IPSECME
Internet-Draft
Intended status: Standards Track
Expires: September 14, 2014

D. Migault (Ed)
Orange
V. Smyslov
ELVIS-PLUS

March 13, 2014

Clone IKE SA Extension
draft-mglt-ipsecme-clone-ike-sa-01.txt

Abstract

This document considers a VPN End User setting a VPN with a security gateway where at least one of the peer has multiple interfaces.

With the current IKEv2, the outer IP addresses of the VPN are determined by those used by IKEv2 channel. As a result using multiple interfaces requires to set an IKEv2 channel on each interface, or on each paths if both the VPN Client and the security gateway have multiple interfaces. Setting multiple IKEv2 channel involves multiple authentications which may each require multiple round trips and delay the VPN establishment. In addition multiple authentications unnecessarily increase load to the VPN client and the authentication infrastructure.

This document presents the Clone IKE SA extension, where an additional IKEv2 channel is derived from an already authenticated IKEv2 channel. The newly created IKEv2 channel is set without the IKEv2 authentication exchange. The newly created IKEv2 channel can then be assigned to another interface using MOBIKE.

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 September 14, 2014.

Table of Contents

1. Requirements notation 2
2. Introduction 3
3. Terminology 5
4. Protocol Overview 5
5. Protocol Details 6
5.1. Support Negotiation 6
5.2. Cloning IKE SA 6
5.3. Error Handling 7
6. Payload Description 8
7. IANA Considerations 8
8. Security Considerations 9
9. Acknowledgments 9
10. References 10
10.1. Normative References 10
10.2. Informational References 10
Appendix A. Document Change Log 10
Appendix B. Setting a VPN on Multiple Interfaces 11
B.1. Setting VPN_0 11
B.2. Creating an additional IKEv2 Channel 13
B.3. Creation of the Child SA for VPN_1 13
B.4. Moving VPN_1 on Interface_1 14
Authors' Addresses 15

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

The main scenario that motivated this document is a VPN End User setting its VPN with a Security Gateway, and at least one of the peers has multiple interfaces. Figure 1 represents the case where the VPN End User has multiple interfaces, Figure 2 represents the case where the Security Gateway has multiple interfaces, and Figure 3 represents the case where both the VPN End User and the Security Gateway have multiple interfaces. With Figure 1 and Figure 2, one of the peer has n = 2 interfaces and the other has a single interface. This results in the creating of up to n = 2 VPNs. With Figure 3, the VPN End User has n = 2 interfaces and the Security Gateway has m = 2 interfaces. This may lead to up to m x n VPNs.

