Internet Engineering Task Force (IETF)
Request for Comments: 7791
Category: Standards Track
ISSN: 2070-1721

D. Migault, Ed.
Ericsson
V. Smyslov
ELVIS-PLUS

March 2016

Cloning the IKE Security Association in the Internet Key Exchange Protocol Version 2 (IKEv2)

Abstract

This document considers a VPN end user establishing an IPsec Security Association (SA) with a Security Gateway using the Internet Key Exchange Protocol version 2 (IKEv2), where at least one of the peers has multiple interfaces or where Security Gateway is a cluster with each node having its own IP address.

The protocol described allows a peer to clone an IKEv2 SA, where an additional SA is derived from an existing one. The newly created IKE SA is set without the IKEv2 authentication exchange. This IKE SA can later be assigned to another interface or moved to another cluster node.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7791.

Table of Contents

1. Introduction 3
2. Requirements Notation 5
3. Terminology 5
4. Protocol Overview 6
5. Protocol Details 6
5.1. Support Negotiation 6
5.2. Cloning the IKE SA 7
5.3. Error Handling 7
6. Payload Description 8
7. IANA Considerations 9
8. Security Considerations 9
9. References 10
9.1. Normative References 10
9.2. Informative References 10
Appendix A. Setting a VPN on Multiple Interfaces 11
A.1. Setting VPN_0 11
A.2. Creating an Additional IKE SA 12
A.3. Creating the Child SA for VPN_1 12
A.4. Moving VPN_1 on Interface_1 13
Acknowledgments 14
Authors' Addresses 14

1. Introduction

The main scenario that motivated this document is a VPN end user establishing a VPN with a Security Gateway when at least one of the peers has multiple interfaces. Figure 1 represents the case when the VPN end user has multiple interfaces, Figure 2 represents the case when the Security Gateway has multiple interfaces, and Figure 3 represents the case when both the VPN end user and the Security Gateway have multiple interfaces. With Figure 1 and Figure 2, one of the peers has n = 2 interfaces and the other has a single interface. This results in the creation 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.

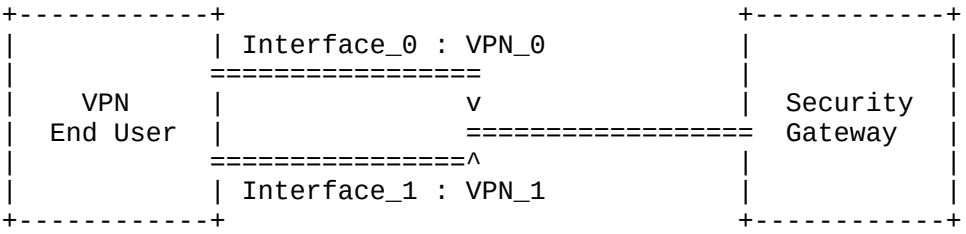

Figure 1: VPN End User with Multiple Interfaces

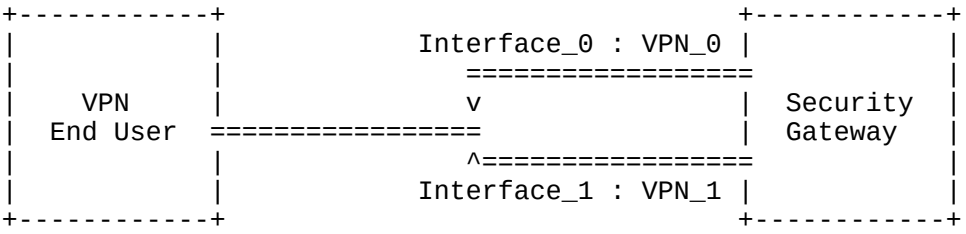

Figure 2: Security Gateway with Multiple Interfaces

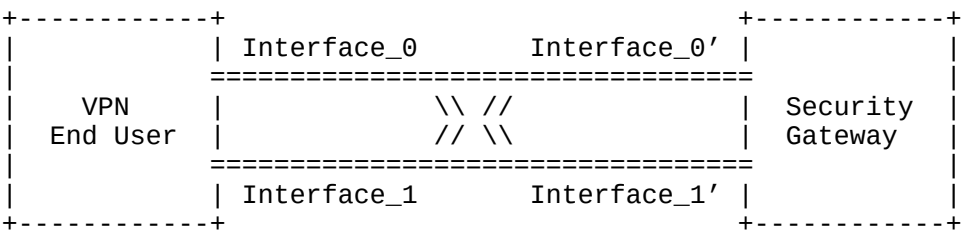

Figure 3: VPN End User and Security Gateway with Multiple Interfaces

With the current IKEv2 protocol [RFC7296], each VPN requires an IKE SA, and setting an IKE SA requires an authentication. Authentication might require multiple round trips and an activity from the end user (like EAP-SIM [RFC4186] or EAP-TLS [RFC5216]) as well as crypto operations that would introduce an additional delay.

