IPSECME
Internet-Draft
Intended status: Standards Track
Expires: August 19, 2013

D. Migault (Ed)
Francetelecom - Orange

February 15, 2013

IKEv2 Alternate Outer IP Address Extension
draft-mglt-ipsecme-alternate-outer-address-00.txt

Abstract

Current IKEv2 protocol has been designed to establish VPNs with the same outer IP addresses as those used for the IKEv2 channel. This describes the alternate outer IP address extension, and IKEv2 extension that enables the VPN End User to negotiate a VPN on different interfaces as those used for the IKEv2 channel.

Thus, this extension makes possible a VPN End User with multiple interfaces to set an IPsec tunnel on each interface with a Security Gateway by using a single IKEv2 channel instead of using an IKEv2 channel per interface. Similarly, for distributed Security Gateways, it also makes possible to split the IKEv2 and IPsec traffic on different interfaces.

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 http://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 August 19, 2013.

Table of Contents

1. Requirements notation

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 [RFC2119].

2. Introduction

When a VPN End User establishes a VPN with a Security Gateway, it starts by establishing an authenticated channel for IKEv2. Then the VPN Security Associations [RFC4301] are negotiated via the IKEv2 [RFC5996] channel. Once the peers agree on the Security Associations, the VPN can be used.

Currently, IKEv2 does not negotiate the outer IP addresses of the VPN. The security Association set these VPN outer IP addresses as the IP addresses used by the IKEv2 channel.

These implicit values are perfect for VPN End Users with a single interface. This was the case for a long time, making them unnecessary to be negotiated. However, today’s VPN End Users and Security Gateways have multiple interfaces. Relying on the default value of the VPN outer IP addresses makes it hard, - or at least in a non optimal way - to take advantage of multiple interfaces. This document specifies how alternate outer IP addresses can be negotiated during the Security Association negotiation. This involves new signaling, thus the document also specify how the VPN End User and the Security Gateway can optionally inform each other they support the alternate outer IP address extension.

The remaining of this document is as follows. Section 3 defines the terminology used in this document. Section 4 provides scenarios that motivate this alternate outer IP address extension. Section 5 describes the new protocol, as well as the new involved entities and Section 6 describes the payload format defined for the protocol. In this document, we assumed that no NAT are between the VPN End User and the Security Gateway, however, Section 7 provides some considerations when NAT is used.

The alternate outer IP address extension provides VPN End Users and Security Gateway a way to take advantage of multiple interfaces for a VPN service.

3. Terminology

This section defines terms and acronyms used in this document.

- VPN End User: designates the End User that initiates the VPN with a Security Gateway. This End User may be mobile and moves its VPN from on Security Gateway to the other.

- Security Gateway: designates a point of attachment for the VPN service. In this document, the VPN service is provided by multiple Security Gateways. Each Security Gateway may be considered as a specific hardware.

- Security Association (SA): The Security Association is defined in [RFC4301].

4. Alternate outer address scenarios

This section provides scenarios where a VPN End User and a Security Gateways share more than one VPN. For each scenario, the document describes the alternatives that currently exist, their limitations and the motivations for the alternate outer IP address extension. The scenarios herein are a subset of the scenarios described in [I-D.mglt-mif-security-requirements].

4.1. VPN End User with Multiple Interfaces

More and more terminals have multiple interfaces, and a VPN End User may take advantage of these multiple interfaces by setting multiple tunnels with its Security Gateways as represented in figure 1. A typical example would be a VPN End User attached to its Radio Access Network via Interface_0 and attached to a WLAN access point via Interface_1. The VPN End User may use one or the other interface according to the Quality of Service or the fees associated to each network. In figure 1. the VPN End User has established two distinct VPNs, one on each of its interfaces. Both VPNs are attached to the same Security Gateway interface. A packet can be sent or received from either one or the other VPN.

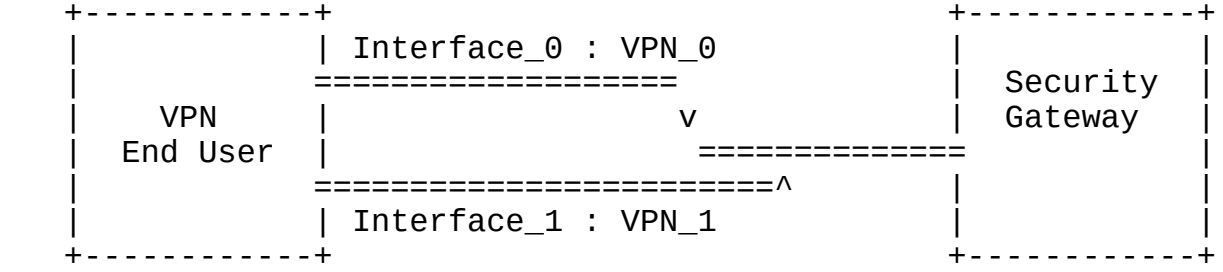

Figure 1: VPN End User with Multiple Interfaces

SAs negotiated for the VPN_0 and VPN_1 have the same network

configuration except that the outer interface of VPN_0 on the End User side is Interface_0 whereas VPN_1 has Interface_1. More specifically, these SAs have the same Selectors.

[RFC4301] section 4.1 states that parallel SAs are compliant with the IPsec architecture, and that traffic may be sent to one or the other VPN, for example, according to the Differentiated Services Code Point (DSCP). DSCP is called a "classifier" which differs from the Selector. How the End User chooses which interface to use is beyond the scope of this document.