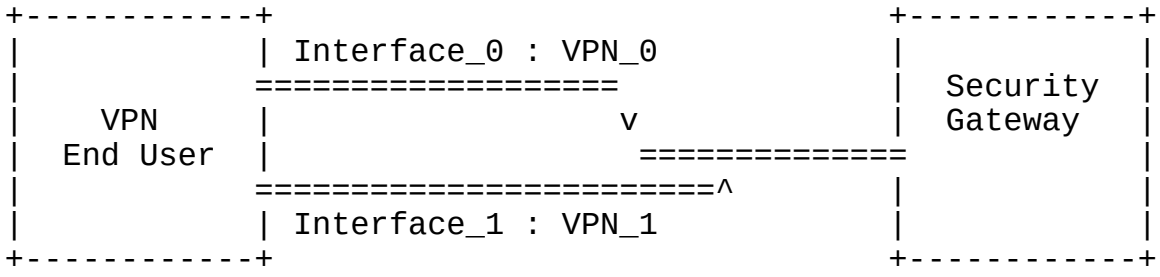

Figure 1: VPN End User with Multiple Interfaces

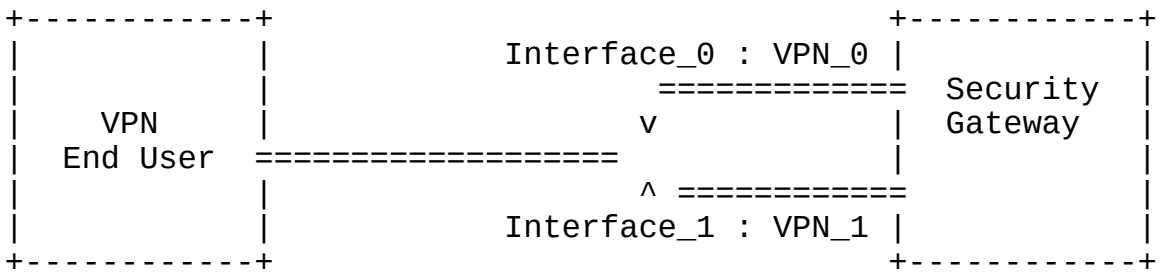

Figure 2: Security Gateway with Multiple Interfaces

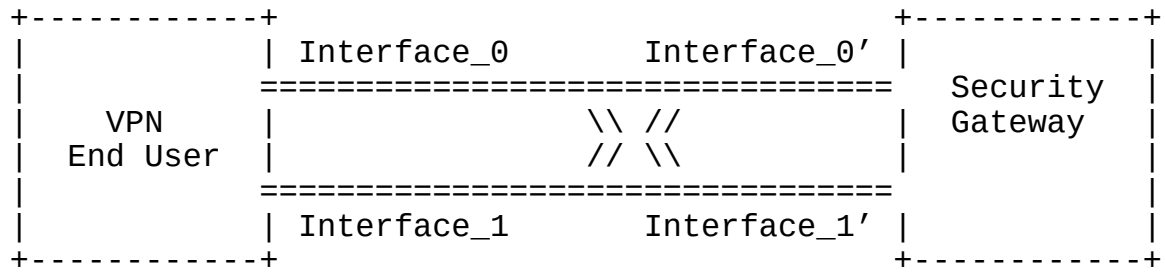

Figure 3: VPN End User and Security Gateway with Multiple Interfaces

With the current IKEv2 [RFC5996], each VPN requires an IKEv2 channel, and setting an IKEv2 channel requires an authentication. Authentication may involve multiple round trips like EAP-SIM [RFC4186] as well as crypto operations that may delay the connectivity.

This document presents the Clone IKE SA extension. The main idea is that the peer with multiple interfaces sets the first authenticated IKEv2 channel. Then it takes advantage of this authentication and derives as many parallel IKEv2 channels as the number of VPNs. On each IKEv2 channel a VPN is negotiated. This results in parallel VPNs. Then the VPN End User moves the VPNs to their proper places using MOBIKE [RFC4555]. Alternatively, the VPN End User may first move the IKEv2 channels and then negotiate the VPNs.

Several documents have addressed the issue of IPsec and multiple interfaces. [I-D.mglt-mif-security-requirements] provides a problem statement for IPsec and multiple interfaces. [I-D.arora-ipsecme-ikev2-alt-tunnel-addresses] and [I-D.mglt-ipsecme-alternate-outer-address] have been proposed to allow tunnel outer IP addresses to differ from those of the IKEv2 channel.

The advantage of the Clone IKE SA extension is that it requires very few modifications to already existing IKEv2 implementations. Then, it reuses already existing and widely deployed protocol MOBIKE [RFC4555]. Finally by keeping a dedicated IKEv2 channel for each VPN, it eases reachability tests and VPN maintenance.

Note also that the Clone IKE SA extension is independent from MOBIKE and MAY also address other future scenarios.

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 one Security Gateway to another.

- 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.

- IKE SA: The IKE SA (IKE Security Association) is defined in [RFC5996].

4. Protocol Overview

The goal of the document is to specify how to create a new IKEv2 channel without performing authentication. In order to achieve this goal, the document proposes that the two peers agree they support the Clone IKE SA extension. This is done during the IKE_AUTH exchange using CLONE_IKE_SA_SUPPORTED Notify Payload. To create a new parallel IKE SA, one of the peers initiates a CREATE_CHILD_SA exchange as if it would rekey the IKE SA. In order to indicate the current IKE SA MUST NOT be deleted, the initiator includes a CLONE_IKE_SA Notify Payload in the CREATE_CHILD_SA exchange. This results in two parallel IKE SA.

