS, that can be used with Open vSwitch.
3. Configure the VPLS network: Use the SDN controller to configure the VPLS network, including setting up the VLANs and virtual interfaces for the network.
4. Install and configure DAA: Install and configure DAA on the devices that will be part of the VPLS network. This will involve generating and securely storing the DAA keys on the devices.
5. Authenticate devices using DAA: Use DAA to authenticate the devices in the VPLS network without revealing any identifying information about the devices. This will involve exchanging anonymous tokens between the devices to verify their authenticity.
6. Encrypt VPLS traffic: Finally, use encryption to secure the VPLS traffic. Use the encryption keys stored on the DAA-enabled devices to encrypt and decrypt the traffic as it is transmitted over the network.

Overall, implementing an SDVPLS network using OVS and DAA can provide a high level of security and privacy for the network. However, it is important to follow best practices and security guidelines when implementing DAA to ensure maximum security and protection of sensitive data.

The implementation of TPM in an SDN-based VPLS would typically involve the following steps:

1. Hardware installation: A TPM module would need to be physically installed on the devices used in the SDN network, such as switches or servers.
2. TPM initialization: Once the hardware is installed, the TPM would need to be initialized and configured with the appropriate security settings. This may involve setting up passwords and other security measures to protect the sensitive data stored on the TPM.
3. SDN controller integration: The SDN controller would need to be integrated with the TPM to access the keys and other sensitive data stored on the TPM.

4. Encryption key management: The SDN controller can use the TPM to securely store encryption keys used by the VPLS. This would involve generating and securely storing the keys on the TPM, and then using them for encryption and decryption of the VPLS traffic.
5. Virtual Network Function deployment: If VNFs are being used, they would need to be deployed on the SDN network and configured to use the encryption keys stored on the TPM for secure communication.
6. Testing and verification: Finally, the implementation would need to be tested and verified to ensure that the VPLS is secure, and that the data being transmitted over it is protected.

Figure 1 shows the proposed framework with the use of TPM.

It is important to note that the exact implementation details may vary depending on the specific SDN and VPLS solutions being used, as well as the hardware and software being used to deploy the TPM. Additionally, it's important to follow best practices and security guidelines when implementing TPM in an SDVPLS to ensure maximum security and protection of sensitive data.

The proposed framework is an enhancement of SDN-based VPLS proposed by Liyanage et. al. [7]. In the proposed architecture, legacy Provider Equipment (PE) was replaced with OpenFlow switches and OpenFlow protocol is utilized to install flow rules in each PE. However, there is no change in the underlying provider network equipment other than PEs. Hence, existing Provider Routers and switches can be used as intermediate devices without any modification.

Our framework makes use of the Switch Security Application (SSA) [4]. Let's begin by discussing some of the underlying assumptions of our model.

- The SDN controller in the domain has not been jeopardized.
- We assume that PEs and CEs in the SDVPLS have built in TPM chip.
- The part of the switch responsible for communicating the flow rules is not jeopardized.

The SSA is used for detecting attacks on the switches. SSA runs on the top of the SDN controller for more flexibility. As the controller has global access to the information on all the devices in the network. The information stored at the controller is accessible to SSA, which uses it to track and

identify switch attack vectors. Each Customer Equipment (CE) -PE with TPM will perform platform authentication. The Endorsement Key (EK) is a unique combination of public and private keys that is assigned to each TPM during its manufacturing process. The manufacturer validates this key. To authenticate a platform, we use DAA in our model. This method allows the platform owner to avoid relying on a third party.

4. Implementation

The proposed design was executed on a testbed to assess its performance. To implement PEs and CEs, we utilized three laptops equipped with an Intel i7 3210M CPU that runs at 3.7 GHz. Each of these computers is equipped with an OVS. Each OVS represents PE for each customer site. Here, two CEs are connected to each PE. The SDN controller is implemented using Open Network Operating System (ONOS). The network environment was created using Oracle VM Box [8]. Switch Security is a logical architectural component that runs over a Northbound interface as an ONOS application. For generating malicious attacks, we have used Kali Linux [9].

The trusted report is compared with the switch state report by using the VIM diff tool. DAA is implemented in JAVA using Java Runtime Environment. All protocols and algorithms of DAA were implemented as single threaded program. PE1 and PE2 receive a request for flow rules from the State Validation component in SSA. The State Validation component searches the Store Database for switch PE1 and PE2 and calls the State Generation component to produce a reliable report on the expected state of PE1 and PE2. The State Validation Component makes use of the VIM diff tool to compare reports. It matches the reliable report with the reports from PE1 and PE2.

Now, let's consider a scenario where an attacker tries to manipulate switch PE1 by changing some flow rules. To demonstrate this, we created a flow mod message using a Kali Linux machine and injected an unauthorized flow rule into PE1. The switch is later requested by the Validation Component to report the flow rules. The same procedure as before is followed, but with an additional step where the injected flow rule is also included in the switch state report.

When the switch state report and trustworthy report are compared, the State Validation Component identifies the fraudulent flow rule, triggering the alarm and restore component. In such a situation, the SDN administrator receives an alarm from the alert and repair components. To

ensure secure communication, a direct and anonymous system is implemented through the DAA protocols. The Static Diffie-Hellman oracle is used to join and sign in securely.

