BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 14 September 2023

A.S. Sajassi
A.B. Banerjee
S.T. Thoria
Cisco
D.C. Carrel
Graphiant
B.W. Weis
Independent
J.D. Drake
Juniper Networks
13 March 2023

Secure EVPN
draft-sajassi-bess-secure-evpn-06

Abstract

The applications of EVPN-based solutions ([RFC7432] and [RFC8365]) have become pervasive in Data Center, Service Provider, and Enterprise segments. It is being used for fabric overlays and inter-site connectivity in the Data Center market segment, for Layer-2, Layer-3, and IRB VPN services in the Service Provider market segment, and for fabric overlay and WAN connectivity in Enterprise networks. For Data Center and Enterprise applications, there is a need to provide inter-site and WAN connectivity over public Internet in a secured manner with same level of privacy, integrity, and authentication for tenant's traffic as IPsec tunneling using IKEv2. This document presents a solution where BGP point-to-multipoint signaling is leveraged for key and policy exchange among PE devices to create private pair-wise IPsec Security Associations without IKEv2 point-to-point signaling or any other direct peer-to-peer session establishment messages.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 14 September 2023.

Table of Contents

1. Introduction

The applications of EVPN-based solutions have become pervasive in Data Center, Service Provider, and Enterprise segments. It is being used for fabric overlays and inter-site connectivity in the Data Center market segment, for Layer-2, Layer-3, and IRB VPN services in the Service Provider market segment, and for fabric overlay and WAN connectivity in the Enterprise networks. For Data Center and Enterprise applications, there is a need to provide inter-site and WAN connectivity over public Internet in a secured manner with the same level of privacy, integrity, and authentication for tenant's traffic as used in IPsec tunneling using IKEv2. This document presents a solution where BGP point-to-multipoint signaling is leveraged for key and policy exchange among PE devices to create private pair-wise IPsec Security Associations without IKEv2 point-to-point signaling or any other direct peer-to-peer session establishment messages. This method is specially recommended for large scale deployment where large meshes of IKEv2 sessions among PE devices are not appropriate.

EVPN uses BGP as control-plane protocol for distribution of information needed for discovery of PEs participating in a VPN, discovery of PEs participating in a redundancy group, customer MAC addresses and IP prefixes/addresses, aliasing information, tunnel encapsulation types, multicast tunnel types, multicast group memberships, and other information. The advantages of using BGP control plane in EVPN are well understood including the following:

1. A full mesh of BGP sessions among PE devices can be avoided by using Route Reflector (RR) where a PE only needs to setup a single BGP session between itself and the RR as opposed to setting up N BGP sessions to N other remote PEs; therefore, reducing number of BGP sessions from O(N^2) to O(N) in the network. Furthermore, RR hierarchy can be leveraged to scale the number of BGP routes on the RR.

2. MP-BGP route filtering and constrained route distribution can be leveraged to ensure that the control-plane traffic for a given VPN is only distributed to the PEs participating in that VPN.

For setting up point-to-point security association (i.e., IPsec tunnel) between a pair of EVPN PEs, it is important to leverage BGP point-to-multipoint singling architecture using the RR along with its route filtering and constrain mechanisms to achieve the performance and the scale needed for large number of security associations (IPsec tunnels) along with their frequent re-keying requirements. Using BGP signaling along with the RR (instead of peer-to-peer protocol such as IKEv2) reduces number of message exchanges needed for SAs establishment and maintenance from O(N^2) to O(N) in the network.

Many key exchange methods (such as IKEv2) use a Diffie-Hellman (DH) algorithm to derive keys. When combined with an authentication method, the key exchange method allows two network devices to generate private pair-wise keys with each other. This document presents a key exchange method making use of the PE-to-RR trust model, where an RR is used to distribute keying material and policy between PE devices, also resulting in the PEs generating private pair-wise keys with each other. DH public values are provided to controllers from IPsec devices, where the controller relays the DH public values to authorized peers of that IPsec device as defined by a centralized policy. PE devices then create and install private pair-wise IPsec session keys to be used to secure communications with their peers.

Although IKEv2 is not used in this approach, the key management interfaces between IKEv2 and IPsec defined in RFC 7296 are maintained as much as possible.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119] RFC 8174 [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Terminology

* AC: Attachment Circuit.

* ARP: Address Resolution Protocol.

* BD: Broadcast Domain. As per RFC7432 [RFC7432], an EVI consists of a single or multiple BDs. In case of VLAN-bundle and VLAN-based service models (see RFC7432 [RFC7432]), a BD is equivalent to an EVI. In case of VLAN-aware bundle service model, an EVI contains multiple BDs. Also, in this document, BD and subnet are equivalent terms.

* BD Route Target: refers to the Broadcast Domain assigned Route Target RFC4364 [RFC4364]. In case of VLAN-aware bundle service model, all the BD instances in the MAC-VRF share the same Route Target.

* BT: Bridge Table. The instantiation of a BD in a MAC-VRF, as per RFC7432 [RFC7432].

* DGW: Data Center Gateway.

* Ethernet A-D route: Ethernet Auto-Discovery (A-D) route, as per [RFC7432].

* Ethernet NVO tunnel: refers to Network Virtualization Overlay tunnels with Ethernet payload. Examples of this type of tunnels are VXLAN or GENEVE [GENEVE].

* EVI: EVPN Instance spanning the NVE/PE devices that are participating on that EVPN, as per [RFC7432].

* EVPN: Ethernet Virtual Private Networks, as per [RFC7432].

* GRE: Generic Routing Encapsulation.

* GW IP: Gateway IP Address.

* IPL: IP Prefix Length.

* IP NVO tunnel: it refers to Network Virtualization Overlay tunnels with IP payload (no MAC header in the payload).

* IP-VRF: A VPN Routing and Forwarding table for IP routes on an NVE/PE. The IP routes could be populated by EVPN and IP-VPN address families. An IP-VRF is also an instantiation of a layer 3 VPN in an NVE/PE.

* IRB: Integrated Routing and Bridging interface. It connects an IP-VRF to a BD (or subnet).

* MAC-VRF: A Virtual Routing and Forwarding table for Media Access Control (MAC) addresses on an NVE/PE, as per [RFC7432]. A MAC-VRF is also an instantiation of an EVI in an NVE/PE.

* ML: MAC address length.

* ND: Neighbor Discovery Protocol.

* NVE: Network Virtualization Edge.

* GENEVE: Generic Network Virtualization Encapsulation, [GENEVE].

* NVO: Network Virtualization Overlays.

* RT-2: EVPN route type 2, i.e., MAC/IP advertisement route, as defined in [RFC7432].

* RT-5: EVPN route type 5, i.e., IP Prefix route. As defined in Section 3 of [EVPN-PREFIX].

* SBD: Supplementary Broadcast Domain. A BD that does not have any ACs, only IRB interfaces, and it is used to provide connectivity among all the IP-VRFs of the tenant. The SBD is only required in IP-VRF- to-IP- VRF use-cases (see Section 4.4.).

* SN: Subnet.

* TS: Tenant System.

* VA: Virtual Appliance.

* VNI: Virtual Network Identifier. As in [RFC8365], the term is used as a representation of a 24-bit NVO instance identifier, with the understanding that VNI will refer to a VXLAN Network Identifier in VXLAN, or Virtual Network Identifier in GENEVE, etc. unless it is stated otherwise.

* VTEP: VXLAN Termination End Point, as in RFC 7348 [RFC7348].

* VXLAN: Virtual Extensible LAN, as in RFC 7348 [RFC7348].

This document also assumes familiarity with the terminology of [RFC7432], [RFC8365], and [RFC7365].

3. Requirements

The requirements for secured EVPN are captured in the following subsections.

3.1. Tenant's Layer-2 and Layer-3 data and control traffic

Tenant's layer-2 and layer-3 data and control traffic must be protected by IPsec cryptographic methods. This implies not only tenant's data traffic must be protected by IPsec but also tenant's control and routing information that are advertised in BGP must also be protected by IPsec. This in turn implies that BGP session must be protected by IPsec.

3.2. Tenant's Unicast and Multicast Data Protection

Tenant's layer-2 and layer-3 unicast traffic must be protected by IPsec. In addition to that, tenant's layer-2 broadcast, unknown unicast, and multicast traffic as well as tenant's layer-3 multicast traffic must be protected by IPsec when ingress replication or assisted replication are used. The use of BGP P2MP signaling for setting up P2MP SAs in P2MP multicast tunnels is for future study.

3.3. P2MP Signaling for SA setup and Maintenance

BGP P2MP signaling must be used for IPsec SAs setup and maintenance. This reduces the number of message exchanges from O(N^2) to O(N) among the participating PE devices.