IKEv2 [RFC5996] specifies the CREATE_CHILD_SA exchange that makes possible to rekey an IKE SA, create or rekey a new Child SA. The difference between rekeying an IKE SA and creating a new IKE SA is that the old IKE SA must not be deleted. Deleting of the current IKE SA can be done either by sending a Delete Payload or be an implementation design of IKEv2.

Note that IKEv2 [RFC5996] Section 1.3.2 and Section 2.18 do not explicitly mention that the old IKE SA must be deleted. However, there are currently no signaling advertising that the IKE SA must not be deleted. The purpose of this document is to avoid this uncertainty when rekeying the IKE SA. In other words, the document avoids the situation when one peer expects an additional IKE SA to be created whereas the other simply proceeds to a replacement of the old IKE SA.

Currently, one may check whether or not the old IKE SA has been deleted by waiting a for some time and then initiating an empty

INFORMATIONAL exchange using the old IKE SA. The absence of response will indicate that the old IKE SA has been removed.

5. Protocol Details

5.1. Support Negotiation

The initiator and the responder indicate their support for the Clone IKE SA extension by exchanging the CLONE_IKE SA_SUPPORTED Notifications. This notification MUST be sent in the IKE_AUTH exchange (in case of multiple IKE_AUTH exchanges, in the message containing the SA payload). If both initiator and responder send this notification during IKE_AUTH exchange, peers MAY use the Clone IKE SA extension, explicitly specifying when an IKE SA is being rekeyed, if the IKE SA has to be be cloned, or may be deleted. In the other case the Clone IKE SA extension MUST NOT be used.

```
Initiator                          Responder
-------------------------------------------------------------------
HDR, SAi1, KEi, Ni -->
                                <-- HDR, SAr1, KEr, Nr
HDR, SK { IDi, CERT, AUTH,
                CP(CFG_REQUEST),
                SAi2, TSi, TSr,
                N(CLONE_IKE_SA_SUPPORTED) }
                                <-- HDR, SK { IDr, CERT, AUTH,
                                    CP(CFG_REPLY), SAr2, TSi, TSr,
                                     N(CLONE_IKE_SA_SUPPORTED) }
```

5.2. Cloning IKE SA

The initiator of the rekey exchange sends the CLONE_IKE_SA Notification in a CREATE_CHILD_SA request for rekeying the IKE SA. The CLONE_IKE_SA Notification indicates that the current IKE SA MUST NOT be deleted. Instead two parallel IKEv2 channels are expected to coexist. The current IKE SA becomes the old IKE SA and the newly negotiated IKE SA becomes the new IKE SA. Peers MUST NOT send CLONE_IKE_SA (and MUST ignore it if the other party sends it) if support for the Clone IKE SA extension wasn’t previously negotiated in IKE_AUTH exchange. The CLONE_IKE_SA Notification MUST appear only in request message of CREATE_CHILD_SA exchange concerning IKE SA rekey. If the CLONE_IKE_SA Notification appears in any other message, it MUST be ignored.

```
Initiator                          Responder
-------------------------------------------------------------------
HDR, SK { N(CLONE_IKE_SA), SA, Ni, KEi } -->
```

If the CREATE_CHILD_SA request concerns an IKE SA rekey and contains CLONE_IKE_SA Notification, the Responder proceeds to the IKE SA rekey, creates the new IKE SA, and keeps the old IKE SA. No additional Notify Payload is included in the CREATE_CHILD_SA response as represented below:

<-- HDR, SK { SA, Nr, KEr }

When using Clone IKE SA Extension peers MUST NOT transfer existing Child SAs, that were created by old IKE SA, to newly created IKE SA. So, all signalling messages, concerning those Child SAs MUST continue to be send over old IKE SA. This is different from regular IKE SA rekey.

5.3. Error Handling

There may be conditions when responder for some reason is unable or unwilling to perform IKE SA cloning. This inability may be temporary or permanent.

Temporary inability occurs when responder doesn’t have enough resources at the moment to clone IKE SA or when IKE SA is being deleted by responder. In this case responder SHOULD reject request to clone IKE SA with TEMPORARY_FAILURE notification.

<-- HDR, SK { N(TEMPORARY_FAILURE) }

After receiving this notification initiator MAY retry its request after waiting some period of time. See Section 2.25 of [RFC5996] for details.