As mentioned in [RFC5996] the VPN uses the IP addresses of the IKEv2 channel as outer IP addresses. One way to establish these two VPNs is to create an IKEv2 channel for each interface. This results in unnecessary IKE negotiations with multiple authentications Nbr(EU_interfaces) X Nbr(SG_interface) X Nbr(Flows). This number rapidly grows with the number of involved interfaces both on the Security Gateway and on the End User.

[RFC6027] section 3.8 mentions that peers using different IP addresses for the VPN and the IKEv2 channel SHOULD be modified unless they may drop the packets. The alternate outer IP address described in this document is described so that any VPN End User can interact with any Security Gateway.

[I-D.arora-ipsecme-ikev2-alt-tunnel-addresses] addresses this issue. The End User VPN indicates during the SA negotiation the outer IP address it wants, and in return the Security Gateway indicates the outer IP address of the Security Gateway. Motivations for [I-D.arora-ipsecme-ikev2-alt-tunnel-addresses] is a cluster of Security Gateways that splits the IKEv2 traffic and the VPN traffic, so that the VPN traffic avoids overloading some equipments like firewalls or load balancers for example.

[I-D.arora-ipsecme-ikev2-alt-tunnel-addresses] would also address the case of figure 1 because the the path used by the VPN is defined by the interface used by the VPN End User VPN. This results from the fact that the Security Gateway has only one interface. However, [I-D.arora-ipsecme-ikev2-alt-tunnel-addresses] would need slight modifications in order to address the more general case where VPN End User and the Security Gateways have multiple interfaces. In that case, a path would be defined not by a single interface (as in figure 1), but by a pair of interface.

In addition to path negotiation, [I-D.arora-ipsecme-ikev2-alt-tunnel-addresses] uses a Notify Payload that is not bound to a SA Proposal, thus making multiple SA Proposals with different outer IP address difficult. Again this case is very

specific to multiple interfaces. Even though the protocol described in this document address these limitations, it remains very closed to [I-D.arora-ipsecme-ikev2-alt-tunnel-addresses].

4.2. Security Gateway with Multiple Interfaces

In the scenario presented in figure 2, the VPN End User has two interfaces and the VPN End User has a single interface. Like the VPN End User with multiple interfaces presented in Section 4.1, we suppose that the VPNs are established by the VPN End User with the Security Gateway. Unlike the scenarios of Section 4.1, motivations for choosing VPN_0 or VPN_1 are not associated to the interface used by the VPN End User, but the path taken by the packets. As a result, the VPN End User cares about both source and destination outer IP addresses that defines the path.

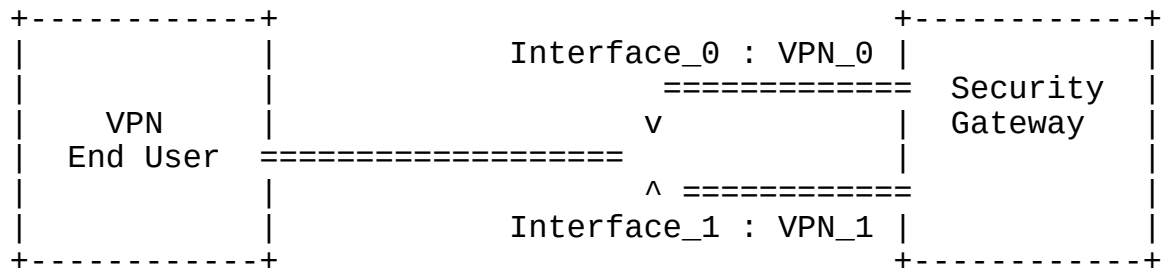

Figure 2: Security Gateway with Multiple Interfaces

Comments of Section 4.1 also applies to this scenario, but this scenario stresses that the choice of the VPN outer IP addresses SHOULD result from a negotiation between the two peers, and both outer IP addresses SHOULD be negotiated.

Note that the scenario described in figure 2, considers that all interfaces are used to setup all different VPNs. As described in Section 4.1, if VPN End Users and Security Gateways have both multiple interfaces, setting up all possible tunnels may be unnecessarily heavy. As a result, the VPN End User SHOULD be able to negotiate both outer IP addresses of its VPN.

Note that if the VPN End User negotiates the outer IP address used by the Security Gateway, the VPN End User may know in advance what interfaces are available. It is beyond the scope of this document to define how the VPN End User may know this information. MOBIKE [RFC4555] defines the ADDITIONAL_IP*_ADDRESSES Notify Payload, and [I-D.mglt-ipsecme-security-gateway-discovery] defines how these pieces of information may be provided by other Security Gateways.

4.3. Distributed Security Gateways

The scenario described in figure 3 considers a distributed Security Gateway. The IKEv2 channel and the VPNs are handled by different nodes. As a result the VPN does not uses the same outer IP addresses as the IKEv2 channel.