3.4. Granularity of Security Association Tunnels

The solution must support the setup and maintenance of IPsec SAs at the following level of granularities:

* Per PE: A single IPsec tunnel between a pair of PEs to be used for all tenants' traffic supported by the pair of PEs.

* Per tenant: A single IPsec tunnel per tenant per pair of PEs. For example, if there are 1000 tenants supported on a pair of PEs, then 1000 IPsec tunnels are required between that pair of PEs.

* Per subnet: A single IPsec tunnel per subnet (e.g., per VLAN/EVI) of a tenant on a pair of PEs.

* Per L3 flow: A single IPsec tunnel per pair of IP addresses of a tenant on a pair of PEs.

* Per L2 flow: A single IPsec tunnel per pair of MAC addresses of a tenant on a pair of PEs.

* Per AC pair: A single IPsec tunnel per pair of Attachment Circuits between a pair of PEs.

3.5. Support for Policy and DH-Group List

The solution must support a single policy and DH group for all SAs as well as supporting multiple policies and DH groups among the SAs.

4. SA and Key Management

The BGP Route Reflector (RR) acts as a trusted third party, which relays policy and keying material between PE devices. Communications between the RR and the PEs MUST be authenticated, encrypted, and integrity-protected. All algorithms are selected by the management station associated with the RR. The combination of the RR and a set of PE devices comprises of a cooperating group of devices that make up a VPN, where each PE device is authorized to communicate with other PE devices in the group. Policies can allow a PE device to communicate with all other PE devices in the group, or may restrict it to a subset of those devices.

DH public values from each PE are distributed to other authorized peer PEs via the RR. Each PE device creates and maintains a DH pair, which it uses to communicate with other members of the VPN. This distribution of DH public values (and other related values) is intended to be embedded into the BGP protocol as described later. In particular, the RR provides a mechanism for secure key management. However, it does not provide policy information or configuration as that is assumed to be provided by the management station.

4.1. Generating Initial IPsec SAs

When an PE device (PE) begins operation, it generates a private/public DH pair, using an algorithm defined in the IKEv2 Diffie-Hellman Group Transform IDs [IKEV2-IANA]. If the device does not have any active peers it simply distributes its DH public value to the BGP RR, along with a nonce to be used during SA creation. Whenever a private/public DH pair is created, a new nonce MUST also be created. Whenever DH public values are transmitted, they are

transmitted with the corresponding nonce. Whenever a DH private or DH public value is used, it is used along with the corresponding nonce. However, in the diagrams and descriptions below, the nonces are often left out for the sake of clarity.

Upon receiving a peer's DH public value and nonce, the receiver creates IPsec SAs (as described in Section 5.2). For each peer, a pair of IPsec SAs are created by combining the PE device's own DH private value with the DH public number received from the Controller.

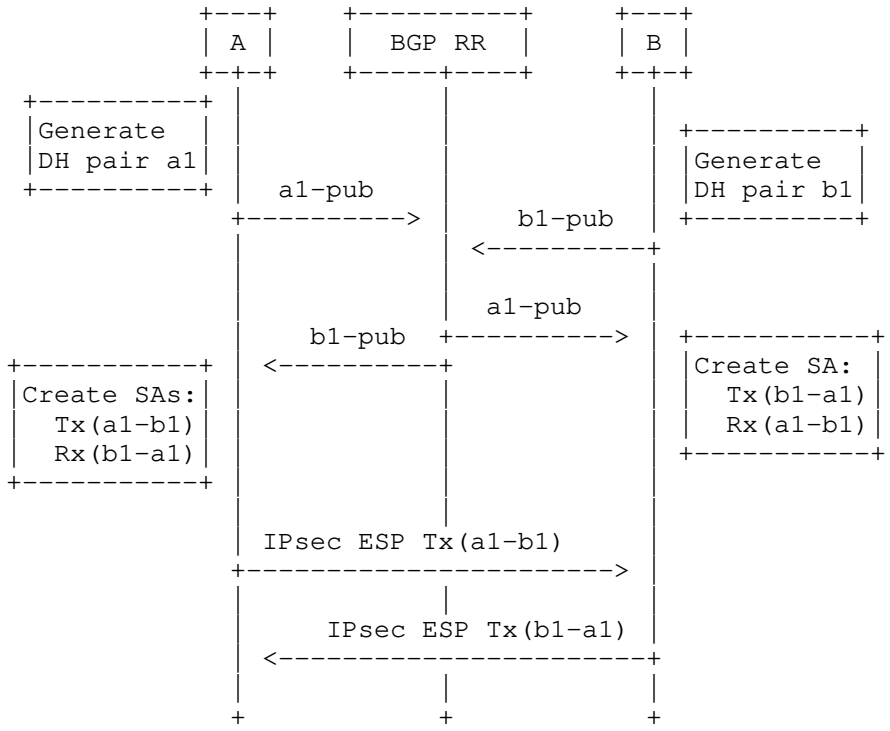


Figure 1: Generation of Initial IPsec SAs between two peers

Figure 1 shows IPsec SA generation between a pair of PE devices. Two PE devices (A and B shown in Figure 1) join the network. Each creates it's own DH pair (labelled "a1" on A and "b1" on B), and distributes the DH public value (labelled a1-pub and b1-pub) to the BGP RR. The BGP RR forwards the DH public value to all authorized peers, although for simplicity of exposition the figure only shows the two IPsec devices.

When each device receives the peer's DH public value, a pair of IPsec SAs are generated: one outbound and one inbound. As shown in the figure, A generates an outbound SA labeled Tx(a1-b1), representing that it has been generated using A's DH pair labeled a1 and B's DH pair labeled b1. B generates the same IPsec SA as an inbound SA, which is labeled Rx(a1-b1). Similarly, A generates an inbound IPsec SA labelled Rx(b1-a1), which is the same IPsec SA on B which is labelled Tx(b1-a1).

This process repeats on both A and B as they discover other PE devices with which they are authorized to communicate.

4.2. Rekey of IPsec SAs

Any IPsec device may initiate a rekey at any time. Common reasons to perform a rekey include a local time or volume based policy, or may be the result of a cipher counter mode Initialization Vector (IV) counter nearing its final value. The rekey process is performed individually for each remote peer. If rekeying is performed with multiple peers simultaneously, then the decision process and rules described in this rekey are performed independently for each peer.

A decision process choosing an outbound IPsec SA is followed when certain events occur, as described in the rules below. The same decision process is followed regardless of whether the device is performing a rekey or responding to a peer's rekey. The decision process is:

1. Determine the outbound SAs with the remote peer's most recently distributed DH public value.

2. Determine which of those outbound SAs are "live". A "live" outbound SA is one built from a DH value from the local peer for which it has observed inbound traffic using any SA based on the same local DH pair. This proves that the remote peer is prepared to receive traffic protected by that DH pair.

3. Choose the "live" outbound SA built from the local peer's most recent DH public value.

A rekey operation follows these four basic rules.

Rule 1: When an IPsec device needs to perform a rekey with a remote peer, it creates a new pair of IPsec SAs by combining the new DH private value with the peer's DH public values. If the remote peer is also in the midst of a rollover and its DH public value has already been received, then this may result in creating two sets of SAs: one pair with the remote peer's old DH public value, and one pair with the remote peer's new DH public value.

Rule 2: When an IPsec device receives a new remote peer's DH public value from the controller it creates and installs a new pair of IPsec SAs by combining the remote peer's new DH public value with its own current local DH private values. If both devices are in the midst of a rollover, this may result in creating two sets of SAs with the remote peer's new DH public value: one with the local old DH private value, and one with the local new DH private value. The outbound SA decision process is performed.

Rule 3: The first IPsec packet received by a rekeying IPsec device on an inbound SA using its new DH pair causes it to perform the outbound SA decision process. It may also shorten the lifetime of IPsec SAs using its own old DH pair that are shared with this peer, as they are no longer in use (other than the inbound SA might receive packets in transit).

Rule 4: The first IPsec packet received from a remote rekeying IPsec device using the remote peer's new DH pair allows the IPsec device to shorten the lifetime of IPsec SAs shared with this peer using unused remote DH pairs.

Two examples follow: a single IPsec device performing a rekey with its peers, and two IPsec devices performing a simultaneous rekey. The same rekey operations described above are exhibited in both cases.

4.2.1. Single IPSec Device Rekey

When a single IPsec device begins a rekey, it first generates a new DH pair and generates new IPsec SA pairs for each peer with which it is communicating. It does this by combining the new DH private value with each peer's existing DH public value. Only when the new IPsec SAs have been installed and the device is prepared to receive on those new SAs does it then distribute the new DH public value to the Controller, which forwards the new DH public value to its authorized peers. The rekeying IPsec device continues to transmit on the old SAs for each peer until it observes that peer begin to transmit on the new SAs.