Another scenario is a load-balancing solution. Load-sharing clusters are often built to be transparent for VPN end users. In the case of IPsec, this means that IKE and IPsec SA states are duplicated on every cluster node where the load balancer can redirect packets. The drawback of such an approach is that anti-replay related data (in particular, Sequence Number) must be reliably synchronized between participating nodes per every outgoing Authentication Header (AH) or Encapsulating Security Payload (ESP) packet, which makes building high-speed systems problematic. Another approach for building load-balancing systems is to make VPN end users aware of them, which allows for having two or more Security Gateways sharing the same ID, but each having its own IP address. In this case, the VPN end user first establishes an IKE SA with one of these gateways. Then, at some point of time the gateway makes a decision to move the client to a different cluster node. This can be done with Redirect Mechanism for IKEv2 [RFC5685]. The drawback of such an approach is that it requires a new IKE SA to be established from scratch, including full authentication. In some cases, this could be avoided by using IKEv2 Session Resumption [RFC5723] with a new gateway. However, this requires the VPN end user to know beforehand which new gateway to connect to. So, it is desirable to be able to clone the existing IKE SA, move it to a different Security Gateway, and then indicate to the VPN end user to use this new SA. This would allow participating Security Gateways to share the load between them.

This document introduces the possibility of cloning the IKE SA in the Internet Key Exchange Protocol Version 2 (IKEv2). The main idea is that the peer with multiple interfaces sets the first IKE SA as usual. Then it takes advantage of the fact that this SA is completed and derives as many new parallel IKE SAs from it as the desired number of VPNs. On each IKE SA a VPN is negotiated by creating one or more IPsec SAs. This results in coexisting parallel VPNs. Then the VPN end user moves each IPsec SA to its proper location using the IKEv2 Mobility and Multihoming Protocol (MOBIKE) [RFC4555]. Alternatively, the VPN end user may first move the IKE SAs and then create the IPsec SAs.

Note that it is up to the host’s local policy to decide which additional VPNs to create and when to do it. The process of selecting address pairs for migration is a local matter.

Furthermore, in the case of multiple interfaces on both ends, care should be taken to avoid the VPNs being duplicated by both ends or moved to both interfaces.

In addition, multiple MOBIKE operations may be involved from the Security Gateway or the VPN end user. Suppose, as depicted in Figure 3 for example, that the cloned VPN is between Interface _0 and Interface_0', and the VPN end user and the Security Gateway want to move it to Interface_1 and Interface_1'. The VPN end user may initiate a MOBIKE exchange in order to move it to Interface_1, in which case the cloned VPN is now between Interface_1 and Interface_0'. Then the Security Gateway may also initiate a MOBIKE exchange in order to move the VPN to Interface_1', in which case the VPN has reached its final destination.

The combination of the IKE SA cloning with MOBIKE protocol provides IPsec communications with multiple interfaces the following advantages. First, cloning the IKE SA requires very few modifications to already existing IKEv2 implementations. Then, it takes advantage of the already existing and widely deployed MOBIKE protocol. Finally, it keeps a dedicated IKE SA for each VPN, which simplifies reachability tests and VPN maintenance.

Note also that the cloning of the IKE SA is independent from MOBIKE and can also address other future scenarios not described in the current document.

2. 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].

3. Terminology

This section defines terms and acronyms used in this document.

- VPN: Virtual Private Network -- one or more Child (IPsec) SAs created in tunnel mode between two peers.

- VPN End User: designates the end user that initiates the VPN with a Security Gateway. This end user may be mobile and move 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: IKE Security Association as defined in [RFC7296].