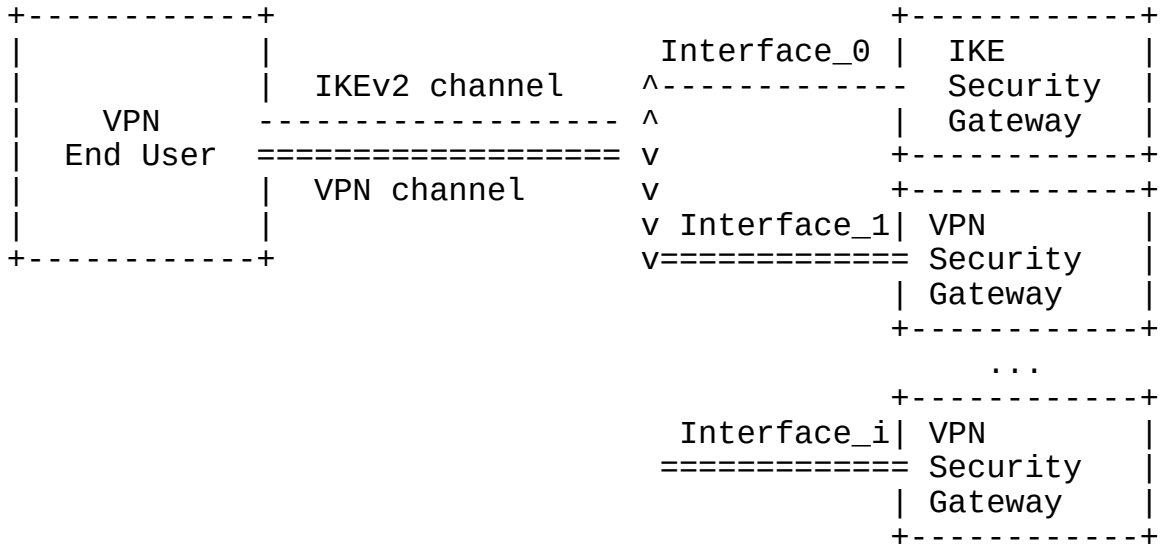

Figure 3: Distributed Security Gateways

This scenario is addressed by [I-D.arora-ipsecme-ikev2-alt-tunnel-addresses] where each part can choose the interface it will use as the outer IP address for the VPN. As mentioned in Section 4.1, being able to specify a single interface is not sufficient to select a path. More specifically, in figure 3, this would not provide the possibility for the VPN End User to choose between Interface_1 and Interface_i.

5. Protocol Overview

The alternate outer IP address extension, makes possible two peers to negotiate and agree on alternate IP addresses for their VPN. We consider the outer IP addresses as parameters of the Security Association. (Tunnel header IP source and destination address as described in [RFC4301]). As a consequence, the negotiation of these parameters occurs during the negotiation of the Security Association, that is during the IKE_INIT exchange or the CREATE_CHILD_SA exchange as described in [RFC5996].

VPN SA negotiation can be initiated by either VPN End User or the Security Gateway, so in the remaining of the document we simply use

initiator and responder, as the peer initiating the negotiation.

Note that these negotiations makes possible that any peer can negotiate one, or both outer IP address, that is to say, the outer IP address source and destination.

Section 5.1 briefly reminds how the Security Association' parameters are negotiated with IKEv2, and then proposes the new involved payloads to negotiate the outer IP addresses. Basically a new Alternate Outer Address Transform (OADD) and a new IP Attribute are defined. Section 5.2 and Section 5.3 and Section 5.4 are focused on the exchanged when both peers support the alternate outer IP address extension. Section 5.2 describes how the initiator builds a SA Proposal and Section 5.3 defines how the responder handles it. Section 5.4 defines the case where the Proposal MUST be discarded. Although not mandatory, there MAY be an advantage that peers are informed whether the alternate outer IP address is supported or not before sending Proposals. Section 5.5 presents how peers can inform each other the support this extension. At last, Section 5.6 illustrates the different exchanged described in the document.

5.1. Alternate outer IP addresses Transform

This section does not intend to explain how SAs are negotiated, and the reader is expected to refer to [RFC5996] section 3.3. This section briefly sums up the different type of payload involved in order to clarify our purpose. Figure 4 is copied from [RFC5996] to illustrate concepts involved in the Security Association negotiation.

```
SA Payload
   |
   +--- Proposal #1 ( Proto ID = ESP(3), SPI size = 4,
   |     |            7 transforms,      SPI = 0x052357bb )
   |     |
   |     +-- Transform ENCR ( Name = ENCR_AES_CBC )
   |     |     +-- Attribute ( Key Length = 128 )
   |     |
   |     +-- Transform ENCR ( Name = ENCR_AES_CBC )
   |     |     +-- Attribute ( Key Length = 192 )
   |     |
   |     +-- Transform ENCR ( Name = ENCR_AES_CBC )
   |     |     +-- Attribute ( Key Length = 256 )
   |     |
   |     +-- Transform INTEG ( Name = AUTH_HMAC_SHA1_96 )
   |     +-- Transform INTEG ( Name = AUTH_AES_XCBC_96 )
   |     +-- Transform ESN ( Name = ESNs )
   |     +-- Transform ESN ( Name = No ESNs )
   |
   +--- Proposal #2 ( Proto ID = ESP(3), SPI size = 4,
         |            4 transforms,      SPI = 0x35a1d6f2 )
         |
         +-- Transform ENCR ( Name = AES-GCM with a 8 octet ICV )
         |     +-- Attribute ( Key Length = 128 )
         |
         +-- Transform ENCR ( Name = AES-GCM with a 8 octet ICV )
         |     +-- Attribute ( Key Length = 256 )
         |
         +-- Transform ESN ( Name = ESNs )
         +-- Transform ESN ( Name = No ESNs )
```

Figure 4: Security Association Payload Structure

A Security Association is defined by various parameters such as Encryption (ENCR) or Integrity (INTEG), Pseudorandom Function (PRF), Diffie-Hellman group (D-H) or Extended Sequence Numbers (ESN). These parameters are defined through Transforms and each parameter is a Transform Type.

A Security Association is negotiated through the SA Payload which contains one or more Proposals Payloads. Each Proposal contains one or multiple acceptable "values" for each Transformed Type. These "values" can be seen as an OR. The Proposal is accepted if for each Transform Type one of the proposed "value" is accepted by the responder. If the responder cannot choose an acceptable "value" for each Transform Type, the proposition is rejected. A "value" is composed of a Transform ID, like the name of the encryption algorithm, and eventually one or more Attributes, like the key length

for example.