In some cases responder may have restrictions on the number of co-existing IKE SAs with one peer. These restrictions may be either implicit (some devices may have enough resources to handle only a few IKE SAs) or explicit (provided by some configuration parameter). If initiator wants to clone more IKE SAs, than responder is able or is configured to handle, the responder SHOULD reject the request with NO_ADDITIONAL_SAS notification.

<-- HDR, SK { N(NO_ADDITIONAL_SAS) }

This condition is considered permanent and initiator SHOULD NOT retry to clone IKE SA until some of existing IKE SAs with the responder are deleted.

6. Payload Description

Figure 4 illustrates the Notify Payload packet format as described in section 3. 10 of [RFC5996]. This is the format we use for both the CLONE_IKE_SA or CLONE_IKE_SA_SUPPORTED notifications.

The CLONE_IKE_SA_SUPPORTED Notify Payload is used in an IKEv2 exchange of type IKE_AUTH and the CLONE_IKE_SA is used in an IKEv2 exchange of type CREATE_CHILD_SA.

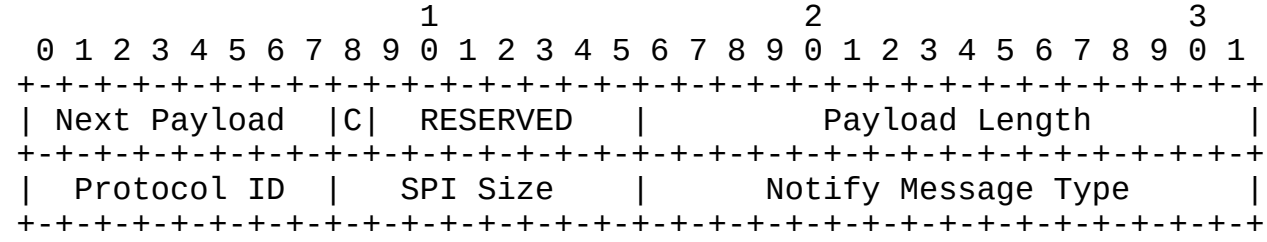

Figure 4: Notify Payload

- Next Payload (1 octet): Indicates the type of payload that follows after the header.

- Critical Bit (1 bit): Indicates how the responder handles the Notify Payload. As notify payload is mandatory to support in IKEv2, the Critical Bit is not set.

- RESERVED (7 bits): MUST be set to zero; MUST be ignored on receipt.

- Payload Length (2 octet): Length in octets of the current payload, including the generic payload header.

- Protocol ID (1 octet): set to zero.

- SPI Size (1 octet): set to zero.

- Notify Message Type (2 octets): Specifies the type of notification message. It is set to <TBA by IANA> for CLONE_IKE_SA notification or to <TBA by IANA> for CLONE_IKE_SA_SUPPORTED Notification.

7. IANA Considerations

IANA is requested to allocate two values in IKEv2 Notify Message Types - Status Types registry:

IKEv2 Notify Message Types - Status Types	
CLONE_IKE_SA_SUPPORTED	- TBA
CLONE_IKE_SA	- TBA

8. Security Considerations

The protocol defined in this document does not modify IKEv2. Security considerations for Clone IKE SA extension are mostly the same as those for base IKEv2 protocol described in [RFC5996].

This extension provides the ability for an initiator to clone existing IKE SAs. As a result it may influence any accounting or control mechanisms based on a single IKE SA per authentication.

Suppose a system has a limit on the number of IKE SAs it can handle. In this case, the Clone IKE SA extension may provide a way for resource exhaustion, as a single end user may populate multiple IKE SAs.

Suppose a system shares the IPsec resources by limiting the number of Child SAs per IKE SA. With a single IKE SA per end user, this provides an equal resource sharing. The Clone IKE SA provides means for a end user to overpass this limit. Such system should evaluate the number of Child SAs over the number of all IKE SAs associated to an end user.

Note, that these issues are not unique for Clone IKE SA extensions, as multiple IKE SAs between two peers may be created without this extension. Note also, that implementation can always limit the number of cloned IKE SAs.