4. Protocol Overview

This document specifies how to clone existing IKE SAs without performing new authentication. In order to achieve this goal, this document proposes that the two peers agree upon their ability to clone the IKE SA. This is done during the IKE_AUTH exchange by exchanging the CLONE_IKE_SA_SUPPORTED notifications. To create a new parallel IKE SA, one of the peers initiates a CREATE_CHILD_SA exchange as if it would rekey the existing IKE SA. In order to indicate that the current IKE SA must not be deleted, the initiator includes the CLONE_IKE_SA notification in the CREATE_CHILD_SA exchange. This results in two parallel IKE SAs.

Note that without the CLONE_IKE_SA notification, the old IKE SA would be deleted after the rekey is successfully completed (as specified in Section 2.8 of [RFC7296].

5. Protocol Details

5.1. Support Negotiation

The initiator and the responder indicate their support for cloning IKE SA 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 first IKE_AUTH message from initiator and in the last IKE_AUTH message from responder). If both initiator and responder send this notification during the IKE_AUTH exchange, peers may clone this IKE SA. In the other case, the IKE SA MUST NOT be cloned.

```
Initiator                          Responder
-------------------------------------------------------------------
HDR, SA, KEi, Ni -->
                                <-- HDR, SA, KEr, Nr
HDR, SK {IDi, AUTH,
     SA, TSi, TSr,
     N(CLONE_IKE_SA_SUPPORTED)} -->
                                <-- HDR, SK {IDr, AUTH,
                                         SA, TSi, TSr,
                                         N(CLONE_IKE_SA_SUPPORTED)}
```

5.2. Cloning the IKE SA

The initiator of the rekey exchange includes 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 will not be immediately deleted once the new IKE SA is created. Instead, two parallel IKE SAs are expected to coexist. The current IKE SA becomes the old IKE SA and the newly negotiated IKE SA becomes the new IKE SA. The CLONE_IKE_SA notification MUST appear only in the request message of the CREATE_CHILD_SA exchange concerning the 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 is concerned with an IKE SA rekey and contains the 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 Payloads are included in the CREATE_CHILD_SA response as represented below:

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

When the IKE SA is cloned, peers MUST NOT transfer existing Child SAs that were created by the old IKE SA to the newly created IKE SA. So, all signaling messages concerning those Child SAs would continue to be sent over the old IKE SA. This is different from the regular IKE SA rekey in IKEv2.

5.3. Error Handling

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

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

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

After receiving this notification, the initiator MAY retry its request after waiting some period of time. See Section 2.25 of [RFC7296] for details.

In some cases, the responder may have restrictions on the number of coexisting 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 the initiator wants to clone more IKE SAs than the responder is able or is configured to handle, the responder SHOULD reject the request with the NO_ADDITIONAL_SAS notification.

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

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

6. Payload Description

Figure 4 illustrates the Notify Payload packet format as described in Section 3.10 of [RFC7296]. This format is used for both the CLONE_IKE_SA and the CLONE_IKE_SA_SUPPORTED notifications.

The CLONE_IKE_SA_SUPPORTED notification 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.

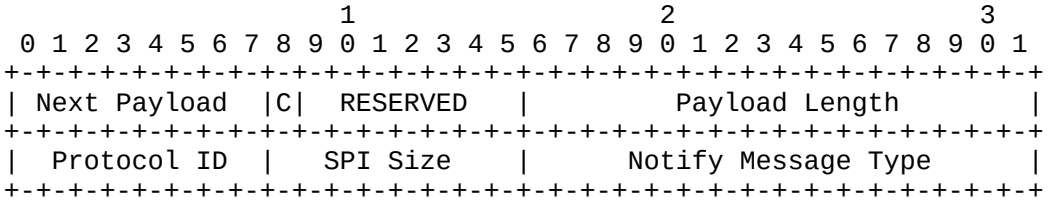

Figure 4: Notify Payload

The fields Next Payload, Critical Bit, RESERVED, and Payload Length are defined in [RFC7296]. Specific fields defined in this document are:

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

- Security Parameter Index (SPI) Size (1 octet): Set to zero.

- Notify Message Type (2 octets): Specifies the type of notification message. It is set to 16432 for the CLONE_IKE_SA_SUPPORTED notification or 16433 for the CLONE_IKE_SA notification.