In our case, we consider a new Transform Type OADD. This Transform Type has two Transform ID (INIT or RESP) that designates the initiator outer IP address (INIT) or the responder outer IP address (RESP). The Attributes associated to each Transform ID is the IP Attribute that can be an IPv4 address, an IPv6 address or a specific value.

5.2. Initiator: Sending OADD Transforms in Proposals

In Section 5.2 and Section 5.3 we suppose that both the initiator and the responder support the alternate outer IP address extension, that no USE_TRANSPORT_MODE Notify Payload is sent in conjunction of the SA Payload, and that the Proposal Payload as defined in [RFC5996] Section 3.3.1 has its Protocol ID set to AH or ESP. Other cases are discussed in Section 5.4

If the initiator wants to propose the Security Gateway to choose among a set of the initiator's interfaces IP_init_0, ..., IP_init_k for the VPN outer IP address, it MUST include k+1 Transforms with Transform Type OADD and Transform ID set to INIT. The Transform is associated to the Attribute of Type IP. Transform Attributes are defined in [RFC5996] 3.3.5.

Similarly, if the initiator wants to select on the Security Gateway one interface among a set of interface IP_resp_0, ..., IP_resp_l, it MUST include l+1 OADD Transform with Transform ID set to RESP, and an Attribute of Type IP.

If the initiator does not know the interface that the responder may choose, it may indicate the responder to define the most appropriated interface with a OADD Transform with Transform ID set to RESP and an Attribute of Type with the specific value ANY_IP.

If no OADD Transform with Transform ID set to INIT (Respectively RESP) are provided in the Proposal, the default value for the outer IP address is the one used by the IKEv2 channel. More specifically, if the initiator considers the interface used for the IKEv2 channel as an alternative to other IP addresses, a OADD Transform with this IP address MUST explicitly be in the Proposal.

Note that a Proposal does not need to have both OADD Transform with Transform ID INIT and RESP. The initiator can choose to have only OADD Transforms with Transform ID INIT (respectively RESP).

5.3. Responder: Receiving OADD Transforms in Proposals

As mentioned in Section 5.2, we suppose the responder supports the alternate outer IP address extension. If a Proposal contains one or multiple OADD with a Transform ID set to INIT (respectively RESP), the responder choose one of these. If selected OADD Transform (INIT or RESP) with an IP Attribute, the responder returns the Transform without modification. Otherwise, if selected OADD Transform is with an ANY_IP Attribute, the responder returns a IP Attribute with the correct value.

If the responder has no OADD Transform with Transform ID INIT (respectively RESP), then by default the outer IP address of the VPN is equal to the IP address used by the IKEv2 channel.

5.4. Incompatible Proposal with OADD Transforms

The alternate outer IP address extension only makes sense for the IPsec tunnel mode. The SA Payload with Proposals that contains one or more OADD Transforms MUST NOT be used with a USE_TRANSPORT_MODE Notify Payload. Responder MUST reject these Proposals.

Similarly, Proposals with a Protocol other than AH or ESP, (that is to say IKE), MUST NOT be used with OADD Transforms. Responder MUST reject these Proposals.

As mentioned in [RFC5996] Section 3.3.6, a responder that does not support the alternate outer IP address extension MUST reject any Proposal that contains a Transform with a Transform Type OADD. If the responder rejects all Proposals, it MUST send a NO_PROPOSAL_CHOSEN Notify Payload.

5.5. Supporting alternate outer IP address exchange

This section describes an informational exchange where each peer informs the other that it supports the alternate outer IP address extension. This exchange is not mandatory, but is recommended as it MAY ease to format the Proposals for the Security Association negotiation.

In fact the negotiation of the alternate outer IP address is included in SA negotiation. As described in Section 5.1, this introduces new Transform Type and new Attributes. [RFC5996] Section 3.3.6 mentions that a peer does not understand the new Transform Type or the new Attributes, it MUST reject the Proposal. As a result, if the initiator does not know if the responder supports the alternate outer IP address extension, it SHOULD include proposals without the associated Transform Type and Attributes to avoid that all Proposals

are rejected by the responder and receives a NO_PROPOSAL_CHOSEN Notify Payload.

To limit the number of proposals to be sent by the initiator during the SA negotiation, we define the supporting alternate outer IP address exchange where the initiator can advertise it supports the alternate outer IP address extension by sending a ALTERNATE_OUTER_IP_ADDRESS_SUPPORTED Notify Payload. When a node receives this Notify Payload and support the alternate outer IP address extension, it MUST send back the same Notify Payload.

5.6. Basic Exchange

Figure 5 provides a basic exchange. The initiator and the responder agree on supporting the alternate outer IP address extension. This exchange is optional but recommended. In Figure 5 this exchange occurs during the IKE_INIT exchange, but it MAY occur anytime.

The SA negotiation consists in sending multiple Proposals. In figure 5, the OADD Transform specify the initiator and responder’s IP address. The responder choose one of the proposed transformed.