Suppose VPN or any other IPsec based service monitoring is based on the liveliness of the first IKE SA. Such system considers a service is accessed or used from the time IKE performs an authentication to the time the IKE SA is deleted. Such accounting methods were fine as any IKE SA required an authentication exchange. As the Clone IKE SA skips the authentication phase, Clone IKE SA may make possible to delete the initial IKE SA while the service is being used on the cloned IKE SA. Such accounting method should consider the service is being used from the first IKE SA establishment to until the last IKE SA is being removed.

9. Acknowledgments

The ideas of this draft came from various inputs from the ipsecme and discussions with Tero Kivinen and Michael Richardson. Yaron Sheffer,

Tero Kivinen provided significant inputs to set the current design of the protocol as well as its designation.

10. References

10.1. Normative References

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

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

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

10.2. 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-alternate-outer-address] Migault, D., "IKEv2 Alternate Outer IP Address Extension", draft-mglt-ipsecme-alternate-outer-address-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.

[RFC4186] Haverinen, H. and J. Salowey, "Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM)", RFC 4186, January 2006.

Appendix A. Document Change Log

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

-01: Valery Smyslov is now a co-author.

1. Exchange of CLONE_IKE_SA_SUPPORTED notifications made limited to IKE_AUTH exchange only.

2. Some clarifications about processing CLONE_IKE_SA notification are added.

3. Some words that with Clone IKE SA existing Child SAs must not be transferred to newly created IKE SA (unlike regular rekey) are added.

4. Reduced exchanges (combined IKE_AUTH with cloning IKE SA and CREATE_CHILD_SA with transferring to different IPs) are removed.

5. Error handling while cloning IKE SA is described.

6. Clarification text thanks to Tero’s comments

7. Section Security Considerations enhanced with Tero’s suggestions.

8. NO_ADDITIONAL_SAS is added in the error handling section.

-00: Comments from Valery Smyslov, Tero Kivinen and Yaron Sheffer. SUPPORTED Notify Payload can be placed in a INFORMATIONAL or IKE_AUTH exchange. CLONE_IKE_SA is sent in a CREATE_CHILD_SA exchange and is provided both in the query and in the response.

-00: First version published. draft-mglt-ipsecme-keep-old-ike-sa-00

Appendix B. Setting a VPN on Multiple Interfaces

This section is informational and exposes how a VPN End User as illustrated in Figure 1 can builds two VPNs on its two interfaces without multiple authentications. Other cases represented in Figure 2 and Figure 3 are similar and can be easily derived from this case. The mechanism is based on the CLONE_IKE_SA extension and the MOBIKE extension [RFC4555].

B.1. Setting VPN_0

First, the VPN End User negotiates a VPN using one interface. This involves a regular IKEv2 exchanges. In addition, the VPN End User and the Security Gateway advertise their support for MOBIKE. At the end of the IKE_AUTH exchange, VPN_0 is set as represented in Figure 5.

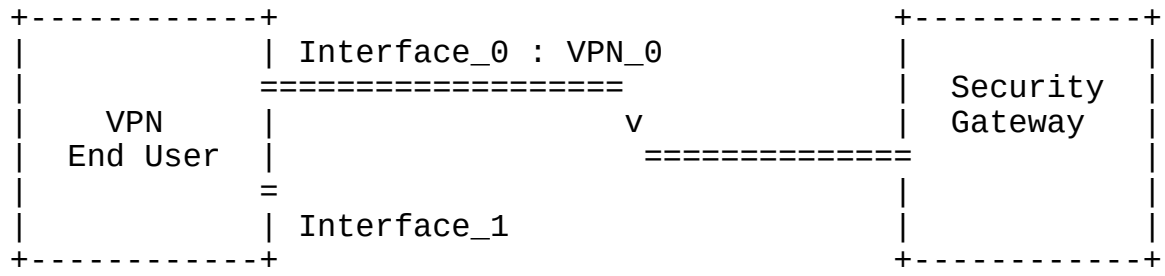

Figure 5: VPN End User Establishing VPN_0

The exchanges are completely described in [RFC5996] and [RFC4555]. First, peers negotiate IKE SA parameters and exchange nonces and public keys in IKE_SA_INIT exchange. In the figure below they also proceed to NAT detection because of the use of MOBIKE.

```
Initiator                          Responder
-------------------------------------------------------------------
(IP_I0:500 -> IP_R:500)
HDR, SAi1, KEi, Ni,
     N(NAT_DETECTION_SOURCE_IP),
     N(NAT_DETECTION_DESTINATION_IP)  -->

                     <--  (IP_R:500 -> IP_I0:500)
                          HDR, SAr1, KEr, Nr,
                               N(NAT_DETECTION_SOURCE_IP),
                               N(NAT_DETECTION_DESTINATION_IP)
```

