Secure Hierarchical Virtual Private LAN Services for Provider Provisioned Networks

Madhusanka Liyanage*, Mika Ylianttila*, Andrei Gurtov†
*Centre for Wireless Communication, University of Oulu, Finland.
{madhusanka, mika.ylianttila}@ee.oulu.fi
† Department of Computer Science and Engineering, Aalto University, Finland.
{gurtov}@hiit.fi

***Abstract*—Virtual Private LAN Service (VPLS) is a widely used Layer 2 (L2) Virtual Private Network (VPN) service. Initially, VPLS architectures were proposed as flat architectures. They were used only for small and medium scale networks due to the lack of scalability. Hierarchical VPLS architectures are proposed to overcome these scalability issues. On the other hand, the security is an indispensable factor of a VPLS since it delivers the private user frames via an untrusted public network. However, the existing hierarchical architectures unable to provide a sufficient level of security for a VPLS network.**

In this paper, we propose a novel hierarchical VPLS architecture based on Host Identity Protocol (HIP). It provides a secure VPLS network by delivering vital security features such as authentication, confidentiality, integrity, availability, secure control protocol and robustness to the known attacks. The simulations verify that our proposal provides the control, forwarding and security plane scalability by reducing the number of tunnels in the network as well as the number of keys stored at a node and the network. Finally, the simulation results confirm that the control protocol of the proposed architecture is protected from IP based attacks.

***Index Terms*—Virtual Private Networks, Virtual Private LAN Service, Security, Host Identity Protocol**

I. Introduction

Provider provisioned VPN services are popular among the enterprise customers as they interconnect the geographical distributed customer's private network via the provider network. In this paper, we particularly focus on VPLS which is one of the most popular provider provisioned Layer 2 VPN (L2VPN) service.

Internet Engineering Task Force (IETF) defined two standard frameworks to develop a VPLS network by using Border Gateway Protocol (BGP) [1] and Label Distribution Protocol (LDP) [2]. Initially, most of the VPLS architectures are proposed as flat architectures [1] [2] [3] [4] [5]. They are capable enough to build a functional VPLS over small and medium scale provider networks. However, the existing flat VPLS architectures are inefficient to deploy in large scale networks which need to facilitate thousands of nodes. For instance, mobile networks and distributed data centers over the Internet. Mainly, these flat architectures are lacking of control plane scalability due to the excessive number of tunnel establishments.

The first functional Hierarchical VPLS (H-VPLS) architecture is proposed in [2]. Some other research studies also focused on enhancing the features of H-VPLS networks [6] [7] [8] [9] [10] [11]. A L2VPN architecture that provides point-to-point and point-to-multipoint layer 2 data communication services by using a hierarchical LAN switching architecture was presented in [7]. It achieved the scalability and manageability by adding the cost of functionality to the forwarding plane to simplify the control plane. In [6], authors proposed a H-VPLS architecture by using a hub and spoke connectivity model to reduce the signaling and replication overhead. An enhanced H-VPLS architecture by using a control word technique was presented in [8]. A protection scheme for H-VPLS network was proposed in [11]. On the other hand, IETF specified the general requirements of a VPLS in [12]. Security is considered as an indispensable factor of a VPLS since it delivers customer private frames via an untrusted public network [12]. However, these existing hierarchical VPLS architectures are still unable to provide the demanded level of security.

In-fact, a limited number of secure architectures are proposed even for flat VPLS networks. HIP-enabled virtual private LAN Service (HIPLS) was proposed a use-case of HIP to provide a secure VPLS over an untrusted network [4]. However, HIPLS is lacking of scalability in all three planes, namely, control, forwarding and security. HIPLS is suitable only for unicast-only IPLS (IP-only Layer Services) [3] networks. A Session key based HIP VPLS architecture (S-HIPLS) was proposed in [5] [13]. Authors proposed a customized version of HIP with a session key based security mechanism. S-HIPLS provides forwarding and security plane scalability for HIPLS architecture. However, S-HIPLS is still lacking of control plane scalability due to the requirement to establish a massive number of tunnels. On the other hand, both HIPLS and S-HIPLS architectures are able to provide the demanded level of security only for a flat VPLS network.

- Our Contribution

In this paper, we propose a novel hierarchical VPLS architecture based on HIP. Hence, we name it as Hierarchical HIP enabled virtual private LAN Service (H-HIPLS). Our architecture offers a secure VPLS architecture by providing vital security features such as authentication, confidentiality, integrity, availability, secure control protocol and robustness to the known attacks. To the best of our knowledge, this is the first proposal of a secure hierarchical VPLS architecture.

Furthermore, H-HIPLS provides same level of control and forwarding plane scalability as other non-secure hierarchical VPLS architectures. Also, it offers the same level of security plane scalability as other secure flat VPLS architecture.

The rest of the paper is organized as follows. Section II briefly describes the hierarchical VPLS and its security threats. The proposed H-HIPLS architecture is presented in Section III. The simulation model and the numerical results are illustrated in Section IV. Section V and VI respectively contain the discussion and the conclusion of the research.

II. Background

A. Virtual Private LAN Service (VPLS)

VPLS is a layer 2 VPN service. It provides multipoint-to-multipoint connectivity to extend the Ethernet broadcast domain over geographically dispersed sites. VPLS services are becoming an interesting choice among the enterprise customers since they offer high speed connectivity, any-to-any forwarding at layer 2 and support many enterprise applications.

There are three main elements in a VPLS network, i.e. Provider Edge Equipments (PEs), Customer Edge Equipments (CEs), and the provider network. The CE is the intermediate device to interconnect the customer network to the provider network. It can be a router or a switch which is located at the customer's premise. All VPN operations such as tunnels establishment, address learning occur at the PE. They belong to the service providers. The control protocol of the VPLS is used to establish and maintain VPN tunnels between PEs. The provider network is the core/underlay network of VPLS. It operates as a layer 3 network by using common network protocols such as MPLS (Multiprotocol Label Switching) or IP.