7. IANA Considerations

IANA has allocated two values in the "IKEv2 Notify Message Types - Status Types" registry:

Value	Notify Messages - Status Types
16432	CLONE_IKE_SA_SUPPORTED
16433	CLONE_IKE_SA

8. Security Considerations

The protocol defined in this document does not modify IKEv2. Security considerations for cloning an IKE SA are mostly the same as those for the base IKEv2 protocol described in [RFC7296].

Cloning an IKE SA allows an initiator to duplicate existing 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, cloning an IKE SA 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. In this case, cloning the IKE SA provides the means for an end user to overpass this limit. Such a 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 to the ability of cloning the IKE SA, as multiple IKE SAs between two peers may be created without involving a cloning method. Note also that implementation can always limit the number of cloned IKE SAs.

Suppose the VPN or any other IPsec-based service monitoring is based on the liveliness of the first IKE SA. Such a 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 cloning the IKE SA skips the authentication phase, it may make it possible to delete the initial IKE SA while the service is being used on the cloned IKE SA. Such accounting methods should consider that the service is being used from the first IKE SA establishment to until the last IKE SA is removed.

When this solution is used to build load-balancing systems, then there is a necessity to transfer IKE SA states between nodes of a load-sharing cluster. Since IKE SA state contains sensitive information, such as session keys, implementations must take all due precautions. Such precautions might include using technical and/or administrative means to protect IKE SA state data. The details of what is transferred and how it is protected are out of scope of this document.

9. References

9.1. Normative References

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

[RFC4555] Eronen, P., "IKEv2 Mobility and Multihoming Protocol (MOBIKE)", RFC 4555, DOI 10.17487/RFC4555, June 2006, <http://www.rfc-editor.org/info/rfc4555>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <http://www.rfc-editor.org/info/rfc7296>.

9.2. Informative References

[RFC4186] Haverinen, H., Ed. and J. Salowey, Ed., "Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM)", RFC 4186, DOI 10.17487/RFC4186, January 2006, <http://www.rfc-editor.org/info/rfc4186>.

[RFC5216] Simon, D., Aboba, B., and R. Hurst, "The EAP-TLS Authentication Protocol", RFC 5216, DOI 10.17487/RFC5216, March 2008, <http://www.rfc-editor.org/info/rfc5216>.

[RFC5685] Devarapalli, V. and K. Weniger, "Redirect Mechanism for the Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5685, DOI 10.17487/RFC5685, November 2009, <http://www.rfc-editor.org/info/rfc5685>.

[RFC5723] Sheffer, Y. and H. Tschofenig, "Internet Key Exchange Protocol Version 2 (IKEv2) Session Resumption", RFC 5723, DOI 10.17487/RFC5723, January 2010, <http://www.rfc-editor.org/info/rfc5723>.

Appendix A. Setting a VPN on Multiple Interfaces

This section is informational and exposes how a VPN end user, as illustrated in Figure 1, can build 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 cloning the IKE SA and the MOBIKE extension [RFC4555].

A.1. Setting VPN_0

First, the VPN end user negotiates a VPN using one interface. This involves 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.

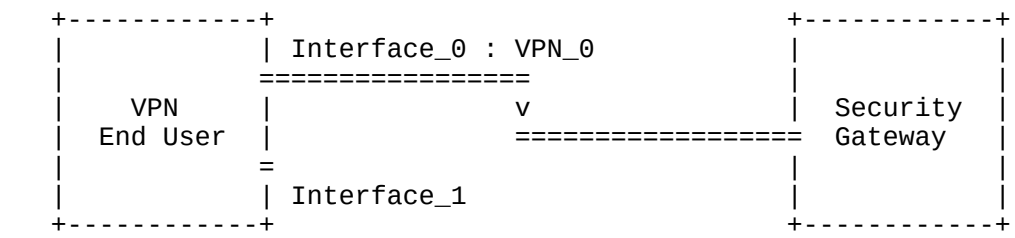

Figure 5: VPN End User Establishing VPN_0

The exchanges are completely described in [RFC7296] and [RFC4555]. First, peers negotiate IKE SA parameters and exchange nonces and public keys in the 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, SA, KEi, Ni,
     N(NAT_DETECTION_SOURCE_IP),
     N(NAT_DETECTION_DESTINATION_IP)  -->