Then the initiator and the responder proceed to the IKE_AUTH exchange, advertise their support for MOBIKE and for the Clone IKE SA extension - with the MOBIKE_SUPPORTED and the CLONE_IKE_SA_SUPPORTED Notifications - and negotiate the Child SA for VPN_0. Optionally, the initiator and the Security Gateway MAY advertise their multiple interfaces using the ADDITIONAL_IP4_ADDRESS and/or ADDITIONAL_IP6_ADDRESS Notify Payload.

```
(IP_I0:4500 -> IP_R:4500)
HDR, SK { IDi, CERT, AUTH,
          CP(CFG_REQUEST),
          SAi2, TSi, TSr,
          N(CLONE_IKE_SA_SUPPORTED)
          N(MOBIKE_SUPPORTED),
          N(ADDITIONAL_IP*_ADDRESS)+ }  -->

                    <--  (IP_R:4500 -> IP_I0:4500)
                         HDR, SK { IDr, CERT, AUTH,
                                   CP(CFG_REPLY),
                                   SAr2, TSi, TSr,
                                   N(CLONE_IKE_SA_SUPPORTED)
                                   N(MOBIKE_SUPPORTED),
                                   N(ADDITIONAL_IP*_ADDRESS)+}
```

B.2. Creating an additional IKEv2 Channel

In our case the the initiator wants to establish a VPN with its Interface_1 between the VPN End User and the Security Gateway. The VPN End User will first establish a parallel IKE SA using a CREATE_CHILD_SA that concerns an IKE SA rekey associated to a CLONE_IKE_SA Notify Payload. This results in two different IKE SAs between the VPN End User and the Security Gateway. Currently both IKE SAs are set using Interface 0 of the VPN End User.

```
Initiator                          Responder
-------------------------------------------------------------------
(IP_I0:4500 -> IP_R:4500)
HDR, SK { N(CLONE_IKE_SA),
          SA, Ni, KEi} -->
                    <--  (IP_R:4500 -> IP_I0:4500)
                         HDR, SK { N(CLONE_IKE_SA),
                                   SA, Nr, KEr}
```

B.3. Creation of the Child SA for VPN_1

Once the new IKEv2 channel has been created, the VPN End User MAY initiate a CREATE_CHILD_SA exchange that concerns the creation of a Child SA for VPN_1. The newly created VPN_1 will use Interface_0 of the VPN End User.

It is out of scope of the document to define how the VPN End User handles traffic with multiple interfaces. The VPN End User MAY use the same IP inner address on its multiple interfaces. In this case, the same Traffic Selectors (that is the IP address used for VPN_0 and VPN_1) MAY match for both VPNs VPN_0 and VPN_1. The end user VPN SHOULD be aware of such match and be able to manage it. It MAY for

example use distinct Traffic Selectors on both VPNs using different ports, manage the order of its SPD or have SPD defined per interfaces. Defining these mechanisms are out of scope of this document. Alternatively, the VPN End User MAY use a different IP address for each interface. In the latter case, if the inner IP address is assigned by the Security Gateway, the Configuration Payload (CP) MUST be placed before the SA Payload as specified in [RFC5996] Section 2.19.

The creation of VPN_1 is performed via the newly created IKE SA as follows:

```
Initiator                          Responder
-------------------------------------------------------------------
(IP_I0:4500 -> IP_R:4500)
HDR(new), SK(new) { [CP(CFG_REQUEST)],
          SAi2, TSi, TSr }  -->

                       <--  (IP_R:4500 -> IP_I0:4500)
                            HDR(new), SK(new) { [CP(CFG_REPLY)],
                                      SAr2, TSi, TSr}
```

The resulting configuration is depicted in Figure 6. VPN_0 and VPN_1 have been created, but both are using the same Interface: Interface_0.