1) Issues of a Flat VPLS: Initially, VPLS architectures were used only for small and medium scale provider networks. Few years later, service providers found several scalability issues when they try to deploy these flat VPLS architectures in large scale networks such as Internet and telecommunication networks.

A flat VPLS architecture requires a pseudowire/VPN tunnel between each and every pair of PEs. O(N*(N-1)/2) pseudowires must be set up for each VPLS network where N is the number of PEs in the provider network. This is called as N-square scalability problem and it causes several scalability issues [9].

Firstly, a flat VPLS suffers from a massive signaling overhead which is required to establish and maintain these VPN tunnels. It reduces the scalability of the control plane.

Secondly, each PE has a maximum limit to support the hardware ingress replications and the simultaneous tunnels (e.g. IPsec, HIP and MPLS). If a PE does not able to support N times hardware ingress replications, then a broadcast/multicast frame needs to be sent N times over the same network link. It will consume N times the expected bandwidth. Furthermore, the Nth frame has an additional delay of (N-1)*MTU*8*BW where MTU is the Maximum Transmission Unit and BW is the bandwidth of the link. It drops the scalability of the forwarding plane.

Thirdly, the forwarding mechanism of frames is also complicated. Every PE should have a global knowledge about the VPLS to forward the traffic through the provider network. Hence, PEs have to facilitate huge forwarding tables and run extensive searching mechanism to find the correct destination address.

Fourthly, the service provisioning is difficult in a flat architecture. When the provider needs to interconnect a new customer site by using a new PE, he has to update all other PEs. Then, every other PE needs to establish a tunnel with the new PE.

B. Hierarchical Virtual Private LAN Service (H-VPLS)

H-VPLS is the straight-through mechanism to resolve these scalability issues of flat VPLS networks. Basically, H-VPLS reduces the number of PEs which are connected in the full mesh topology. Therefore, it requires less number of pseudowires than a flat VPLS architecture.

Figure 1 illustrates a simple H-VPLS architecture.

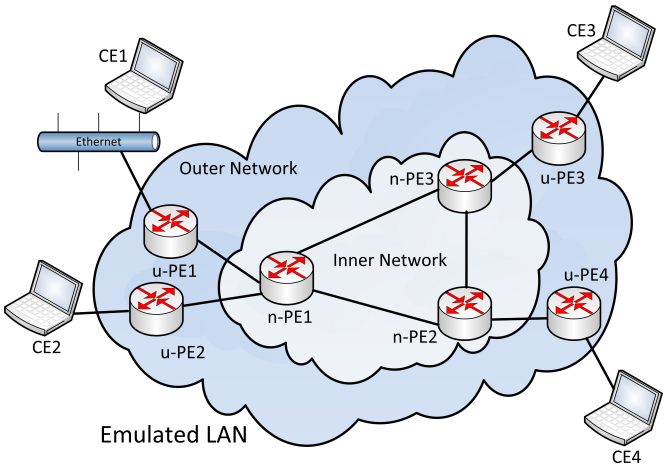

Fig. 1: A simple H-VPLS architecture

H-VPLS utilizes two types of PEs as u-PE and n-PE [10]. u-PEs are user facing PEs, while n-PEs are network facing PEs. A n-PE plays a key role in the VPLS as it has all the intelligence of the VPLS architecture. Specifically, it is responsible for packet forwarding, address learning and auto discovery functions. An u-PE has an aggregation role and it forwards all the packets to the connected n-PE.

On the other hand, H-VPLS not only solves the above scalability issues but also provide additional advantages. As the workload is asymmetric on PEs, it is possible to expand the VPLS network by using simple and cheap equipments as u-PEs. Furthermore, H-VPLS can be deployed over heterogeneous service provider networks due to the technology independence of different layers in the VPLS network [2].

C. Security Issues of the H-VPLS

A H-VPLS network faces a number of security threats. They can jeopardize network resources such as memory space, forwarding tables, network bandwidth, and CPU processing power in PEs.

The most of the H-VPLS architectures use TCP (Transport Control Protocol) based control protocols (e.g. LDP and BGP use TCP sessions) which are vulnerable to several attacks such as DoS (Denial of Service), TCP reset and spoofing attacks.

The data encryption is mandatory for both control and data traffic on a VPLS. If VPLS packets are transmitted in the unencrypted form, then an attacker can eavesdrop these traffic to extract the important information or may be able to alter the data packets in flight. It may cause to interrupt the connectivity or alter the quality of service.

Furthermore, the data frames should be delivered only via authorized PEs. Hence, a mutual authentication of PEs is required before the exchange of data. Otherwise, the user traffic may direct to a wrong location and a malicious user can retrieve or destroy the valuable data.

The privacy protection of both PE and CE is also a mandatory requirement. If the privacy of PEs is not protected, then an attacker can easily target the key elements and nodes (e.g. servers, gateways, databases) in the network. If the privacy of CEs is not protected, eavesdroppers can learn about the important internal devices in the customer network.

Moreover, a PE should be able to process the multicast and broadcast traffic efficiently. Otherwise, it is possible to launch a DoS attack by using fake multicast or broadcast frames.

III. The Proposed VPLS Architecture

In this paper, we propose a secure H-VPLS architecture based on HIP to provide the demanded level of security. HIP is an emerging key negotiation and mobility protocol that enables host mobility and multihoming across different address families (IPv4 and IPv6). Furthermore, HIP provides various security features such as the end-to-end encryption, mutual authentication, secure key exchange and privacy protection [14]. The proposed architecture modifies the HIP based session key mechanism which is proposed in S-HIPLS [5] [13] to facilitate the hierarchical architecture. In addition, H-HIPLS proposes a novel encrypted label based forwarding mechanism and dynamic address learning mechanism.