Algorithm 1 demonstrates the working of the SDN-VPLS-TPM framework.

Algorithm 1 SDN VPLS TPM Algorithm

- 1: Initialize SDN Controller, devices, VPLS, TPM modules
- 2: EstablishCommunicationLinks(SDN Controller, devices)
- 3: **for** each device in devices **do**
- 4: ConnectDeviceToVPLS(device, VPLS)
- 5: **for** each device in devices **do**
- 6: ConfigureDevice(device)
- 7: IntegrateTPM(devices, TPM modules)
- 8: DefineSecurityPolicies(SDN Controller)
- 9: ManageTraffic(SDN Controller)
- 10: **for** each device in devices **do**
- 11: RegularDeviceAttestation(TPM modules, SDN Controller)
- 12: UseVPLSForEncryption(VPLS)
- 13: ImplementTPMForAuthentication(devices, TPM modules)
- 14: DeployAnomalyDetection(SDN Controller, TPM modules)
- 15: MitigateSecurityThreats(SDN Controller, TPM modules)
- 16: ImplementLoggingAndMonitoring(SDN Controller)
- 17: **for** each device in devices **do**
- 18: PeriodicAuditing(device, TPM modules)
- 19: DevelopIncidentResponsePlan(SDN Controller, TPM modules)
- 20: RegularUpdatesAndPatchManagement(SDN Controller, VPLS, TPM modules)
- 21: MaintainDocumentation()
- 22: ProvideTrainingToAdministrators()
- 23: **for** each device in devices **do**
- 24: ConductRegularTesting(device, TPM modules)
- 25: AssessAndImproveSecurity(SDN Controller)

Further, we have analyzed the traffic flow for SDN-based VPLS with and without TPM. As depicted in Figure 2, the total number of requests sent with the help of ONOS and custom scripts to SDN-based VPLS resulted in decrease of resource availability as the SDN based VPLS network gets overwhelmed with the requests. Overall efficiency was also drastically reduced. On the other hand, the VPLS with TPM has been found effective against the DDoS requests, as shown in Figure 2.

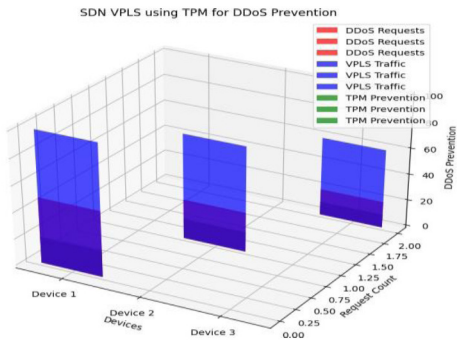


Figure 2
Prevention of Network Attacks with TPM

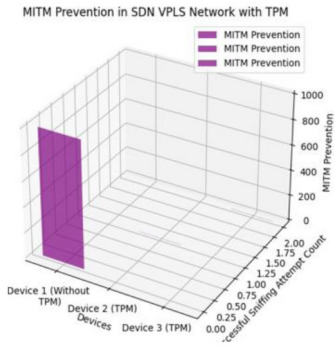


Figure 3
Comparison of Vulnerability Probability and Efficiency

Moreover, the architecture of VPLS with TPM is found to be non-vulnerable to MITM attacks as TPM provides enhanced encryption security. Figure 3 demonstrates this scenario. As shown in the figure, the SDN-based VPLS network device data can be easily intercepted, and the attacker can read the data as it is not encrypted. Whereas the data from

other devices with TPM enabled is in encrypted form with hardware-level encryption. Even if the sniffer succeeds in intercepting the requests, it may not be able to easily decrypt the data.

5. Comparison between SDN Based VPLS and SDN Based VPLS with TPM

SDN-based VPLS and SDN-based VPLS with TPM are both network architectures that leverage SDN to extend a single Ethernet broadcast domain over a wide area network. However, there are some key differences between the two architectures, as outlined below:

5.1 Security

The main difference between SDN-based VPLS and SDN-based VPLS with TPM is the level of security. While SDN-based VPLS provides basic security features such as VLAN segmentation and virtual interfaces, SDN-based VPLS with TPM offers additional security through the use of a hardware security module to store cryptographic keys and other sensitive data. Moreover, it protects the network against DDoS and ransomware attacks, as shown in Figure 4.

Figure 5 shows the comparison between SDVPLS and SDN-VPLS-TPM based on the vulnerability probability against network security attacks and efficiency. It is clear from the graph that SDVPLS network devices have a high probability of being vulnerable to various security attacks compared to devices with TPM enabled and configured. Subsequently, the efficiency of SDVPLS devices is lower in terms of security compared to SDN-VPLS-TPM.