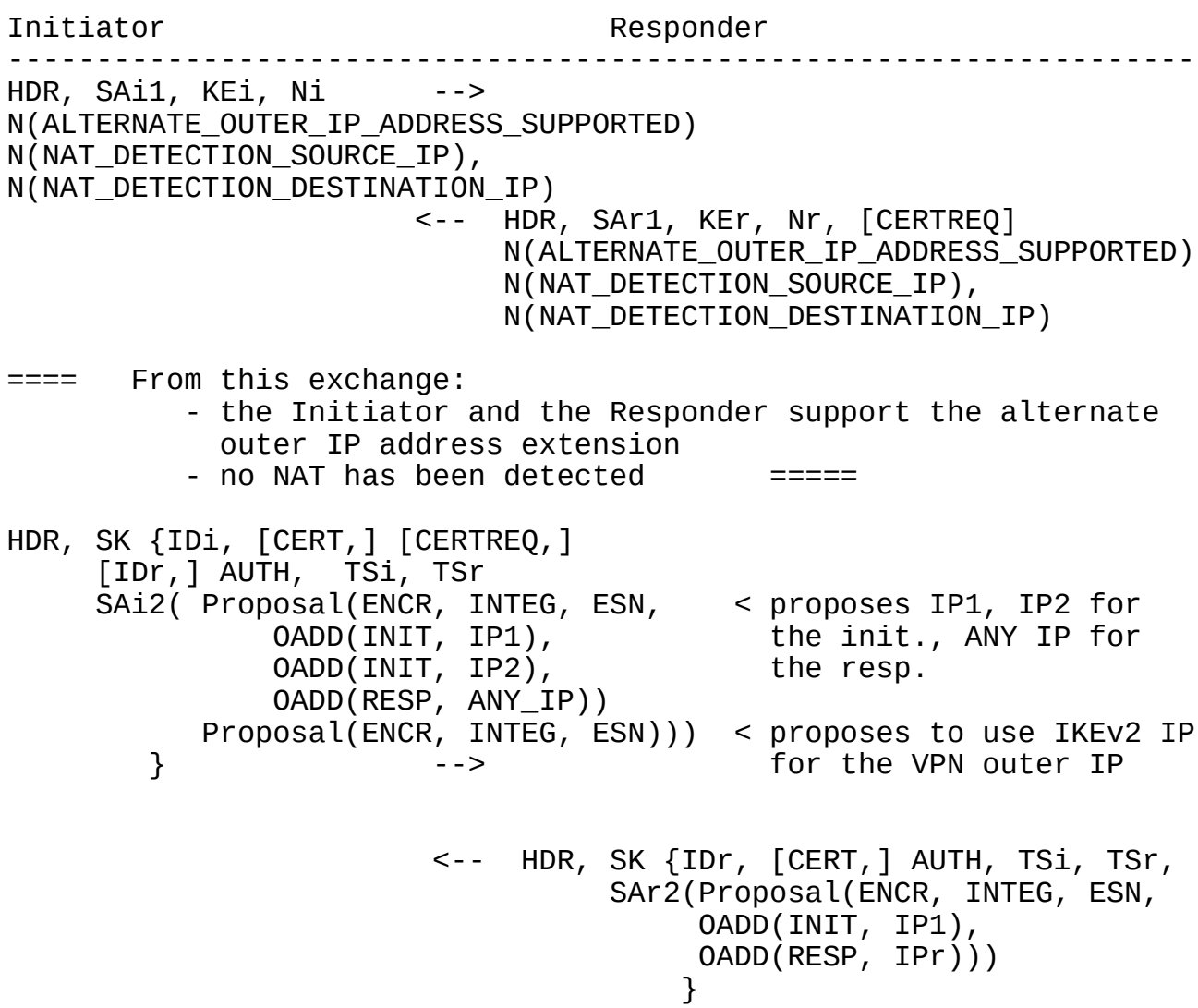

Figure 5: Basic Exchange for VPN alternate outer IP addresses negotiation

6. Payload Formats

As mentioned in Section 5 this document introduces a new Transform of Transform Type OADD. The associated Transform ID are INIT for the initiator outer IP address and RESP for the responder’s IP address. These Transforms are associated a Attributes that are either carrying an IP address (IPv4 or IPv6) or associated to a specific value like ANY_IP.

This document also introduces the ALTERNATE_OUTER_IP_ADDRESS_SUPPORTED Notify Payload, so peers can

inform the other they support the alternate outer IP address extension.

This section describes the format of all new payload introduced for the outer IP address extension.

6.1. Outer IP address Transform OADD

This section specifies the Transform structure as defined in [RFC5996] Section 3.3.2.

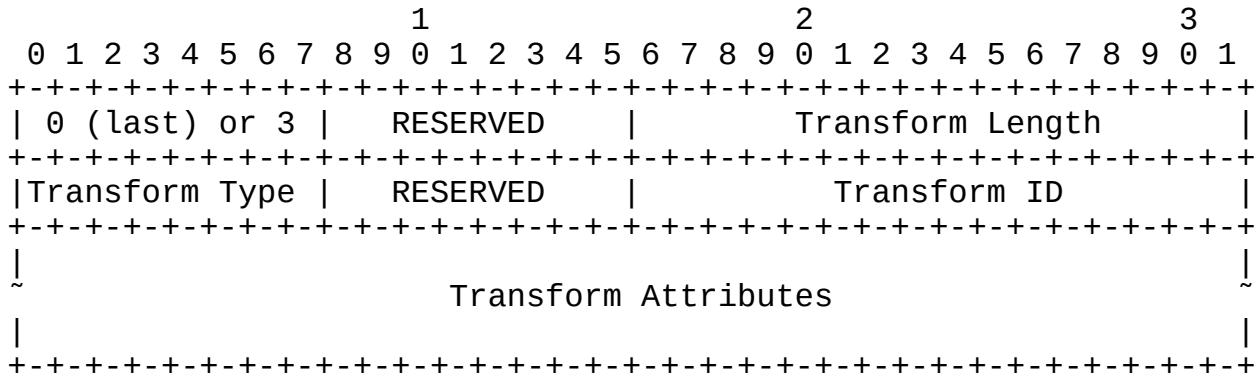

Figure 6: OADD Transform Substructure

- 0 (last) or 3 (more) (1 octet): Specifies whether this is the last Transform Substructure in the Proposal.