The operation of H-HIPLS architecture can be categorized in to five main sections as control protocol, key management, PE management, packet forwarding and address learning.

A. Control Protocol

The control protocol is responsible for the tunnel establishment, address learning and key management functions. The secure operation of the control protocol is mandatory to maintain the proper operation of the VPLS network. Our H-HIPLS proposes a control protocol based on secure HIP signaling. We define a control VPN for the VPLS network. All control frames are encrypted by using the session key of this control VPN.

B. Key Management

Our architecture uses two key types as Content Encryption Key (CEK) and Key Encryption Key (KEK). The CEK uses to encrypt all data frames in a single VPN and the KEK uses to encrypt/decrypt the corresponding CEKs, certificates and any other control information. The CEK is unique to each VPN and the KEK is unique to each PE. There is a Key Distribution Center (KDC) which is responsible to distribute CEKs to corresponding PEs and maintains the Access Control Lists (ACLs).

Every PE needs to be authorized from the KDC before joining the VPLS. During this joining process, each PE shares an unique KEK with the KDC based on D-H (Diffie-Hellman) key exchange. On the other hand, the KDC will periodically generate the session keys (CEKs) and securely distribute to each PE.

C. PE Management

The initial procedure for a potential PE is to be authenticated to access the VPLS. The KDC authenticates these new PEs. We propose a novel authentication mechanism based on HIP. During this authentication phase, the potential PE establishes a HIP tunnel with the KDC. A HIP tunnel is an IPsec BEET (Bounded End-to-End Tunnel) mode tunnel based on ESP (Encapsulating Security Payload) protocol.

According to the HIP specifications, the HIP BEX (Base Exchange) is used to establish these tunnels [15]. The HIP BEX is the initial handshake procedure which establishes Security Associations (SAs) for HIP tunnels and mutually authenticates the end nodes based on cryptographic identities.

We propose a novel authentication mechanism based on this HIP BEX. In contrast to the original HIP BEX, the proposed authentication mechanism not only authenticates the potential PE by using a Public Key Infrastructure (PKI) mechanism but also authorizes the potential user according to ACLs (Access Control Lists).

Figure 2 illustrates the novel authentication procedure based on the HIP BEX. Here, the initiator and responder respectively represent the unregistered PE and the KDC. Furthermore, we use the same terminology which was used for HIP BEX in [16].

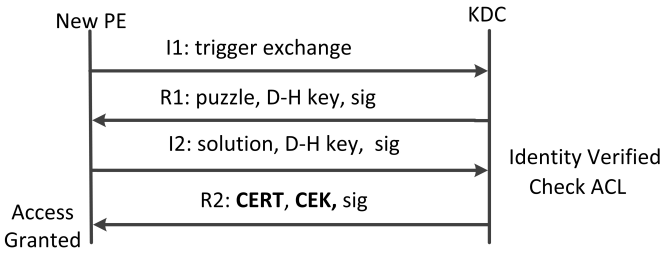

Fig. 2: The proposed authentication mechanism

We use the original HIP BEX [14] message formats for first three messages. After the arrival of the I2 message, the KDC verifies the identity of the new user and checks the

ACL to authorize the user. Then, the KDC sends a certificate and the CEK of the control VPN. The certificate contains an authorization token, configuration information for the PE and other VPN management data. It is encrypted by the KEK to protect the integrity and the confidentiality. The authorization token is mandatory to establish the HIP tunnels between the PEs for packet forwarding.

On the other hand, it is necessary to remove inactive PEs for the efficient operation of the VPLS. H-HIPLS uses both active and passive notifications. In an active notification mechanism, PEs will actively notify their departure to the KDC before they leave the VPLS. In passive notification mechanism, the KDC learns the departure of a PE by a failure to acknowledge a periodic CEK distribution.

D. Packet Forwarding

We propose a novel encrypted label based packet forwarding mechanism and this section describes the proposal.

When a u-PE receives a data frame from a CE, it follows three steps. Since H-HIPLS proposes to exchange the data only through the HIP tunnels, the u-PE checks for an existing HIP tunnel between the n-PE as the first step. If there is no HIP tunnel, the u-PE establishes a HIP tunnel between the n-PE. We propose a novel tunnel establishment mechanism based on the HIP BEX to establish these HIP tunnels for the packet forwarding function. Figure 3 illustrates the proposed tunnel establishment procedure.

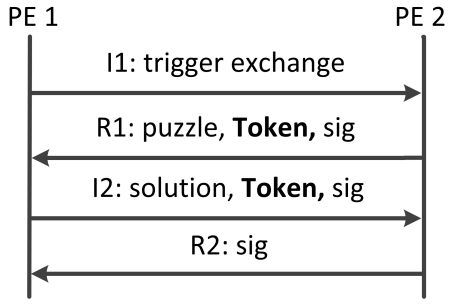

Fig. 3: The proposed tunnel establishment procedure

The message exchanges of the proposed tunnel establishment procedure is almost similar to the message exchanges of the previous authentication procedures. Furthermore, the functions of the obligatory fields in both procedures are same. However, there are two notable deferences. First, the tunnel establishment procedure evades the D-H key exchange for faster tunnel establishment than the original HIP BEX. Therefore, R1 and I2 messages do not contain any D-H key exchange fields. Second, we propose to exchange an authentication token during this tunnel establishment procedure. It prevents the tunnel establishments with the unauthorized users. Hence, I2 and R1 messages contain the authentication token which is provided by the KDC. It ensures that only preauthorize users are able to build the HIP tunnels over the provider network.

In the second step, the source u-PE encrypts the Layer 2 (L2) frame using the corresponding CEK of the VPN. Then it will wrap within the ESP payload.