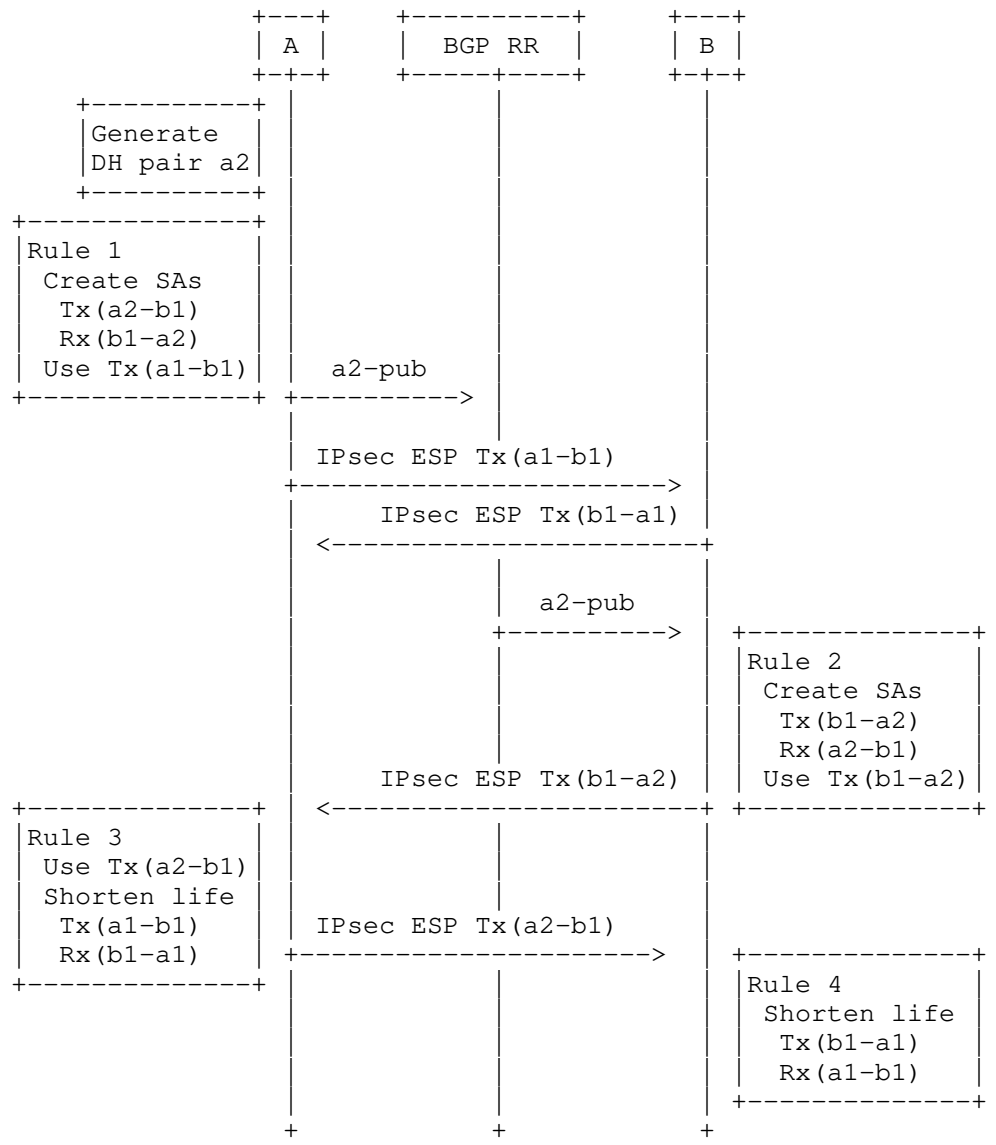


Figure 2: Single IPSec Device Rekey between two peers

In Figure 3, device A is shown as performing a rekey, and it creates a DH pair labelled "a2". The following steps are followed.

1. Rule 1 requires creating new IPsec SAs for each peer. In this example, A creates a new outbound IPsec SA to communicate with B labelled Tx(a2-b1), and a new inbound IPsec SA labelled Rx(b1-a2). A continues to transmit on Tx(a1-b1) (generated as shown in Figure 2).

2. A distributes the new public value (a2-pub) to the Controller who forwards it to A's authorized peers, which includes B. During this time, both A and B continue to use the initial IPsec SAs setup between them using a1 and b1.

3. When B receives a2 from the controller, B follows Rule 2 by creating Tx(b1-a2), Rx(a2-b1). B also follows the outbound SA decision process, which causes it to change its outbound IPsec SA to A to Tx(b1-a2).

4. When A receives a packet protected by Rx(b1-a2), it follows Rule 3 and performs the outbound SA decision process. This causes it to change its outbound IPsec SA to Use Tx(a2-b1). It also optionally shortens the lifetime of the old IPsec SAs shared with this peer.

5. When B receives a packet protected by Tx(a2-b1), it follows Rule 4, in which it may shorten the lifetime of the old IPsec SAs shared with this peer using DH pairs that are no longer in use.

At the end of the rekey, both A and B retain a single DH pair, and a single set of IPsec SAs between them.

4.2.2. Multiple IPSec Device Rekey

When two or more IPsec device simultaneously begin a rekey, they each follow the rekeying method described in the previous section. Every rekeying IPsec device generates a new DH pair and generates new IPsec SA pairs for each peer with which it is communicating by combining their new DH private value with each peer's existing DH public value. When this completes on a particular IPsec device, it distributes the new DH public value to the Controller, which forwards it to its authorized peers. Each continues to transmit on the existing SAs for each peer until it observes that peer transmitting on the new SAs. During a simultaneous rekey up to four pairs of IPsec SAs may be temporarily created, but the four rules ensure that they converge on a single new set of IPsec SAs.

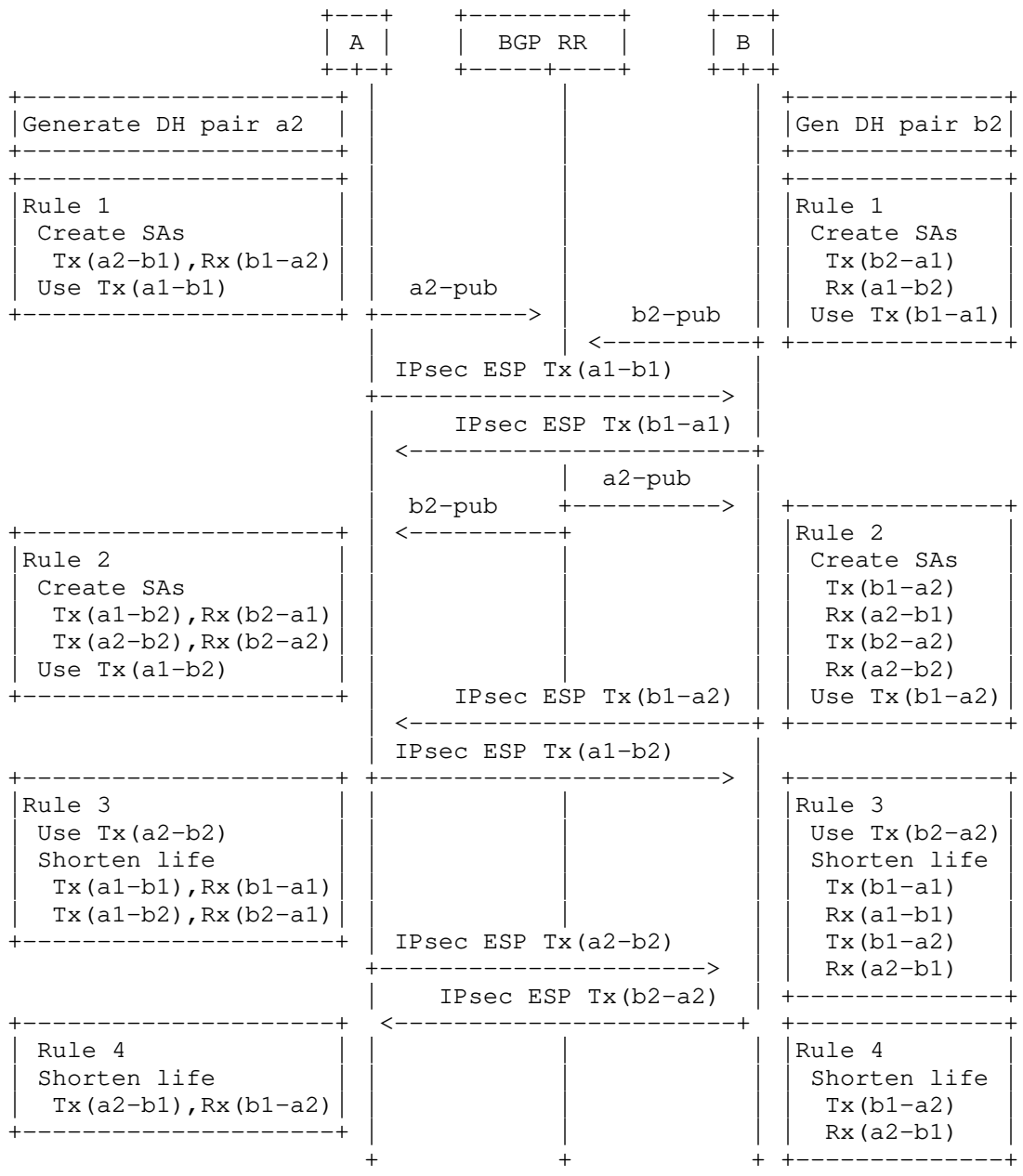


Figure 3: Simultaneous IPsec Device Rekey between two peers

In Figure 4, device A and device B are both shown as performing a rekey. Their initial state corresponds to the final state shown in Figure 2 (i.e., they are communicating using a single pair of IPsec SAs created from DH pairs "a1" and "b1".

