

Clone IKE SA Extension
draft-mglt-ipsecme-clone-ike-sa-00.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 load 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 August 17, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Requirements notation	2
2.	Introduction	3
3.	Terminology	4
4.	Protocol Overview	5
5.	Payload Description	6
6.	Protocol Description	7
6.1.	CLONE_IKE_SA_SUPPORTED Notify Payload	7
6.2.	CLONE_IKE_SA Notify Payload	8
7.	IANA Considerations	8
8.	Security Considerations	8
9.	Acknowledgment	8
10.	References	8
10.1.	Normative References	8
10.2.	Informational References	9
Appendix A.	Document Change Log	9
Appendix B.	Setting a VPN on Multiple Interfaces	9
B.1.	Setting VPN_0	10
B.2.	Creating an additional IKEv2 Channel	11
B.3.	Creation of the Child SA for VPN_1	11
B.4.	Moving VPN_1 on Interface_1	12
B.5.	Reduced Exchange	13
	Author's Address	14

[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 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 has 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 can lead to up to $m \times n$ VPNs.

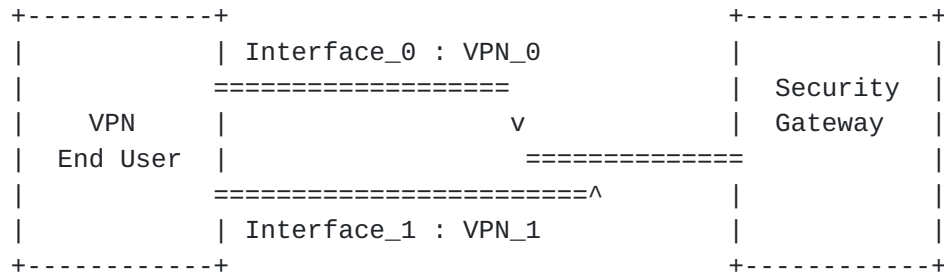


Figure 1: VPN End User with Multiple Interfaces

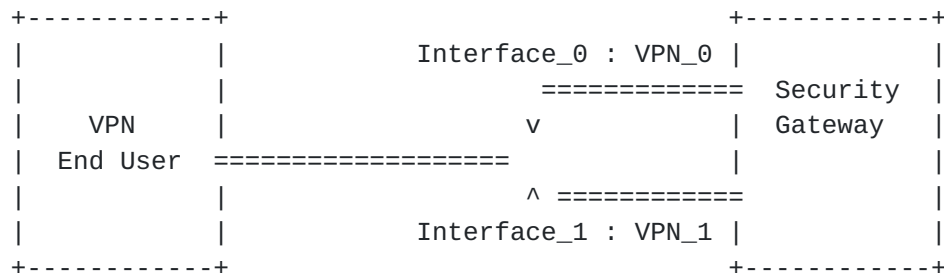


Figure 2: Security Gateway with Multiple Interfaces

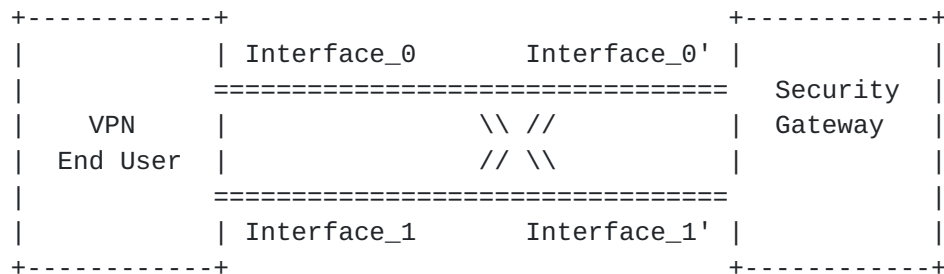


Figure3: VPN End User and Security Gateway
with Multiple Interfaces

With the current IKEv2 [[RFC5996](#)], each VPN requires an IKEv2 channel, and setting an IKEv2 channel requires an authentication. Authentication can 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 an first authenticated IKEv2 channel. Then it takes advantage of this authentication and derives as many parallel IKEv2 channels as 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. Alternatively, the VPN End User can also 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 so tunnel outer IP address can 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 such as MOBIKE [[RFC4555](#)]. Finally by keeping a dedicated IKEv2 channel for each VPN, it eases reachability tests.

Note also that the Clone IKE SA extension is independent of 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 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. Protocol Overview

The goal of the document is to specify how to create a new IKEv2 channel. IKEv2 [[RFC5996](#)] specifies the CREATE_CHILD_SA 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, either by starting a Delete exchange or removing the IKE_SA without the Delete exchange.

Note that IKEv2 [[RFC5996](#)] [Section 1.3.2](#) or [Section 2.18](#) does not explicitly mentions that the old IKE_SA MUST be deleted. However, there are currently no signaling advertising the IKE_SA has not been deleted. The purpose of this document is to avoid this uncertainty when rekeying the IKE_SA. In other words, the document avoids that one peer expects a 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 or not by waiting a for a given time and then initiate and empty INFORMATIONAL exchange using the old IKE_SA. The absence of response MAY indicate the old IKE_SA has been removed.

The initiator and the responder indicate they support the Clone IKE SA extension with CLONE_IKE_SA_SUPPORTED Notify Payload. These Notify Payloads can be sent at any time after the IKE_SA has been negotiated. In the example below, the CLONE_IKE_SA_SUPPORTED exchange is performed during the IKEv2 negotiation. The initiator and the responder support the Clone IKE SA extension, which means both peers can explicitly specify, when a IKE_SA is rekeyed, if the IKE SA MUST be cloned, or MAY be removed. The CLONE_IKE_SA_SUPPORTED Notify Payload can be sent in IKE_AUTH or INFORMATIONAL IKEv2 exchange.