In the third step, the source u-PE inserts the encrypted label into the standard ESP header of the packet and forwards the frame to the n-PE. Figure 4 illustrates a modified ESP header. The encrypted label is the encrypted destination MAC (Media Access Control) address of the frame. It encrypts by using the session key of the control VPN.

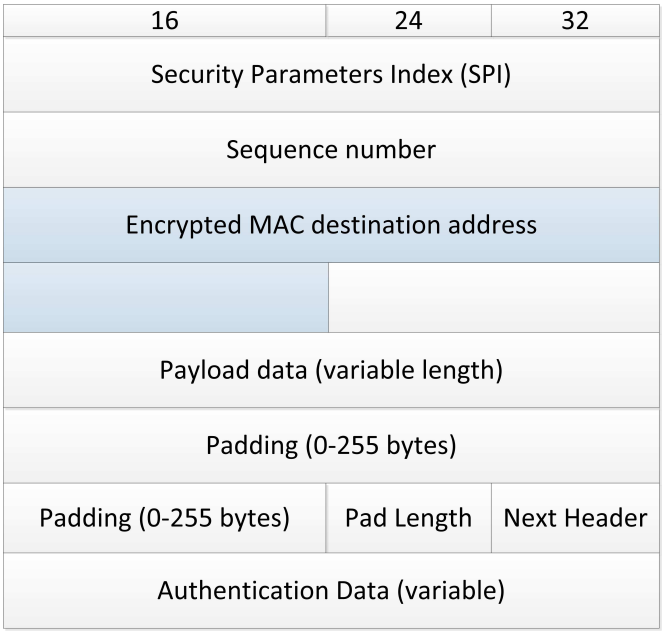

Fig. 4: The Modified ESP Header

When an u-PE receives a data frame from a n-PE, the u-PE removes upper layer headers including the ESP header and decrypts the ESP payload by using the corresponding session key of the VPN. Then it places on the customer access network as a layer 2 frame.

When a n-PE receives a data frame, it follows two steps. First, it decrypts the encrypted label and checks the MAC-PE mapping table for the next hop to forward the packet. The MAC-PE mapping table is the forwarding table of the VPLS which uses to map the destination address of the MAC to the next hop PE. Second, it checks for an existing HIP tunnel between the next PE. If there is no HIP tunnel, it establishes a new tunnel. Then, it forwards the frame to the next PE.

E. Address Learning

Since VPLS is a layer 2 VPN solution, PEs forward the frames based on MAC addresses. On the other hand, each u-PE is responsible for a certain set of CEs in the VPLS network. Hence, frames should be delivered over the provider network to reach the correct PE which is responsible for the destination CE. However, the underlay provider network is a layer 3 network. Therefore, it is needed to map the destination MAC address of the CE to the network address of the corresponding

PE. We propose to maintain a dynamic MAC-PE mapping table in each n-PE to accomplice this requirement.

Each n-PE updates their MAC-PE mapping table by using two address learning instances, namely the u-PE advertisements and the dynamic address requests. In the first case, each u-PE advertises the MAC addresses of the responsible CEs to directly connected n-PE. These advertisements contain the MAC addresses of newly added CEs and inactive CEs. The n-PE updates its MAC-PE table according to the advertisement.

In the second case, a n-PE dynamically fills the MAC-PE table. When a n-PE receives a frame with an unknown destination MAC, the n-PE broadcasts an encrypted address request frame (a.k.a. Dynamic Address Request) to all the other PEs to identify the responsible PE. This dynamic address request frame is encrypted by using the session key of the control VPN. Then, the responsible PE will send an unicast frame as a reply and the PE updates its MAC-PE mapping table.

IV. Numerical Results

We simulate the proposed H-HIPLS architecture on MATLab and evaluate the performance. Since there are no secure hierarchical VPLS architectures, we compare the performance of our proposal with both secure flat VPLS architectures namely HIPLS [4], S-HIPLS [5] and the most popular non secure H-VPLS architectures such as LDP based H-VPLS architecture (H-LDP) [2].

A. Comparison of Tunnel Establishment Requirement

A H-VPLS architecture reduce of the number of tunnels/sessions to achieve the control plane scalability which is the foremost advantage of it. Hence, we compare the tunnel requirement for each architecture.

1) Total Tunnel Requirement in the Network: The number of tunnels in a flat VPLS architecture is only depending on the number of PEs in the network. However, the number of tunnels in a H-VPLS architecture is depending on both the number of PEs in the network and u-PE to n-PE ratio.

Hence, we assume that the maximum number of u-PEs that can be connected for a single n-PE is 10 for the rest of our experiments. Figure 5 illustrates the total tunnel establishment complexity of the VPLS network against the number of PEs.

We observe a significant reduction of the total number of tunnels in hierarchical architectures compared with flat architectures. There is a linear increment of the total number of tunnels with the number of PEs in the network for both H-LDP and proposed H-HIPLS architectures. Comparably, the H-LDP has slightly better performance than the proposed H-HIPLS as H-HIPLS needs an extra tunnel per PE between the KDC for the secure key exchange.

On the other hand, the total number of tunnels requirement in the network is exponentially increased with the number of PEs for both HIPLS and S-HIPLS architectures.