1. A and B follow Rule 1, which includes creating new IPsec SAs for each peer. In this example, A creates a new outbound IPsec SA to communicate with B labelled Tx(a2-b1), and a new inbound IPsec SA labelled Rx(b1-a2). B creates a new outbound IPsec SA to communicate with B labelled Tx(a1-b2), and a new inbound IPsec SA labelled Rx(b2-a1). A and B continue to transmit on IPsec SAs previously created from DH pairs "a1" and "b1".

2. A distributes the new public value (a2-pub) to the Controller who forwards it to A's authorized peers, which includes B. B also distributes the new public value (b2-pub) to the Controller who forwards it to B's authorized peers, which includes A.

3. When A and B receive each other's new peer DH public value from the controller they follows Rule 2. But because now there are four DH values that could be in used between A and B, they must be prepared to use IPsec SAs using each permutation of DH values: a1-b1, a1-b2, a2-b1, a2-b2. Prior to implementing Rule 2, each has already created sets of IPsec SAs matching two of the permutations, so just two more sets must be generated during Rule 2.

   * One pair is created using the IPsec device's old DH pair with the peer's new DH pair. This is necessary, because the peer may transmit on this pair.

   * One pair is created using the IPsec device's new DH pair with the peer's new DH pair. This is the set of IPsec SAs that will be used at the end of the rekey process.

   Each peer begins transmitting on an IPsec SA that combines the remote peer's new DH pair and its own old DH pair, which is the most recent "live" SA on which it can transmit. I.e., A begins transmitting on Tx(a1-b2) and B begins transmitting on Tx(b1-a2).

4. When A receives a packet protected by Rx(b1-a2), it understands that the remote peer has received its new DH public value. A also understands that because of Rule 2 that B must have created IPsec SAs using a2-b2. This allows A to follow Rule 3 and change its outbound IPsec SA to Use Tx(a2-b2). Similarly, when B receives a packet protected by Rx(a1-b2), B recognizes that it can also begin to transmit using Tx(b2-a2). Note that it also possible that A will receive a packet protected by Rx(b2-a2) or B will receive a packet protected by Rx(a2-b2), and then knows it can transmit on an IPsec SA using both of the new DH pairs.

5. Also in Rule 3, Both A and B optionally shorten the lifetime of older IPsec SAs shared with this peer derived from unused DH pairs to be cleaned up. A shortens the lifetime of SAs based on a1. B shortens the lifetime of SAs based on b1.

6. When A and B receive a packet protected by the remote peer's latest DH pair, they shortens the lifetime of SAs based on the remote peer's unused DH pair.

5. IPsec Database Generation

The PAD, SPD, and SAD all need to be setup as defined in the IPsec Security Architecture [RFC4301].

5.1. The Security Policy Database (SPD)

The SPD is implemented using methods outside the scope of this document. The SPD describes the type of traffic that will be protected between IPsec devices and the policy (e.g., ciphers) used to create SAs.

5.2. Security Association Database (SAD)

The SAD is constructed from IPsec policy (e.g., ciphers) obtained (depending on the controller protocol method) either from the controller or distributed by a peer (see Section 6).

Keying Material is generated following the method defined in IKEv2, and depends on SPIs, nonces, and the Diffie-Hellman shared secret.

The following sections describe how the necessary values are determined.

5.2.1. Generating Keying Material for IPsec SAs

5.2.1.1. g^ir

A DH public value is distributed from the peer.

A DH shared secret (g^ir) is computed using the peer's public value, and the device's private value. The DH group to be used must be known by the device. Options include distribution by an SDN controller, or distribution by the peer with the DH public value (see Section 6).

5.2.1.2. Nonces

Nonces are distributed with a DH public value, and are used only with that value. It is RECOMMENDED that nonces are generated as described in Section 2.10 of [RFC7296].

IKEv2 Key derivation specifies an initiator's nonce (Ni) and a responder's nonce (Nr). While neither peer is truly initiating a session), in order to fit the IKE key material models the roles must be assigned. The initiator is chosen as the peer with the larger nonce and the responder is the peer with the smaller. This does mean that the roles can change for each rekey and for each SA within a rekey.

5.2.1.3. SPIs

SPI values that are unique to each generation of keying material need to be determined. While each peer could distribute its own inbound SA value, the SPI value would be used by many peers. Although this is not a problem for an SA lookup (lookup can include the source and destination IP addresses), experience has shown that this is sub-optimal for some hardware SA lookup algorithms. Instead, this specification proposes generating values that are unpredictable and indistinguishable from randomly-generated SPI values.

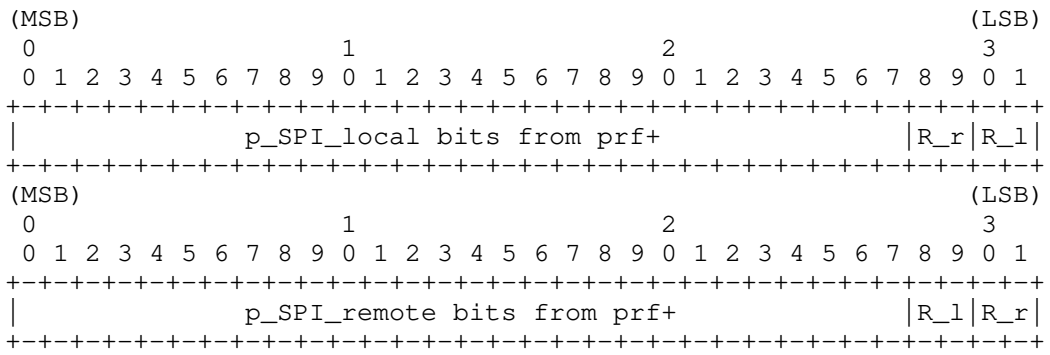
SPI values are generated using the IKEv2 prf+ function, where nonces are used as the input to the prf. This produces a statistically random SPI value that should be unique. However, with a 32 bit value there is still a very small, but non-zero, chance of SPIs repeating for a given pair of peers. To prevent this and ensure uniqueness in the operational window, we also use the lower 2 bits from each peer's rekey counter.

First the SPIs are taken from the prf+ function as 32 bit values and assigned based on which peer is taking the role of initiator and which is taking the role of responder. The p_SPI_i is taken by the device providing Ni, where p_SPI_r is taken by the other device.

{p_SPI_i | p_SPI_r } = prf+(Ni | Nr, "SPI generation")

Next p_SPI_i and p_SPI_r are mapped from initiator and responder roles to local and remote roles based on the choice of Ni and Nr in 5.2.1.2 and are renamed to p_SPI_local and p_SPI_remote.

Then, 2 2-bit Rotation Numbers (RN) are generated from the 2 least significant bits (LSB) of the 2 rekey counter values (see Section 6). These 4 bits replace the least significant bits of p_SPI_local and p_SPI_remote with the local RN bits taking the least significant position in p_SPI_local and the remote RN bits taking the least significant position in p_SPI_remote. This shown in the following two diagrams with RN_local shown as R_l and RN_remote shown as R_r.



The reason for changing terminology from initiator/responder to local/remote is because the roles of initiator/responder can change in every rekey. The order of RN_local and RN_remote needs to remain constant. If that order was based on initiator/responder, there's a risk that if the initiator and responder roles changed and the two peers re-keyed on different frequencies, they could end up with identical RN values.

In some circumstances additional values may also need to be added to the prf for peers to ensure that they have implemented the same policy. Appendix A.3.1 includes a discussion of when this might be needed. In these cases, only the prf+ inputs are modified and the Rotation Numbers MUST still be added as above.