                       <--  (IP_R:500 -> IP_I0:500)
                            HDR, SA, 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 their ability to clone the IKE SA -- with the MOBIKE_SUPPORTED and the CLONE_IKE_SA_SUPPORTED notifications -- and negotiate the Child SA

for VPN_0. Optionally, the initiator and the responder can advertise their multiple interfaces using the ADDITIONAL_IP4_ADDRESS and/or ADDITIONAL_IP6_ADDRESS notifications.

```
(IP_I0:4500 -> IP_R:4500)
HDR, SK {IDi, AUTH,
     SA, TSi, TSr,
     N(MOBIKE_SUPPORTED),
     [N(ADDITIONAL_IP*_ADDRESS)+,]
     N(CLONE_IKE_SA_SUPPORTED)}  -->

                        <--  (IP_R:4500 -> IP_I0:4500)
                             HDR, SK {IDr, AUTH,
                                  SA, TSi, TSr,
                                  N(MOBIKE_SUPPORTED),
                                  [N(ADDITIONAL_IP*_ADDRESS)+,]
                                  N(CLONE_IKE_SA_SUPPORTED)}
```

A.2. Creating an Additional IKE SA

In this case, the VPN end user wants to establish an additional VPN with its Interface_1. The VPN end user will first establish a parallel IKE SA using a CREATE_CHILD_SA that concerns an IKE SA rekey associated with a CLONE_IKE_SA notification. This results in two separate 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 {SA, Nr, KEr}
```

A.3. Creating the Child SA for VPN_1

Once the new IKE SA has been created, the VPN end user can 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 for this document to define how the VPN end user handles traffic with multiple interfaces. The VPN end user can use the same inner IP address on its multiple interfaces. In this case, the same Traffic Selectors (that is, the IP address used for VPN_0 and VPN_1) can match for both VPNs VPN_0 and VPN_1. The VPN end user must be aware of such a match and be able to manage it. It can, for

example, use distinct Traffic Selectors on both VPNs using different ports, manage the order of its Security Policy Database (SPD), or have SPD defined per interfaces. Defining these mechanisms is out of scope for this document. Alternatively, the VPN end user can use a different inner IP address for each interface.

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) {SA, TSi, TSr}  -->

                       <--  (IP_R:4500 -> IP_I0:4500)
                            HDR(new), SK(new) {SA, 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.

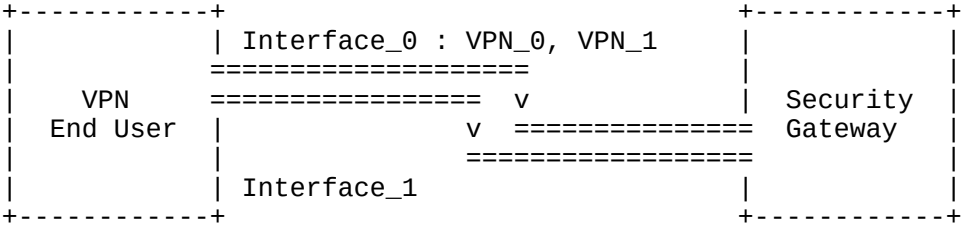

Figure 6: VPN End User Establishing VPN_0 and VPN_1

A.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].

```
(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.

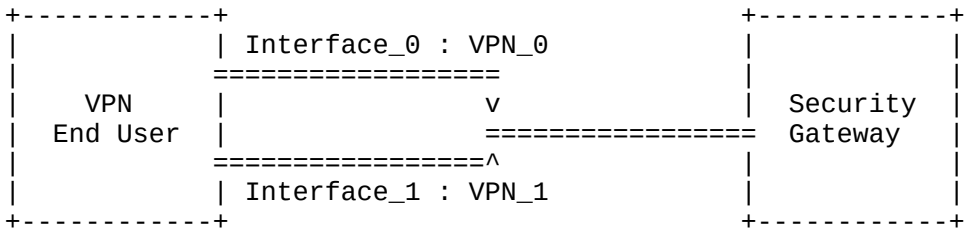

Figure 7: VPN End User with Multiple Interfaces

Acknowledgments

The ideas for this document came from various input from the IP Security Maintenance and Extensions (ipsecme) Working Group and from discussions with Tero Kivinen and Michael Richardson. Yaron Sheffer and Tero Kivinen provided significant input to set the current design of the protocol, as well as its designation.

Authors' Addresses

Daniel Migault (editor)
Ericsson
8400 boulevard Decarie
Montreal, QC H4P 2N2
Canada

Email: daniel.migault@ericsson.com

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

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