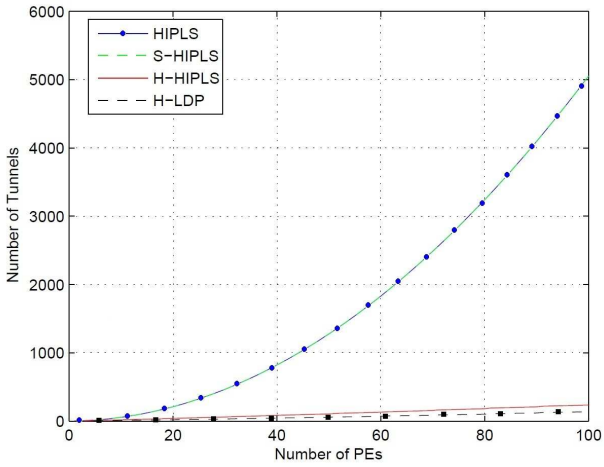

Fig. 5: The total number of tunnel in the Network

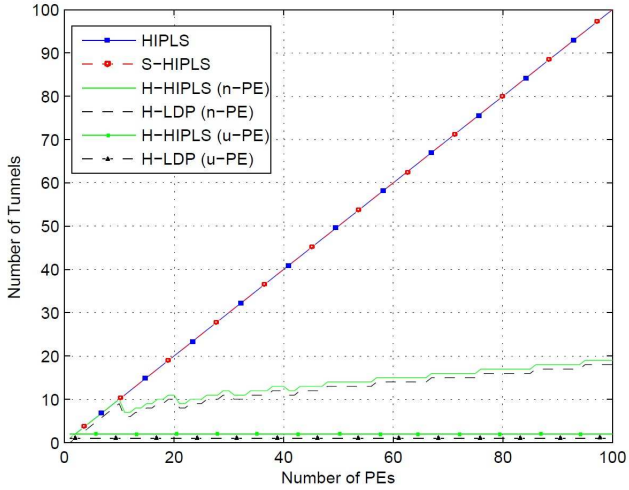

Fig. 6: The total number of tunnel per PE

2) Total Tunnel Requirement per PE: We illustrate the total tunnel establishment complexity of a PE against the number of PEs in Figure 6. Here we change the PEs from 1 to 100.

We observe a significant reduction in the number of tunnels per PE in hierarchical architectures compared with flat architectures. There is a staircase-like linear increment of the number of tunnels per n-PE with the number of PEs for both H-LDP and proposed H-HIPLS architectures. Furthermore, the number of tunnels per u-PE remains constant for both H-LDP and H-HIPLS architectures as it is independent of the number of PEs. Comparably, the H-LDP has slightly a better performance than the proposed H-HIPLS since each PE in a H-HIPLS needs an extra tunnel for the secure key exchange. On the other hand, the number of tunnels per PE is linearly increasing with the number of PEs for both HIPLS and S-HIPLS architectures.

Therefore, the experiment results clearly show that the tunnel establishment complexity in the proposed H-HIPLS scheme is significantly reduced compared with the other

secured architectures i.e. HIPLS and S-HIPLS. Furthermore, it offers almost similar performance as other hierarchical architectures such as H-LDP. Hence, the H-HIPLS improves the scalability of control plane for secure VPLS architectures and provides the similar performance as the existing hierarchical VPLS architectures.

B. Comparison of Key Storage Requirement

The key storage requirement is expressing the security plane scalability. If a PE needs to store a large number of keys, it uses the already scarce memory space of a PE which can use for other functions such as forwarding tables, filters and frame buffering. On the other hand, the large number of keys cause to extensive key searches which use extra processing power and increase the encryption delay. We evaluate the key storage requirement at different entities of the VPLS network for HIPLS, S-HIPLS and our H-HIPLS architectures to study the security plane scalability.

1) Key Storage at a PE: We illustrate the key storage complexity at a PE against the number of PEs in Figure 7. Here we set the number of VPNs as 5 [5] [17] and change the PEs from 1 to 100.

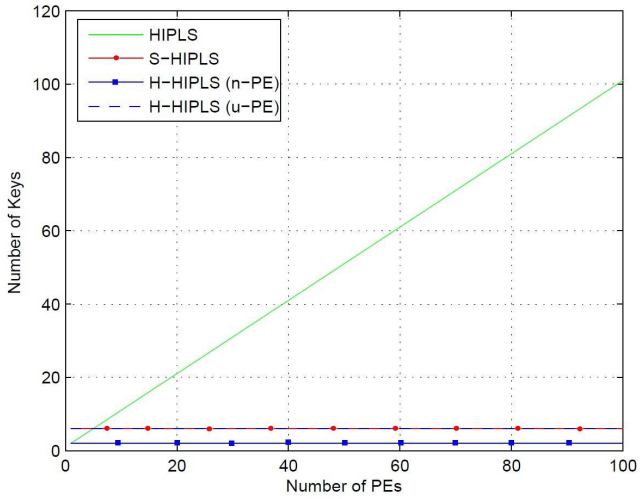

Fig. 7: The number of keys stored at a PE

We observe a linear increment of the total number of key storef at a PE with the number of PEs for HIPLS architecture. Both S-HIPLS and H-HIPLS architectures (only for u-PE) have similar performance and the number of keys stored at a PE remains constant. Hence, the number of keys stored at a PE is independent of the number of PEs for both S-HIPLS and H-HIPLS architectures (only for u-PE). Furthermore, the n-PE of H-HIPLS has minimum key storage requirement as it stores the CEK of control VPN and its own KEK.

2) Key Storage in the Authentication Server (AS)/Key Distribution Center (KDC): We illustrate the key storage complexity in AS/KDC against the number of PEs in Figure 8. Here we fix the number of VPNs as 5 and change the PEs from 1 to 100.

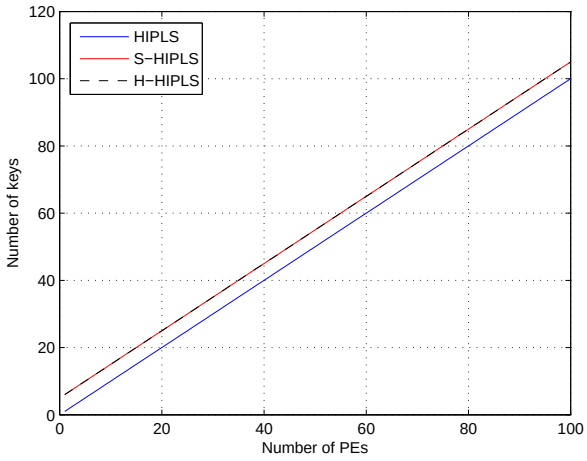

Fig. 8: The number of keys stored at the AS/KDC

We observe a linear increment of the total number of keys stored at AS/KDC with the number of PEs for all three scenarios. However, the number of keys stored at a KDC in H-HIPLS and S-HIPLS are slightly higher than HIPLS. The number of keys stored at AS for HIPLS only depends on the number of PEs in the network. However, the number of keys stored at the KDC for S-HIPLS and H-HIPLS architectures depend on both the number of PEs and the VPNs in the network.