- RESERVED (1 octet): MUST be sent as zero; MUST be ignored on receipt.

- Transform Length (2 octets): The length (in octets) of the Transform Substructure including Header and Attributes.

- Transform Type (2 octets): The type of transform being specified in this transform. Set to OADD in this document.

- Transform ID (2 octets): he specific instance of the Transform Type being proposed. Set to INIT or RESP in this document.

- Transform Attributes (variable length): The IP Attribute in this document.

6.2. IP Attribute with IP addresses

This section specifies the Attribute structure as defined in [RFC5996] Section 3.3.5.

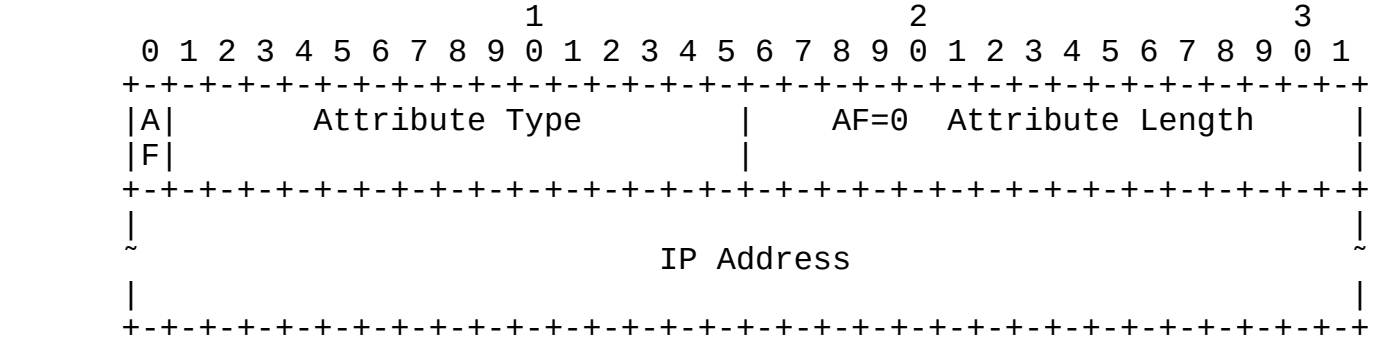

Figure 7: IP Attribute with IP address

- Attribute Format (AF) (1 bit): Set to 0, indicating a TLV format.

- Attribute Type (15 bits): Set to IP in this document.

- Attribute Length (16 bits): The length is either 8 to designate the length of an IPv4 or 20 to designate the length of on IPv6 address. The length includes the headers of 4 octets.

6.3. IP Attribute indicating ANY_IP

This section specifies the Attribute structure as defined in [RFC5996] Section 3.3.5.

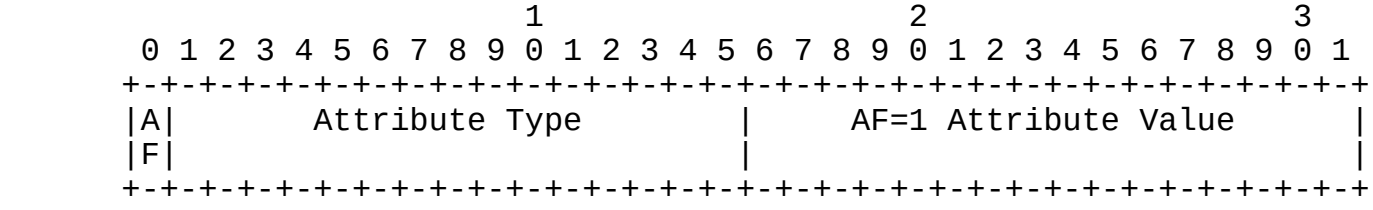

Figure 8: IP Attribute set to ANY_ IP

- Attribute Format (AF) (1 bit): Set to 1, Attribute Value.

- Attribute Value (15 bits): Set to ANY_IP in this document.

6.4. Alternate Outer IP Address Notify Payload

This section presents the Notify Payload as defined in [RFC5996] Section 3.10.

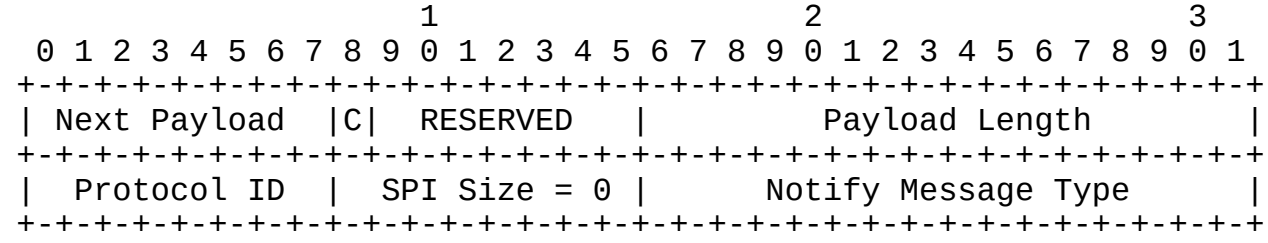

Figure 9: Alternate Outer IP Address Notify Payload

- Next Payload (1 octet): Identifier for the payload type of the next payload in the message. If the current payload is the last in the message, then this field will be 0.

- Critical (1 bit): MUST be set to zero for payload types defined in this document.

- RESERVED (7 bits): MUST be sent as zero; MUST be ignored on receipt.

- Payload Length (2 octets, unsigned integer): Length in octets of the current payload, including the generic payload header. Set to 16 in this document.