Because a device is not choosing its inbound SPI, its SA lookup process needs to be aware that duplicates could occur across different peers. In that case, the inbound SA Lookup SHOULD include a source IP address in addition to the SPI value (see Section 4.1 of [RFC4301]).

5.2.1.4. IPsec key generation

As described in previous sections, a DH public value and a nonce are distributed by peers. These are used to generate IPsec keys following the method defined in the IKEv2. SKEYSEED is generated following Section 2.14 of [RFC7296]:

SKEYSEED = prf(Ni | Nr, g^ir)

KEYMAT can be similarly derived as defined by IKEv2 (Section 2.17 of [RFC7296]), although only SK_d is required to be generated (shown in Section 2.14 of [RFC7296]).

SK_d = prf+ (SKEYSEED, Ni | Nr | SPIi | SPIr)

KEYMAT = prf+(SK_d, Ni | Nr)

However, with the simplification where only SK_d is generated, it can be observed that the derivation of SK_d could be skipped entirely, and an optimized derivation of KEYMAT could be as follows:

KEYMAT = prf+ (SKEYSEED, Ni | Nr | SPIi | SPIr)

Note: A single specification for generating KEYMAT will be determined in a future version of this document.

5.3. Peer Authorization Database (PAD)

The PAD identifies authorized peers. PAD entries are either statically configured, or may be dynamically updated by the controller.

The PAD omits authentication data for each peer, because it has delegated authentication and authorization to the controller.

The controller protocol MUST be able to describe an identity that a receiver can match against its local PAD database, to ensure that the peer is an authorized peer.

6. Policy distributed through the BGP RR

An IPsec device distributes to a controller a DH public value and the associated information and policy needed to create IPsec SAs in a Device Information Message (DIM). The controller then distributes the DIM to all authorized peers of that device. The following data elements MUST be embedded in a DIM message:

* DH public number (used for key computation)

* Nonce (used for key computation and SPI generation)

* Peer identity (used to identify a peer in the PAD)

* An Indication whether this is the initial distributed policy

* A rekey counter, which increases for each unique DIM sent

In cases where a single fixed IPsec policy has been pre-distributed, it is not necessary for the peer to send or receive that policy in a DIM. However, in cases where an IPsec device needs to indicate the policy it is willing to use, the following data elements SHOULD be included in a DIM:

* An IPsec policy or policies

* A lifetime bounding the use of the DH public number. When this DH public number is used to create an IPsec SA, the shortest lifetime is used as an SA lifetime for the pair of generated IPsec SAs. When the lifetime expires, the local version of the DIM and IPsec SAs generated from it MUST be deleted.

Appendix A suggests different ways that this policy may be included in a controller protocol. This document does not define a normative protocol format, because the DIM very likely needs to be integrated into an existing controller protocol rather than be an independent key management protocol. However, the controller protocol MUST provide a strong authentication between the device and the controller, and integrity of the messages MUST be provided. Confidentiality of the messages SHOULD also be provided. It is important that the controller protocol be protected with algorithms that are at least as strong as the algorithms used to protect the IPsec packets.

6.1. IPsec policy negotiation

In many controller based networks, there is a single IPsec policy used by all devices and there is no need to redistribute or select policy details. However, in some circumstances, there may be a need to have multiple policy options. This could happen when a controller changes the policy and wants to smoothly migrate all devices to the new policy. Or it could happen if a network supports devices with different capabilities. In these cases, devices need to be able to choose the correct policy to use for each other device, and must do this without sending additional messages and without sending individual messages to each peer. When a device supports multiple policies, it MUST include those policies within the DIM. This is done by sending multiple distinct policies, in order of preference,

where the first policy is the most preferred. The policy to use is selected by taking the receiver's list of policies (i.e., the list advertised by the device that generates Nr), starting with the first policy, compare against the initiator's (device that generates Ni) list, and choosing the first one found in common. The method conforms to the IKEv2 Cryptographic Algorithm Negotiation described in Section 2.7 of [RFC7296]. (However, see additional discussion when IKEv2 payloads are used in Appendix A.3.1).

If there is no match, this indicates a controller configuration error. These devices MUST NOT establish new SAs until a DIM is received that does produce a match.

When a device supports more than one DH group, then a unique DH public number MUST be specified for each in order of preference. The selection of which DH group to use follows the same logic as Policy selection, using the receiver's list order until a match is found in the initiator's list.

7. BGP Component

The architecture that encompasses device-to-controller trust model, has several components among which is the signaling component. Secure EVPN Signaling, as defined in this document, is the BGP signaling component of the overall Architecture. We will briefly describe this Architecture here to further facilitate understanding how Secure EVPN fits into the overall architecture. The Architecture describes the components needed to create BGP based SD-WANs and how these components work together. Our intention is to list these components here along with their brief description and to describe this Architecture in details in a separate document where to specify the details for other parts of this architecture besides the BGP signaling component which is described in this document.

The Architecture consists of four components. These components are Zero Touch Bring-up, Configuration Management, Orchestration, and Signaling. In addition to these components, secure communications must be provided between the edge nodes and all servers/devices providing the architecture components.

7.1. Zero Touch Bring-up (ZTB)

The first component is a zero touch capability that allows an edge device to find and join its SD-WAN with little to no assistance other than power and network connectivity. The goal is to use existing work in this area. The requirements are that an edge device can locate its ZTB server/component of its SD-WAN controller in a secure manner and to proceed to receive its configuration.

7.2. Configuration Management

After an edge device joins its SD-WAN, it needs to be configured. Configuration covers all device configuration, not just the configuration related to Secure EVPN. The previous Zero Touch Bring-up component will have directed the edge device, either directly or indirectly, to its configuration server/component. One example of a configuration server is the I2NSF Controller. After a device has been configured, it can engage in the next two components. Configuration may include updates over time and is not a one time only component.

7.3. Orchestration

This component is optional. It allows for more dynamic updates of configuration and statistics information. Orchestration can be more dynamic than configuration.

7.4. Signaling

Signaling is the component described in this document. The functionality of a Route Reflector is well understood. Here we describe the signaling component of BGP SD-WAN Architecture and the BGP extension/signaling for IPsec key management and policy.

8. Solution Description

This solution uses BGP P2MP signaling where an originating PE only send a message to the Route Reflector (RR) and then the RR reflects that message to the interested recipient PEs. The framework for such signaling is described in section 4 and it is referred to as device-to-controller trust model. This trust model is significantly different than the traditional peer-to-peer trust model where a P2P signaling protocol such as IKEv2 [RFC7296] is used in which the PE devices directly authenticate each other and agree upon security policy and keying material to protect communications between themselves. The device-to-controller trust model leverages P2MP signaling via the controller (e.g., the RR) to achieve much better scale and performance for establishment and maintenance of large number of pair-wise Security Associations (SAs) among the PEs.

This device-to-controller trust model first secures the control channel between each device and the controller using peer-to-peer protocol such as IKEv2 [RFC7296] to establish P2P SAs between each PE and the RR. It then uses this secured control channel for P2MP signaling in establishment of P2P SAs between each pair of PE devices.

Each PE advertises to other PEs via the RR the information needed in establishment of pair-wise SAs between itself an every other remote PEs. These pieces of information are sent as Sub-TLVs of IPSec tunnel type in BGP Tunnel Encapsulation attribute. These Sub-TLVs are detailed in section 10 and are based on the DIM message components in section 5 and the IKEv2 specification [RFC7296]. The IPsec tunnel TLVs along with its Sub-TLVs are sent along with the BGP route (NLRI) for a given level of granularity.

If only a single SA is required per pair of PE devices to multiplex user traffic for all tenants, then IPsec tunnel TLV is advertised along with IPv4 or IPv6 NLRI representing loopback address of the originating PE. It should be noted that this is not a VPN route but rather an IPv4 or IPv6 route.

If a SA is required per tenant between a pair of PE devices, then IPsec tunnel TLV can be advertised along with EVPN IMET route representing the tenant or can be advertised along with a new EVPN route representing the tenant.

If a SA is required per tenant's subnet (e.g., per VLAN) between a pair of PE devices, then IPsec tunnel TLV is advertised along with EVPN IMET route.

If a SA is required between a pair of tenant's devices represented by a pair of IP addresses, then IPsec tunnel TLV is advertised along with EVPN IP Prefix Advertisement Route or EVPN MAC/IP Advertisement route.

If a SA is required between a pair of tenant's devices represented by a pair of MAC addresses, then IPsec tunnel TLV is advertised along with EVPN MAC/IP Advertisement route.

If a SA is required between a pair of Attachment Circuits (ACs) on two PE devices (where an AC can be represented by {VLAN, port}), then IPsec tunnel TLV is advertised along with EVPN Ethernet AD route.