This weakness is less significant due the limited number of VPNs. In a provider network, the customer VPNs are categorized in to VPN classes based on the service level agreements. Then, the provider considers all the VPNs in a single class as a single VPN. Hence, the number of VPNs in a provider network is limited [5] [17]. On the other hand, a distributed KDC system can also use to reduce effect of this weakness.

3) Total Key Storage in the Network: We illustrate the total key storage complexity of the VPLS network against the number of PEs in Figure 9. We fix the number of VPNs as 5 and change the PEs from 1 to 100.

We observe an exponential increment of the total number of key store in the network with the number of PEs for HIPLS architecture. S-HIPLS and H-HIPLS architectures have almost similar performance and the number of keys store in the network is linearly increased with the number of PEs in the network.

The experiment results clearly show that the key storage requirement in the proposed H-HIPLS is significantly lower than HIPLS and slightly lower than S-HIPLS. Hence, it provides a better security plane scalability than any other secure VPLS architecture.

C. Comparison of the Broadcast Mechanism

An efficient broadcast mechanism is the key requirement for the scalability of forwarding plane . Hence, the performance of the frame broadcasting mechanism in different architectures

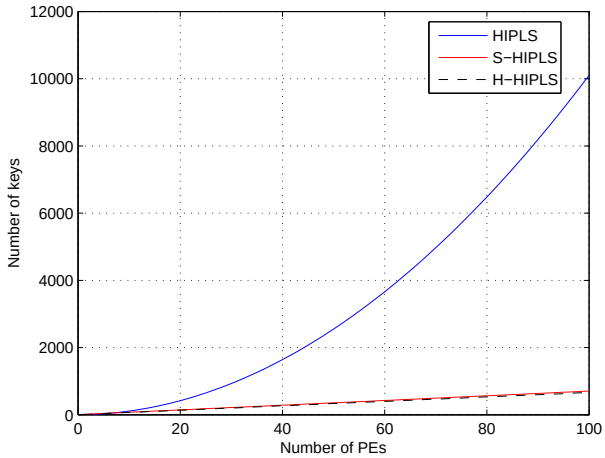

Fig. 9: The total number of keys stored in the VPLS network

has been compared.

Figure 10 illustrates the number of encryption per broadcast frame at the entry PE of the provider network.

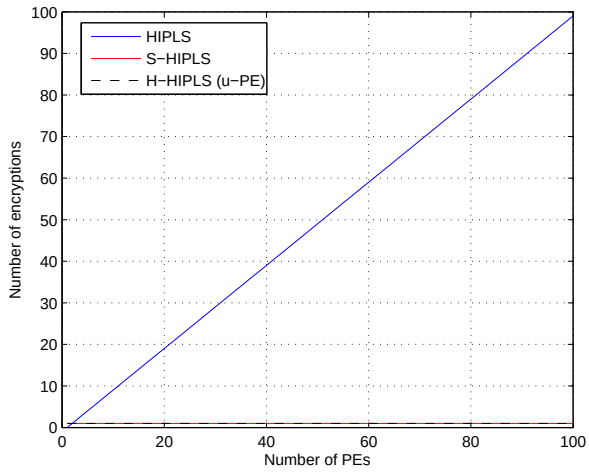

Fig. 10: The number of encryption per broadcast frame

We can see a linear increment of the number of encryptions per broadcast frame at a PE with the number of PEs in the network. However, the number of encryptions of both S-HIPLS and H-HIPLS remains constant at 1.

Therefore, the proposed scheme offers an efficient broadcast mechanism and provides the forwarding plane scalability similar to S-HIPLS and H-LDP.

D. Secure Control Protocol

The control protocol uses to establish and maintain tunnels to ensure a smooth operation of the VPLS. Hence, it should be protected from external attacks. However, the control protocol of most of the existing H-VPLS architectures are vulnerable to IP based attacks such as TCP SYN (Synchronization) DoS (Denial of Service), TCP SYN DDoS (Distributed DoS) and TCP reset [5]. We compare the impact of IP based attacks on the control protocol of the proposed architecture. We use H-LDP [2], HIPLS [4] and S-HIPLS [5] as the reference models to compare the performance under TCP SYN DoS and TCP reset attacks.

1) The Impact of TCP SYN DoS Attack: In a TCP SYN DoS attack, an attacker sends an excessive amount of TCP SYN packets to the target server. Since a server allocates a TCP port for each successfully arrived TCP SYN packet and reserves it for a certain time period (TCP timeout), so that the attacker ables to capture all ports in the server [18]. Then, the server is not responding for the legitimate user traffic.

We use the same simulation model which was presented in [5]. It has a VPLS network with 300 nodes and the bandwidth of the network is 100 Mbps. The attacker also has the same bandwidth of 100 Mbps. The number of TCP ports per user is set to 64000 and the TCP timeout is set to 270 s [18]. The simulation runs for 500 s and the attacker sends the fake TCP SYN packets during 25 s - 75 s time interval for a single user. We measure the packet drop at the user node and Figure 11 illustrates the percentage packet drop over the simulation time.