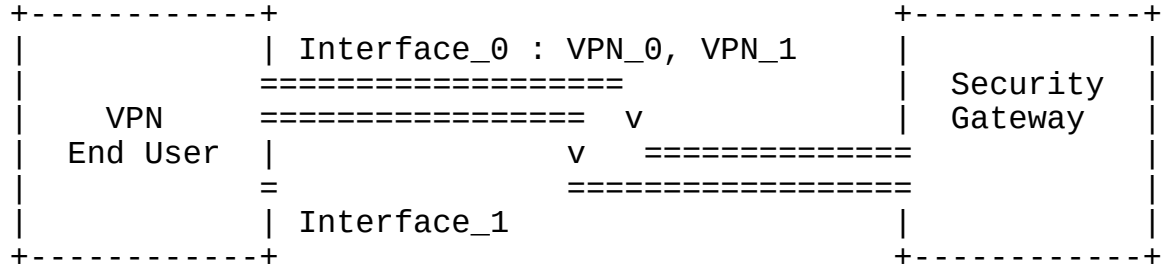

Figure 6: VPN End User Establishing VPN_0 and VPN_1

B.4. Moving VPN_1 on Interface_1

In this section, MOBIKE is used to move VPN_1 on interface_1. The exchange is described in [RFC4555]. All exchanges use the new IKE SA. Eventually, the VPN End User MAY check if the Security Gateway is reachable via Interface_1. The exchanges are described below:

```
   Initiator                          Responder
   -------------------------------------------------------------------
   (IP_I1:4500 -> IP_R:4500)
   HDR(new), SK(new) { N(NAT_DETECTION_SOURCE_IP),
             N(NAT_DETECTION_DESTINATION_IP) }

                         <--  (IP_R:4500 -> IP_I1:4500)
                              HDR(new), SK(new) {
                                   N(NAT_DETECTION_SOURCE_IP),
                                   N(NAT_DETECTION_DESTINATION_IP) }
```

After that initiator requests the peer to switch to new addresses.

```
   (IP_I1:4500 -> IP_R:4500)
   HDR(new), SK(new) { N(UPDATE_SA_ADDRESSES),
             N(NAT_DETECTION_SOURCE_IP),
             N(NAT_DETECTION_DESTINATION_IP),
             N(COOKIE2) }  -->

                         <--  (IP_R:4500 -> IP_I1:4500)
                              HDR(new), SK(new) {
                                   N(NAT_DETECTION_SOURCE_IP),
                                   N(NAT_DETECTION_DESTINATION_IP),
                                   N(COOKIE2) }
```

This results in the situation as described in Figure 7.

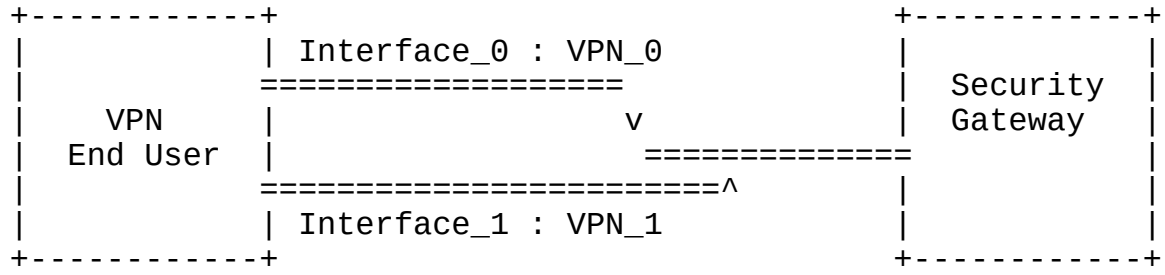

Figure 7: VPN End User with Multiple Interfaces

Authors' Addresses

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

Phone: +33 1 45 29 60 52
Email: daniel.migault@orange.com

Valery Smyslov
ELVIS-PLUS
PO Box 81
Moscow (Zelenograd) 124460
Russian Federation

Phone: +7 495 276 0211
Email: svan@elvis.ru