8.1. Inheritance of Security Policies

Operationally, it is easy to configure a security association between a pair of PEs using BGP signaling. This is the default security association that is used for traffic that flows between peers. However, in the event more finer granularity of security association is desired on the traffic flows, it is possible to set up SAs between a pair of tenants, a pair of subnets within a tenant, a pair of IPs between a subnet, and a pair of MACs between a subnet using the appropriate EVPN routes as described above. In the event, there are no security TLVs associated with an EVPN route, there is a strict

order in the manner security associations are inherited for such a route. This results in an EVPN route inheriting the security associations of the parent in a hierarchical fashion. For example, traffic between an IP pair is protected using security TLVs announced along with the EVPN IP Prefix Advertisement Route or EVPN MAC/IP Advertisement route as a first choice. If such TLVs are missing with the associated route, then one checks to see if the subnets the IPs are associated with has security TLVs with the EVPN IMET route. If they are present, those associations are used in securing the traffic. In the absence of them, the peer security associations are used. The order in which security associations are inherited are from the granular to the coarser, namely, IP/MAC associated TLVs with the EVPN route being the first preference, and the subnet, the tenant, and the peer associations preferred in that fashion.

It should be noted that when a security association is made it is possible for it to be re-used by a large number of traffic flows. For example, a tenant security association may be associated with a number of child subnet routes. Clearly it is mandatory to keep a tenant security association alive, if there are one or more subnet routes that want to use that association. Logically, the security associations between a pair of entities creates a single secure tunnel. It is thus possible to classify the incoming traffic in the most granular sense {IP/MAC, subnet, tenant, peer} to a particular secure tunnel that falls within its route hierarchy. The policy that is applied to such traffic is independent from its use of an existing or a new secure tunnel. It is clear that since any number of classified traffic flows can use a security association, such a security association will not be torn down, if at least there is one policy using such a secure tunnel.

8.2. Distribution of Public Keys and Policies

One of the requirements for this solution is to support a single DH group and a single policy for all SAs as well as to support multiple DH groups and policies among the SAs. The following subsections describe what pieces of information (what Sub-TLVs) are needed to be exchanged to support a single DH group and a single policy versus multiple DH groups and multiple policies.

8.2.1. Minimal DIM

For SA establishment, at the minimum, a PE needs to advertise to other PEs, its DIM values as specified in section 5. These include:

ID Tunnel ID
N Nonce
RC Rekey Counter
I Indication of initial policy distribution
KE DH public value.

When this minimal set of DIM values is sent, then it is assumed that all peer PEs share the same policy for which DH group to use, as well as which IPSec SA policy to employ. Section 5.1 defines the Minimal DIM sub-TLV as part of IPsec tunnel TLV in BGP Tunnel Encapsulation Attribute.

8.2.2. Multiple Policies

There can be scenarios for which there is a need to have multiple policy options. This can happen when there is a need for policy change and smooth migration among all PE devices to the new policy is required. It can also happen if different PE devices have different capabilities within the network. In these scenarios, PE devices need to be able to choose the correct policy to use for each other. This multi-policy scheme is described in section 6. In order to support this multi-policy feature, a PE device MUST distribute a policy list. This list consists of multiple distinct policies in order of preference, where the first policy is the most preferred one. The receiving PE selects the policy by taking the received list (starting with the first policy) and comparing that against its own list and choosing the first one found in common. If there is no match, this indicates a configuration error and the PEs MUST NOT establish new SAs until a message is received that does produce a match.

8.2.3. Multiple DH-groups

It can be the case that not all peers use the same DH group. When multiple DH groups are supported, the peer may include multiple KE Sub-TLVs. The order of the KE Sub-TLVs determines the preference. The preference and selection methods are specified in section 6.

8.2.4. Multiple or Single ESP SA policies

In order to specify an ESP SA Policy, a DIM may include one or more SA Sub-TLVs. When all peers are configured by a controller with the same ESP SA policy, they MAY leave the SA out of the DIM. This minimizes messaging when group configuration is static and known. However, it may also be desirable to include the SA. If a single SA is included, the peer is indicating what ESP SA policy it uses, but is not willing to negotiate. If multiple SA Sub-TLVs are included, the peer is indicating that it is willing to negotiate. The order of the SA Sub-TLVs determines the preference. The preference and

selection methods are specified in section 6.

8.3. Initial IPsec SAs Generation

The procedure for generation of initial IPsec SAs is described in section 4. This section gives a summary of it in context of BGP signaling. When a PE device first comes up and wants to setup an IPsec SA between itself and each of the interested remote PEs, it generates a DH pair along for each [what word here? "tennant"?] using an algorithm defined in the IKEv2 Diffie-Hellman Group Transform IDs [IKEv2-IANA]. The originating PE distributes the DH public value along with the other values in the DIM (using IPsec Tunnel TLV in Tunnel Encapsulation Attribute) to other remote PEs via the RR. Each receiving PE uses this DH public number and the corresponding nonce in creation of IPsec SA pair to the originating PE - i.e., an outbound SA and an inbound SA. The detail procedures are described in Section 4.1.

8.4. Re-Keying

A PE can initiate re-keying at any time due to local time or volume based policy or due to the result of cipher counter nearing its final value. The rekey process is performed individually for each remote PE. If rekeying is performed with multiple PEs simultaneously, then the decision process and rules described in this rekey are performed independently for each PE. Section 4.2 describes this rekeying process in details and gives examples for a single IPsec device (e.g., a single PE) rekey versus multiple PE devices rekey simultaneously.

8.5. IPsec Databases

The Peer Authorization Database (PAD), the Security Policy Database (SPD), and the Security Association Database (SAD) all need to be setup as defined in the IPsec Security Architecture RFC 4301 [RFC4301]. Section 5 of this document gives a summary description of how these databases are setup where key is exchanged via P2MP signaling through the RR and the policy can be either signaled via the RR (in case of multiple policies) or configured by the management station (in case of single policy).

9. Encapsulation

Vast majority of Encapsulation for Network Virtualization Overlay (NVO) networks in deployment are based on UDP/IP with UDP destination port ID indicating the type of NVO encapsulation (e.g., VxLAN, GPE, GENEVE, GUE) and UDP source port ID representing flow entropy for load-balancing of the traffic within the fabric based on n-tuple that includes UDP header. When encrypting NVO encapsulated packets using IP Encapsulating Security Payload (ESP), the following two options can be used: a) adding a UDP header before ESP header (e.g., UDP header in clear) and b) no UDP header before ESP header (e.g., standard ESP encapsulation). The following subsection describe these encapsulation in further details.

9.1. Standard ESP Encapsulation

When standard IP Encapsulating Security Payload (ESP) is used (without outer UDP header) for encryption of NVO packets, it is used in transport mode as depicted below. When such encapsulation is used, for BGP signaling, the Tunnel Type of Tunnel Encapsulation TLV is set to ESP-Transport and the Tunnel Type of Encapsulation Extended Community is set to NVO encapsulation type (e.g., VxLAN, GENEVE, GPE, etc.). This implies that the customer packets are first encapsulated using NVO encapsulation type and then it is further encapsulated and encrypted using ESP-Transport mode.

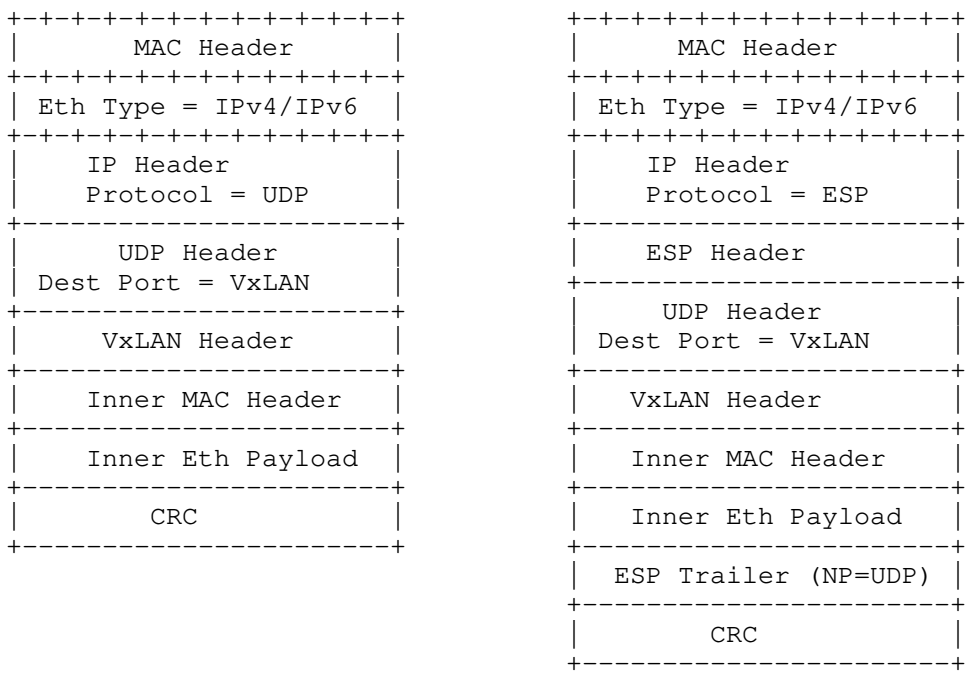


Figure 4

9.2. ESP Encapsulation within UDP packet

In scenarios where NAT traversal is required (RFC 3948 [RFC3948]) or where load balancing using UDP header is required, then ESP encapsulation within UDP packet as depicted in the following figure is used. The ESP for NVO applications is in transport mode. The outer UDP header (before the ESP header) has its source port set to flow entropy and its destination port set to 4500 (indicating ESP header follows). A non-zero SPI value in ESP header implies that this is a data packet (i.e., it is not an IKE packet). The Next Protocol field in the ESP trailer indicates what follows the ESP header, is a UDP header. This inner UDP header has a destination port ID that identifies NVO encapsulation type (e.g., VxLAN). Optimization of this packet format where only a single UDP header is used (only the outer UDP header) is for future study.