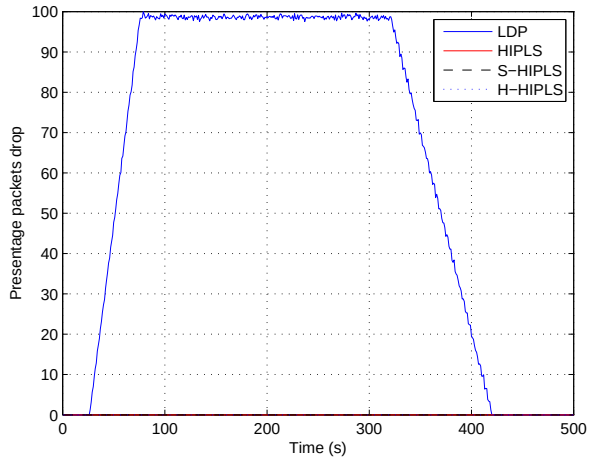

Fig. 11: The impact of TCP SYN DoS attack

HIPLS, S-HIPLS and the proposed H-HIPLS have similar performance under the TCP SYN DoS attack. These architectures have almost zero packet drop for the whole simulation period. However, the H-LDP lost almost all the packets during the attack period. Although the attack is taken place for 50 s, H-LDP architecture requires at least 270 s (a TCP timeout period) to fully recover from the attack. The simulation results verify that the control protocol of the proposed H-HIPLS is secured from TCP SYN DoS attacks.

2) The Impact of TCP Reset Attack: A TCP reset attack can terminate an ongoing TCP connection between two users by injecting the fake TCP packets to the network. An attacker eavesdrops the TCP connection and collect the TCP header information. Later, this information is used to generate fake TCP packets. The attacker set the “Reset Bit” to “1” in these TCP packets. Usually, the “Reset Bit” is used to indicate

unexpected failures on either side of the TCP connection and request to reset the connection. Since a typical end user does not have a mechanism to identify these fake TCP packets, end users terminate the TCP connection upon the arrival of these fake TCP packets [19].

We evaluate the impact of TCP rest attack on the proposed H-HIPLS architecture and compare the performance with H-LDP, HIPLS and S-HIPLS. We use the same simulation setup which is used to evaluate the impact of TCP SYN DoS attack. Figure 12 illustrates the probability of a successful attack against the size of the file. We change the size of files according to the Pareto distribution with the minimum file size of 4.5 KBytes and to the maximum size of 20 MBytes [20].

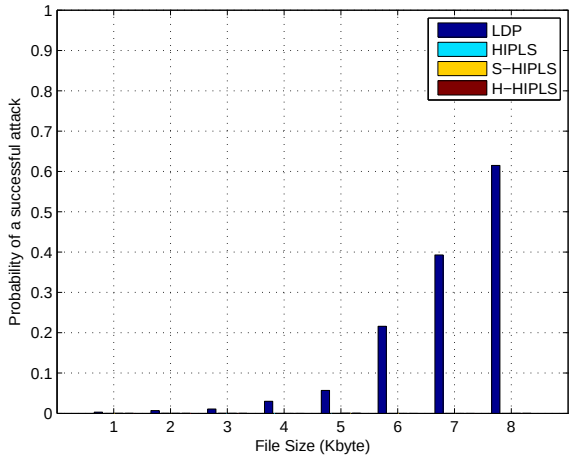

Fig. 12: The impact of TCP reset attack

We observe that the probability to successfully attack the H-LDP architecture is increasing with the file size. The attacker gets more time to reset the connection due to the longer transmission delay of larger files. On the other hand, HIPLS, S-HIPLS and the proposed H-HIPLS have a zero probability of a successful attack. Hence, it verifies that the control protocol of the proposed H-HIPLS is secured from TCP reset attacks.

V. Discussion

A. Scalability

The key motivation of our proposal is to establish a scalable and secure VPLS architecture. We achieve the scalability in all three planes namely security, forwarding and control.

1) Scalability of the Control Plane : The simulation results verify (See Figure 5,6) that the proposed H-HIPLS establishes a less number of tunnels than S-HIPLS and HIPLS. Furthermore, it requires almost the same number of tunnels as H-LDP architecture. Hence, we can conclude that H-HIPLS significantly outruns the other secure VPLS architectures (S-HIPLS and HIPLS) in terms of control plane scalability. On the other hand, it provides almost similar performance as non-secured hierarchical VPLS architectures.

2) Scalability of the Forwarding Plane : Both S-HIPLS and H-HIPLS provide a similar performance for broadcast and multicast traffic (See Figure 10). Both architectures need a single encrypted frame for each broadcast frame. This frame can be replicated according to the spanning or multicast tree during the delivery. Hence, H-HIPLS also provide the similar forwarding plane scalability as other non secure hierarchical architectures.

3) Scalability of the Security Plane : The simulation results clearly shows (See Figure 7,9) that proposed H-HIPLS has similar or less key storage complexity than S-HIPLS. Furthermore, it significantly minimizes the key storage complexity of HIPLS in the network and in a PE. Hence we can conclude that H-HIPLS has better security plane scalability than other secure VPLS architectures namely S-HIPLS and HIPLS.

B. Security Assessment

The proposed H-HIPLS architecture provides the demanded security features for a VPLS network, namely authentication, confidentiality, integrity, availability, secure control protocol and robustness to the known attacks.

H-HIPLS architecture uses HIP signaling as the control protocol. The simulation results verified that it can protect the control protocol from IP based attacks such as TCP SYN DoS and TCP reset. Furthermore, HIP uses IPsec ESP mode of operation. Hence, both control and data frames are encrypted. Thus, it provides the integrity, confidentiality for the frames and protects the privacy. HIP based control protocol also prevents the eavesdropping on data and the in-flight alternation of control frames [21].

We propose an encrypted label based routing to ensure the end-to-end security. It also provides the privacy protection for CEs and eliminates the fake address advertisements.