- Protocol ID (1 octet): This field MUST be sent as zero and MUST be ignored on receipt.

- Notify Message Type (2 octets): Set to ALTERNATE_OUTER_IP_ADDRESS_SUPPORTED in this document.

7. NAT considerations

In the document we assumed that there were no NAT between the VPN End User and the Security Gateway. This means that the VPN End User and the Security Gateway know 1) the interface they are receiving data on is the interface used as a destination by the other peer and 2) the interface set as destination is the interface used by the other peer to receive the data. As a result, if the VPN End User (respectively the Security Gateway) is behind a NAT, the VPN End User may be seen by the Security Gateway (respectively the VPN End User) with another IP address unknown to the VPN End User (respectively the Security Gateway).

NATs impact the alternate outer IP address extensions in two ways:

- IPsec configuration: The alternate outer IP addresses the two peers are negotiating may not be the ones in the Security Associations. More specifically, suppose the VPN End User and the Security Gateway depicted in figure 10 have negotiated the alternate outer IP addresses src_0, dst_1. src_0 is NATted with NAT_0, and may be unreachable, the outer IP address in the Security Gateway SA should be src_nat_0 instead.

- NAT traversal: NATs may make an IP address behind it reachable only if this IP address has initiated a connection. More specifically, suppose the VPN End User and the Security Gateway depicted in figure 10 have established an IKEv2 channel between src_0 and dst_1 and are MOBIKE enabled. Suppose the VPN End User sends the Security Gateway an ADDITIONAL_IP*_ADDRESS with src_1 or eventually with src_nat_1. Unless NAT_1 has been configured to forward the traffic from the Security Gateway to the VPN End User, this traffic will most likely be discarded by NAT_1. Similarly, if the Security Gateway moves the VPN from dst_0 to dst_1, the VPN may be broken. Note that we use MOBIKE to illustrate the problems of reachability through NATs, but these operations are discussed more in depth in [RFC4555].

This section does not intend to discussed all NATs configuration as described in [RFC5389]. Instead the only NAT scenario we consider is a single NAT and the VPN End User behind that NAT initiates the alternate outer IP address exchange. The architecture this section considers is depicted in figure 10. Furthermore, this section does not consider the NAT traversal aspect. We assume that the VPN End User is NAT aware and perform the necessary actions to make/configure the NATs so that they do not block the traffic.

Section 7.1 defines how the End User MAY prohibit the alternate outer IP address extension if a NAT is detected. Then, in Section 7.2 how the VPN End User can detect the presence of NAT. Section 7.3 discusses the case where the VPN End User does not know the values of the NATted IP addresses and Section 7.4 discusses the case where the VPN End User knows all NATted IP addresses values.

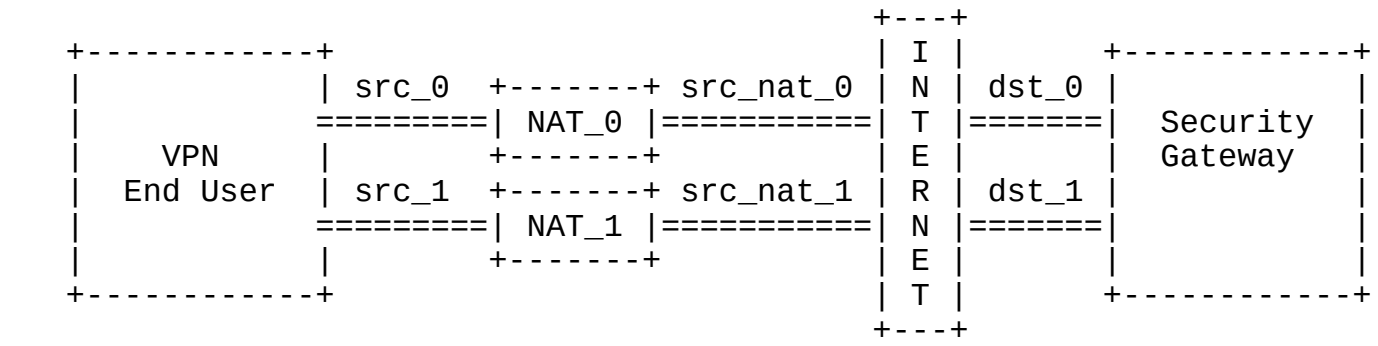

Figure 10: VPN End User behind a NAT scenario

7.1. Prohibiting NAT

This section considers that the VPN End User does not want to use the alternate outer IP address extension if a NAT is detected. This section differs from the NAT detection because it both detects the existence of a NAT and provide an indication that some supported functionalities like MOBIKE SHOULD NOT be used if a NAT is detected.

The NO_NATS_ALLOWED Notify Payload is defined in [RFC4555]. If the VPN End User supports MOBIKE, it MAY send a NO_NATS_ALLOWED Notify Payload with the original IP addresses and ports. When the Notify Payload is received by the Security Gateway, it checks the IP addresses values in the IP header and in the Payload, in case of mismatch, a UNEXPECTED_NAT_DETECTED Notify Payload is returned.

In our case, the NO_NATS_ALLOWED MAY be used by the VPN End User if both the VPN End User and the Security Gateway support MOBIKE. When the Security Gateway receives the NO_NATS_ALLOWED Notify Payload, it MUST NOT use MOBIKE and SHOULD NOT use the alternate outer IP address extension.