When such encapsulation is used, for BGP signaling, the Tunnel Type of Tunnel Encapsulation TLV is set to ESP-in-UDP-Transport and the Tunnel Type of Encapsulation Extended Community is set to NVO encapsulation type (e.g., VxLAN, GENEVE, GPE, etc.). This implies that the customer packets are first encapsulated using NVO encapsulation type and then it is further encapsulated and encrypted using ESP-in-UDP with Transport mode.

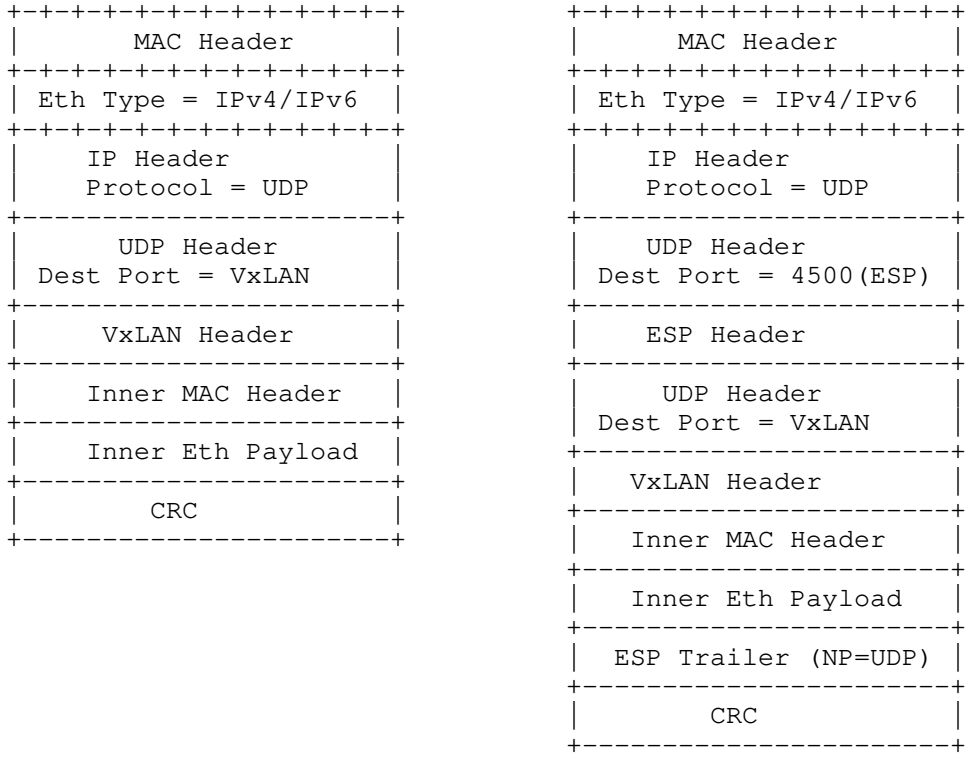


Figure 5

10. BGP Encoding

This document defines two new Tunnel Types along with its associated sub-TLVs for The Tunnel Encapsulation Attribute [TUNNEL-ENCAP]. These tunnel types correspond to ESP-Transport and ESP-in-UDP-Transport as described in section 4. The following sub-TLVs apply to both tunnel types unless stated otherwise.

10.1. The Base (Minimal Set) DIM Sub-TLV

The Base DIM is described in 3.2.1. One and only one Base DIM may be sent in the IPSec Tunnel TLV.

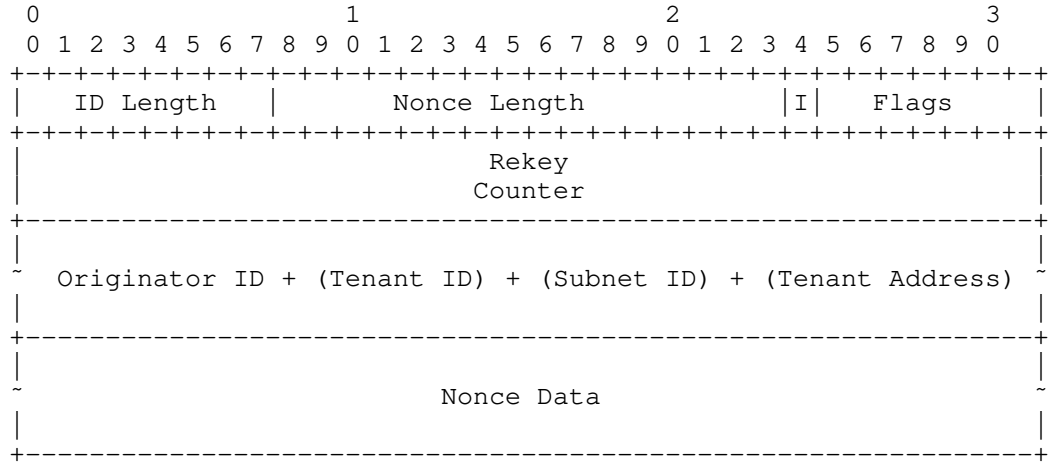


Figure 6

ID Length (16 bits) is the length of the Originator ID + (Tenant ID) + (Subnet ID) + (Tenant Address) in bytes. Nonce Length (8 bits) is the length of the Nonce Data in bytes I (1 bit) is the initial contact flag Flags (7 bits) are reserved and MUST be set to zero on transmit and ignored on receipt. The Rekey Counter is a 64 bit rekey counter The Originator ID + (Tenant ID) + (Subnet ID) + (Tenant Address) is the tunnel identifier and uniquely identifies the tunnel. Depending on the granularity of the tunnel, the fields in () may not be used - i.e., for a tunnel at the PE level of granularity, only Originator ID is required. The Nonce Data is the nonce. Its length is a multiple of 32 bits. Nonce lengths should be chosen to meet minimum requirements described in IKEv2 [RFC7296].

10.2. The Key Exchange Sub-TLV

The KE Sub-TLV is described in 3.2.1 and 3.2.2.1. A KE is always required. One or more KE Sub-TLVs may be included in the IPSec Tunnel TLV.

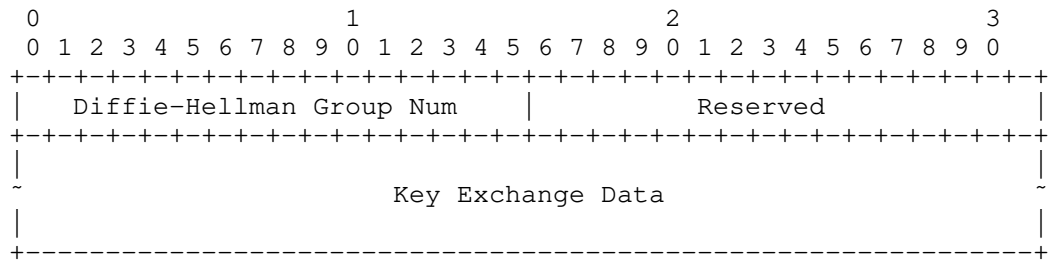


Figure 7

Diffie-Hellman Group Num 916 bits) identifies the Diffie-Hellman group in the Key Exchange Data was computed. Diffie-Hellman group numbers are discussed in IKEv2 [RFC7296] Appendix B and [RFC5114].

The Key Exchange payload is constructed by copying one's Diffie-Hellman public value into the "Key Exchange Data" portion of the payload. The length of the Diffie-Hellman public value is described for MOPD groups in [RFC7296] and for ECP groups in [RFC4753].