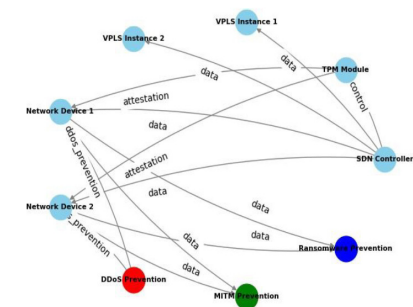


Figure 4
Prevention of Network Attack



Figure 5
Vulnerability Comparison

5.2 Complexity

Implementing SDVPLS with TPM can be more complex than implementing SDVPLS alone. This is because TPM requires additional hardware and software components to be installed and configured, and it may require additional expertise to implement.

5.3 Cost

The cost of implementing SDVPLS with TPM is typically higher than that of implementing SDVPLS alone. This is because TPM hardware and software components can be expensive and additional expertise may be required to implement the solution.

5.4 Scalability

Both SDN-based VPLS and SDN-based VPLS with TPM can be scaled to support large networks. However, SDN-based VPLS with TPM may be more challenging to scale due to the additional hardware and software components required.

In summary, SDN-based VPLS with TPM offers additional security features compared to SDN-based VPLS alone, but it can be more complex and expensive to implement. The choice between the two architectures depends on the specific needs of the organization, including the level of security required, the complexity of the network, and the available budget. Table 1 shows a comparative summary of the proposed method with similar state-of-the-art techniques based on protection against various security attacks, complexity, cost, scalability, and reliability.

6. Conclusion and Future Work

In the field of network security, Software-Defined Networking (SDN) and Virtual Private LAN Service (VPLS) augmented by Trusted Platform Module (TPM) constitute a powerful collaboration. TPM enhances SDN VPLS security with hardware-based protection, safeguarding cryptographic keys and ensuring secure boot methods to minimize the risks of unauthorized access and data breaches. The proposed method prevents SDN-based VPLS against various security attacks such as MITM, DDoS, ransomware, and so on. The probability of an SDN network being vulnerable to these attacks is greater than 0.8, which indicates that the role of TPM is crucial in protection against these attacks.

Table 1

Comparison of state-of-the-art methodologies having extended scope with this article.(✓= Article covers the content, ★= Article covers limited content, ×= Article covers negligible content)

Methodology /Technique	Protection against Security Attacks				Complexity	Cost	Scalability	Reliability
	Ransomware	Spoofing	MITM	DDoS				
SDxVPNMirkhanzadeh et al. (2016 [1]	×	×	×	×	★	★	×	×
L2OVXHu et al. (2016)[[2]	×	×	×	×	★	★	★	★
SoftVPLS Liyanage and Gurtov (2013)[[3]	★	★	×	×	★	★	✓	★
Switch Security Application (SSA) Tupakula et al. (2020)[4]	★	★	✓	✓	✓	✓	✓	★
SDN-VPLS-TPM [This Article]	✓	✓	✓	✓	✓	✓	✓	✓

It is clear from the table that the proposed study covers the above-mentioned attributes and is more secure against network security attacks. The integration of TPM into the SDVPLS architecture is considered to be a crucial step towards creating a highly secure, dependable, and resilient network infrastructure. By incorporating TPM's hardware-based security methods, the SDVPLS architecture can effectively mitigate various network security threats, safeguard critical assets, and ensure uninterrupted and reliable network operations.

References

- [1] B. Mirkhanzadeh, N. Taheri, and S. Khorsandi, "SDxVPN: A software-defined solution for VPN service providers," in NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium, pp. 180–188 (2016). doi: 10.1109/NOMS.2016.7502811.
- [2] J.-W. Hu, C.-S. Yang, and T.-L. Liu, "L2OVX: An On-demand VPLS Service with Software-Defined Networks," in 2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 861–866 (2016). doi: 10.1109/WAINA.2016.9.

- [3] M. Liyanage and A. Gurtov, "A scalable and secure VPLS architecture for provider provisioned networks," in 2013 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1115–1120 (2013). doi: 10.1109/WCNC.2013.6554720.
- [4] U. Tupakula, V. Varadharajan, and K. K. Karmakar, "Attack Detection on the Software Defined Networking Switches," in 2020 6th IEEE Conference on Network Softwarization (NetSoft), pp. 262–266 (2020). doi: 10.1109/NetSoft48620.2020.9165459.
- [5] N. and others Kenchaiah, "Trustworthiness Among Controllers and Switches In Multi-Provider Software Defined Network Deployments Using A Trusted Platform Module (Tpm) And Secure Ledger," (2020).
- [6] S. Chattopadhyay, A. K. Sahoo, and S. Jasola, "Improvement in DDoS attack detection in software defined network using ML algorithm," *Journal of Discrete Mathematical Sciences & Cryptography*, vol. 26, no. 7, pp. 2025–2044 (2023), doi: 10.47974/JDMSC-1848.
- [7] M. Liyanage, M. Ylianttila, and A. Gurtov, "Improving the tunnel management performance of secure VPLS architectures with SDN," in 2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC), pp. 530–536 (2016). doi: 10.1109/CCNC.2016.7444836.
- [8] Oracle VM Box. Available at <https://www.virtualbox.org/>. Accessed on 20 Feb 2024.
- [9] K. Linux, "Kali Linux," Obtenido de Official Kali Linux Documentation: <http://docs.kali.org> (2020).