On the other hand, all the PEs are authenticated based on ACLs. Moreover, both ends of the communication tunnels are mutually authenticated before forwarding any frame. Thus, the unauthorized access is not permitted. Finally, the efficient broadcast and multicast mechanism prevents the broadcast/multicast frame based DoS attacks.

C. The Distribution of Service Provision

The proposed H-HIPLS architecture distributes the service provision among different PEs. The u-PEs are responsible for the data encryption. Hence, they store the CEKs of VPNs. The intermediate n-PEs do not need to store CEKs and participate in data encryption. On the other hand, the n-PEs are responsible for other service provision functions such as dynamic address learning, forwarding table maintenance and dynamic HIP tunnel establishment. Hence, the proposed H-HIPLS distributes the service provision functions to optimized network resources such as memory space and processing power at PEs. Ultimately, it further enhances the control plane scalability.

VI. Conclusion

In this paper, we proposed a novel hierarchical VPLS solution based on Host Identity Protocol (HIP). It provides not only

TABLE I: A comparison of different VPLS architectures

	LDP based H-VPLS	**BGP based H-VPLS**	**HIPLS**	**S-HIPLS**	**Proposed H-HIPLS**
Architecture	Hierarchical	Hierarchical	Flat	Flat	Hierarchical
Scalability of Forwarding Plane	High	High	Low	High	High
Scalability of Security Plane	-	-	Low	High	High
Scalability of Control Plane	High	High	Low	Low	High
Protected Control Protocol	No	No	Yes	Yes	Yes
Data Traffic Encryption	No	No	Yes	Yes	Yes
IP Attack Protection	No	No	Yes	Yes	Yes
Efficiency of the Broadcast Mechanism	High	High	Low	High	High

the scalability in control, security and forwarding planes but also a range of security features. The simulation results verified that the proposed architecture significantly outruns the other secure VPLS solutions in terms of control plane scalability and slightly outrun in terms of security and forwarding plane scalability. Furthermore, it is able to provide almost the same level of control and forwarding plane scalability compared with the insecure hierarchical VPLS architectures. Finally, the simulation results confirmed that the proposed architecture has a secure control protocol which is protected from IP based attacks such as TCP SYN DoS and TCP reset.

In future studies, we will focus on studying the impact of the mobile PE on the H-VPLS networks.

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 consortiums view, but the consortium is not liable for any use that may be made of any of the information contained therein.

REFERENCES

[1] K. Kompella and Y. Rekhter, "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling," RFC 4761, IETF, January 2007.

[2] M. Lasserre and V. Kompella, "Virtual private LAN service (VPLS) using label distribution protocol (LDP) signaling," RFC 4762, IETF, January 2007.

[3] E. R. H. Shah and G. Heron, "IP-Only LAN Service (IPLS)," IETF, February 2007.

[4] T. Henderson, S. Venema, and D. Mattes, "HIP-based Virtual Private LAN Service (HIPLS)," *Internet Draft*, IETF, September 2011.

[5] M. Liyanage and A. Gurtov, "A Scalable and Secure VPLS Architecture for Provider Provisioned Networks," in *Proc. of IEEE Wireless Communication and Networking Conference: WCNC, Shanghai, China*, 2013.

[6] S. Khandekar, V. Kompella, J. Regan, *et al.*, "Hierarchical Virtual Private LAN Service," *Internet Draft*, IETF, June 2002.

[7] A. Sodder, K. Ramakrishnan, C. DelRegno, , and J. Wils, "Virtual Hierarchical LAN Services," *Internet Draft*, IETF, April 2003.

[8] C. Hu, C. Yuan, K. Liu *et al.*, "Enhanced H-VPLS service architecture using control word," Aug. 4 2009, US Patent 7,570,648.

[9] "DEMYSTIFYING H-VPLS," Juniper Networks, Inc, Tech. Rep., 2010. [Online]. Available: http://www.juniper.net/us/en/local/pdf/app-notes/3500116-en.pdf

[10] "H-VPLS N-PE Redundancy for QinQ and MPLS Access," CISCO Cooperation, Tech. Rep., 2011. [Online]. Available: http://www.cisco.com/en/US/docs/ios/mpls/configuration/guide/

[11] D. Zelig, L. Bruckman, and Y. Kotser, "Hierarchical virtual private LAN service protection scheme," Oct. 16 2007, US Patent 7,283,465.

[12] W. Augustyn and Y. Serbest, "Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks," RFC 4665, IETF, September 2006.

[13] M. Liyanage and A. Gurtov, "Securing Virtual Private LAN Service by Efficient Key Management," *Security and Communication Networks*, 2013.

[14] P. Nikander, A. Gurtov, and T. Henderson, "Host Identity Protocol (HIP): Connectivity, Mobility, Multi-Homing, Security, and Privacy over IPv4 and IPv6 Networks," *Communications Surveys & Tutorials, IEEE*, vol. 12, no. 2, pp. 186–204, 2010.

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

[16] R. Moskowitz, P. Nikander, and P. Jokela, "Host Identity Protocol," RFC 5201, 2008.

[17] "Architectural Considerations for Backhaul of 2G/3G and Long Term Evolution Networks," CISCO Cooperation, Tech. Rep., 2010.

[18] W. Eddy, "TCP SYN flooding attacks and common mitigations," RFC 4987, IETF, August 2007.

[19] P. A. Watson, "Slipping in the Window: TCP Reset attacks," Tech. Rep., 2004.

[20] G. Keller and A. Beylot, "Improving flow level fairness and interactivity in WLANs using size-based scheduling policies," in *Proc. of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile system*, 2008.

[21] M. Liyanage and A. Gurtov, "Secured VPN Models for LTE Backhaul Networks," in *Proc. of 76th Vehicular Technology Conference (VTC2012-Fall)*. IEEE, 2012.