10.3. ESP SA Proposals Sub-TLV

The SA Sub-TLV is described in 3.2.2.2. Zero or more SA Sub-TLVs may be included in the IPSec Tunnel TLV.

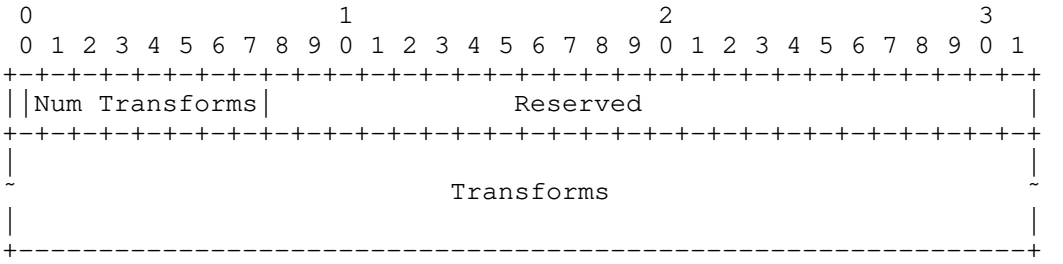


Figure 8

Num Transforms is the number of transforms included. Reserved is not used and MUST be set to zero on transmit and MUST be ignored on receipt.

10.3.1. Transform Substructure

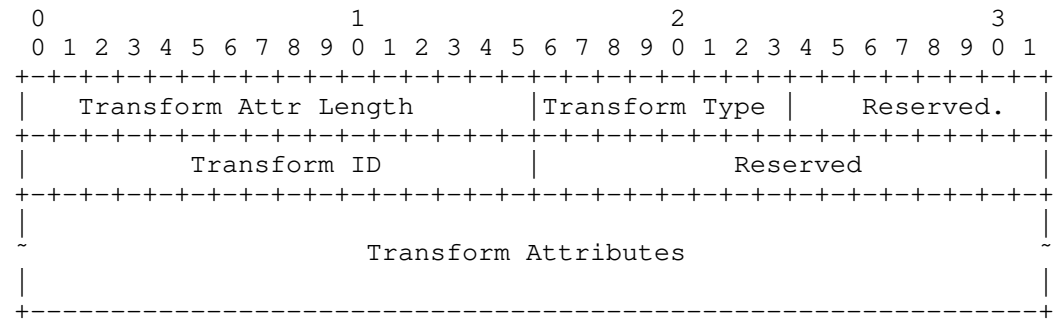


Figure 9

The Transform Attr Length is the length of the Transform Attributes field. The Transform Type is from Section 3.3.2 of [RFC7296] and [IKEV2IANA]. Only the values ENCR, INTEG, and ESN are allowed. The Transform ID specifies the transform identification value from [IKEV2IANA]. Reserved is unused and MUST be zero on transmit and MUST be ignored on receipt. The Transform Attributes are taken directly from 3.3.5 of [RFC7296].

11. Applicability

Although P2MP BGP signaling for establishment and maintenance of SAs among PE devices is described in this document in context of EVPN, there is no reason why it cannot be extended to other VPN technologies such as IP-VPN RFC 4364 [RFC4364], VPLS RFC 4761 [RFC4761] and RFC 4762 [RFC4762], and MVPN RFC 6513 [RFC6513] and RFC 6514 [RFC6514] with ingress replication. The reason EVPN has been chosen is because of its pervasiveness in DC, SP, and Enterprise applications and because of its ability to support SA establishment at different granularity levels such as: per PE, Per tenant, per subnet, per Ethernet Segment, per IP address, and per MAC. For other VPN technology types, a much smaller granularity levels can be supported. For example for VPLS, only the granularity of per PE and per subnet can be supported. For per-PE granularity level, the mechanism is the same among all the VPN technologies as IPsec tunnel type (and its associated TLV and sub-TLVs) are sent along with the PE's loopback IPv4 (or IPv6) address. For VPLS, if per-subnet (per bridge domain) granularity level needs to be supported, then the IPsec tunnel type and TLV are sent along with VPLS AD route.

The following table lists what level of granularity can be supported by a given VPN technology and with what BGP route.

Functionality	EVPN	IP-VPN	MVPN	VPLS
per PE	IPv4/v6 route	IPv4/v6 route	IPv4/v6 rte	IPv4/v6
per tenant	IMET (or new)	lpbk (or new)	I-PMSI	N/A
per subnet	IMET	N/A	N/A	VPLS AD
per IP	EVPN RT2/RT5	VPN IP rt	*,G or S,G	N/A
per MAC	EVPN RT2	N/A	N/A	N/A

Figure 10

12. Acknowledgements

TBD.

13. IANA Considerations

A new transitive extended community Type of 0x06 and Sub-Type of TBD for EVPN Attachment Circuit Extended Community needs to be allocated by IANA.

14. Security Considerations

This document proposes that a device re-use an ephemeral Diffie-Hellman exponential with multiple peers. There are some known potential vulnerabilities to this approach, which can be mitigated by the device first validating a peer's public value to be a safe public value before combining its own private value with it. The tests which MUST be performed are described in [RFC6989]. See [REUSE] for additional security considerations when reusing ephemeral Diffie-Hellman keys.

A controller acts as a "trusted third party", which asserts that a particular Diffie-Hellman public value is associated with a particular entity. A device receiving the public key is not required to validate the assertion.

A subverted controller can act as a "man-in-the-middle" between a pair of devices. The easiest attack would be for the attacker to adjust the routing for the desired traffic through a compromised gateway and directly observe the cleartext. It is also possible that a subverted controller could provide a device with a Diffie-Hellman public value that actually belongs to a compromised gateway rather

than the intended gateway, but doing so does not seem to be necessary. Nonetheless, the attack of a subverted controller can be mitigated by having a device sign its Diffie-Hellman public value (e.g, as a CMS Signed data object), where the receiver validates the digital signature on the object. However, this adds significant processing cost to a rekey and does not fit the controller-based network architecture model.

A subverted IPsec device whose DH pair has been compromised would be vulnerable to all of its IPsec traffic using that DH pair being compromised. Assuming the use of strong DH algorithms (including quantum resistant algorithms as they become available), the compromise would most likely be due to the device itself being compromised. Such a compromised device is also vulnerable to a direct plaintext compromise.

PFS is achieved between rekey periods, as DH pairs are required to be generated independently. However, because a device uses the same long-term key to generate session key with multiple peers, there is no PFS between sessions within the same rekey period. To reduce key exposure outside of a rekey period, when a connection is closed each endpoint MUST forget not only the keys used by the connection but also any information that could be used to recompute those keys. However, the DH private key value and the nonce distributed with it may be forgotten only once the last IPsec SA that uses the private key value is removed from the SAD and there is no chance that a new IPsec SA could be setup that requires the private key value.

If quantum resistance is considered to be an issue, the controller can distribute a PSK, which could be used to create the SK_d in the manner shown in [I-D.ietf-ipsecme-qr-ikev2].

15. References

15.1. Normative References

[GENEVE] Gross, J., et al., "Geneve: Generic Network Virtualization Encapsulation", 2018, <https://tools.ietf.org/html/draft-ietf-nvo3-geneve-06>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3948] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, DOI 10.17487/RFC3948, January 2005, <https://www.rfc-editor.org/info/rfc3948>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <https://www.rfc-editor.org/info/rfc7432>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8365] Sajassi, A., Ed., Drake, J., Ed., Bitar, N., Shekhar, R., Uttaro, J., and W. Henderickx, "A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)", RFC 8365, DOI 10.17487/RFC8365, March 2018, <https://www.rfc-editor.org/info/rfc8365>.

15.2. Informative References

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <https://www.rfc-editor.org/info/rfc4761>.

[RFC4762] Lasserre, M., Ed. and V. Kompella, Ed., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, DOI 10.17487/RFC4762, January 2007, <https://www.rfc-editor.org/info/rfc4762>.

[RFC6513] Rosen, E., Ed. and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February 2012, <https://www.rfc-editor.org/info/rfc6513>.

[RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012, <https://www.rfc-editor.org/info/rfc6514>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

Appendix A. Additional Stuff

TBD.

Authors' Addresses

Ali Sajassi
Cisco
170 W Tasman Drive
San Jose, CA
United States of America
Email: sajassi@cisco.com

Ayan Banerjee
Cisco
170 W Tasman Drive
San Jose, CA
United States of America
Email: ayabaner@cisco.com

Sameer Thoria
Cisco
170 W Tasman Drive
San Jose, CA
United States of America
Email: sthoria@cisco.com

David Carrel
Graphiant
CA
United States of America
Email: carrel@graphiant.com

Brian Weis
Independent
CA
United States of America

Email: bew.stds@gmail.com

John Drake
Juniper Networks
CA
United States of America
Email: jdrake@juniper.net