There are corner cases that are not considered by this policy. First, a VPN End User or a Security Gateway that do not support MOBIKE cannot use the NO_NATS_ALLOWED Notify Payload. However, it seems hardly possible that peers supporting the alternate outer IP address extension support MOBIKE. Second, a VPN End User using the NO_NATS_ALLOWED applies the same policy for MOBIKE and the alternate outer address extension. Here again, it seems unlikely that NAT policies differ. Furthermore, the NO_NATS_ALLOWED exchange only prevent the Security Gateway to initiate a MOBIKE or alternate outer IP address negotiation. The VPN End User can still use one or the other extension. From our experience, this constraint seems acceptable.

7.2. NAT detection

This section details how NAT can be detailed with IKEv2 extensions. We do not consider here other mechanisms like ICE described in [RFC5768] or STUN [RFC5389].

The VPN End User can detect the NAT by using the NAT_DETECTION_SOURCE_IP and NAT_DETECTION_DESTINATION_IP Notify Payload as described in [RFC5996]. These Notify Payloads carry the SHA-1 of the source (respectively the destination) IP address. At the reception, the Security Gateway can compare their contend with the SHA-1 of the IP addresses in the IP header. A mismatch between the two values indicates the presence of a NAT, but do not provide the value of the original IP address. Usually, this exchange is performed during the IKE_INIT exchange to decide whether or not IKEv2 should proceed to UDP encapsulation.

Note that with the NAT detection exchange, the NAT is detected on the IKEv2 channel. If the IKEv2 channel is using src_0, the NAT detection exchange will detect NAT_0. To detect NAT_1 using IKEv2, the VPN End User SHOULD move the IKEv2 channel on src_1 with MOBIKE for example. Since the UPDATE_SA Notify Payload is initiated by the VPN End User, NAT_1 is expected to accept the traffic from the Security Gateway. Note also that the NAT detection exchange does not provide the value of the src_nat* IP addresses.

7.3. The VPN End User does not know the NATted IP addresses

This section analyses how the alternate outer IP address extension can be used when the VPN End User does not know the values of the NATted IP addresses, i.e. src_nat_0 and src_nat_1.

In that case, the VPN End User MAY only select the destination outer IP address corresponding to the Security Gateway IP addresses. How the VPN End User gets these IP addresses is out of scope of the document, however, if the VPN End User and the Security Gateway support MOBIKE, the MOBIKE ADDITIONAL_IP*_ADDRESS Notify Payload MAY be used for that purpose. It is recommended that the VPN End User does not provide the outer source IP, in which case, the one from the IKEv2 channel will be considered by default. More specifically, the VPN End User cannot provide the Security Gateway its alternate IP addresses.

The VPN End User MAY use the ANY_IP IP Attribute for the source outer IP address. This would enable the Security Gateway to select an alternate IP address that differs from the one used by the IKEv2 channel. In order to select the IP addresses associated to the VPN End User, the Security Gateway has to be aware of the NATted IP

addresses depicted as src_nat_0 and src_nat_1. One possibility is that the Security Gateway log the IP addresses used by the VPN End User when it moves from src_0 to src_1. This also means that the VPN is being negotiated with a CREATE_CHILD_SA exchange after the initial IKE_INIT exchange.

7.4. The VPN End User does know the NATted IP addresses

In this section the VPN End User knows the NATted IP addresses src_nat_0 and src_nat_1. How the End User get these values is out of scope of the document. This case should be considered only if the VPN End User exactly know what it is doing.

In this case, the VPN End User can proceed as if no NAT exist. The VPN End User considers in the alternate outer IP address negotiation that its IP addresses are the NATted IP addresses that is src_nat_0 and src_nat_1. On the other hand, the VPN End User MUST configure properly its SAs with src_0 if src_nat_0 is selected or with src_1 if src_nat_1 is selected.

The VPN End User is also responsible to make the NAT Traversal possible.

8. IANA Considerations

The new fields and number are the following:

IKEv2 Notify Message Types - Status Types

ALTERNATE_OUTER_IP_ADDRESS_SUPPORTED	TBD

Transform Attribute Types

OADD	TBD

Transform Type OADD IDs

INIT	TBD
RESP	TBD

Attribute Type

IP	TBD

IP Attribute Type Values

ANY_IP	TBD

9. Security Considerations

The exchange described in this document is protected by the IKEv2 channel.

10. Acknowledgment

The author would like to thank Yoav Nir for its helpful comments.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4555] Eronen, P., "IKEv2 Mobility and Multihoming Protocol (MOBIKE)", RFC 4555, June 2006.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[RFC5768] Rosenberg, J., "Indicating Support for Interactive Connectivity Establishment (ICE) in the Session Initiation Protocol (SIP)", RFC 5768, April 2010.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

[RFC6027] Nir, Y., "IPsec Cluster Problem Statement", RFC 6027, October 2010.

11.2. Informational References

[I-D.arora-ipsecme-ikev2-alt-tunnel-addresses] Arora, J. and P. Kumar, "Alternate Tunnel Addresses for IKEv2", draft-arora-ipsecme-ikev2-alt-tunnel-addresses-00 (work in progress), April 2010.

[I-D.mglt-ipsecme-security-gateway-discovery]
Migault, D. and K. Pentikousis, "IKEv2 Security Gateway Discovery", draft-mglt-ipsecme-security-gateway-discovery-00 (work in progress), February 2013.

[I-D.mglt-mif-security-requirements]
Migault, D. and C. Williams, "IPsec Multiple Interfaces Problem Statement", draft-mglt-mif-security-requirements-03 (work in progress), November 2012.

Appendix A. Document Change Log

[RFC Editor: This section is to be removed before publication]

-00: First version published.

Author's Address

Daniel Migault
Francetelecom - Orange
38 rue du General Leclerc
92794 Issy-les-Moulineaux Cedex 9
France

Phone: +33 1 45 29 60 52
Email: mglt.ietf@gmail.com