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) }

The initiator of the rekey exchange sends the CLONE_IKE_SA Notify Payload in a CREATE_CHILD_SA request for rekeying the IKE_SA. The

CLONE_IKE_SA Notify Payload indicates the current IKE_SA MUST NOT be deleted. Instead two parallel IKEv2 channel are expected to coexist. The current IKE_SA becomes the old IKE_SA and the newly negotiated IKE_SA becomes the new IKE_SA. If the Initiator does not want or does not care that two parallel IKE SA exists, the CLONE_IKE_SA Notify Payload SHOULD be omitted. The CLONE_IKE_SA Notify Payload is always part of a CREATE_CHILD_SA IKEv2 exchange.

Initiator

Responder

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

The responder supports the `CLONE_IKE_SA` Notify Payload as it provided a `CLONE_IKE_SA_SUPPORTED` Notify Payload. If the `CREATE_CHILD_SA` request concerns a `IKE_SA` rekey. The responder MUST proceed to the `IKE_SA` rekey, create the new `IKE_SA`, and keep the old `IKE_SA` and respond with a `CLONE_IKE_SA` Notify Payload as represented below:

```
<-- HDR, SK { N(CLONE_IKE_SA)
           SA, Nr, KEr}
```

If the CLONE_IKE_SA Notify Payload is not associated to a IKE_SA rekey, the responder MUST return an INVALID_SYNTAX Notification as described in [section 3.10.1 of \[RFC5996\]](#). The exchange will be:

```
<-- HDR, SK {SA, Nr, KEr
          N(INVALID_SYNTAX)}
```

5. Payload Description

Figure 7 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 Notify Payload.

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

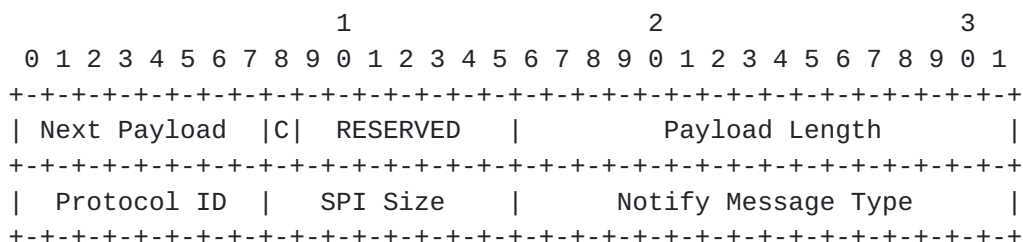


Figure 7: Notify Payload

- 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. In this document the Critical Bit is not set.
- RESERVED (7 bits): MUST be set as 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 CLONE_IKE_SA or CLONE_IKE_SA_SUPPORTED in this document.

6. Protocol Description

6.1. CLONE_IKE_SA_SUPPORTED Notify Payload

The CLONE_IKE_SA_SUPPORTED Notify Payload is sent by the initiator of the INFORMATIONAL or IKE_AUTH exchange to announce its support of the Clone IKE SA extension.

If the CLONE_IKE_SA_SUPPORTED is not sent in a message of type INFORMATIONAL or IKE_AUTH, the responder SHOULD send an INVALID_SYNTAX Notify Payload.

Upon reception of the CLONE_IKE_SA_SUPPORTED Notify Payload, the responder that supports the Clone IKE SA extension SHOULD send a CLONE_IKE_SA_SUPPORTED Notify Payload as a response. This indicates the initiator the responder also supports the Clone IKE SA extension. A responder that does not support the Clone IKE SA extension MUST ignore the CLONE_IKE_SA_SUPPORTED Notify Payload as specified in [\[RFC5996\]](#).

The Clone IKE SA extension is considered supported by both peers if and only if the initiator and the responder have sent and received a CLONE_IKE_SA_SUPPORTED Notify Payload. In any other case the extension is considered not supported and SHOULD NOT be used in latter exchanges.

6.2. CLONE_IKE_SA Notify Payload

The CLONE_IKE_SA Notify Payload SHOULD be used only if the Clone IKE SA extension is supported by the two peers.

The CLONE_IKE_SA Notify Payload MUST always be sent in a CREATE_CHILD_SA message that concerns an IKE_SA rekey as described in [section 1.3.2 of \[RFC5996\]](#). If not, a INVALID_SYNTAX Notify Payload MUST be sent.

Upon reception of a CLONE_IKE_SA Notify Payload from the responder, the initiator got the confirmation two parallel IKE_SA have been created on the responder.

7. IANA Considerations

The new fields and number are the following:

IKEv2 Notify Message Types - Status Types

CLONE_IKE_SA - TBD

CLONE_IKE_SA_SUPPORTED - TBD

8. Security Considerations

The protocol defined in this document does not modify IKEv2. It signals what has been implementation dependent on how to manage an old IKE_SA after a rekey.

9. Acknowledgment

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

[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.

- [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]

-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 build two VPNs on its two interfaces without multiple authentications. Other cases represented in figure 2 and 3 are similar and can be easily derived from the 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 setting. In addition, the VPN End User and the Security Gateway advertise they support MOBIKE. At the end of the exchange, VPN_0 is set as represented in figure 4.

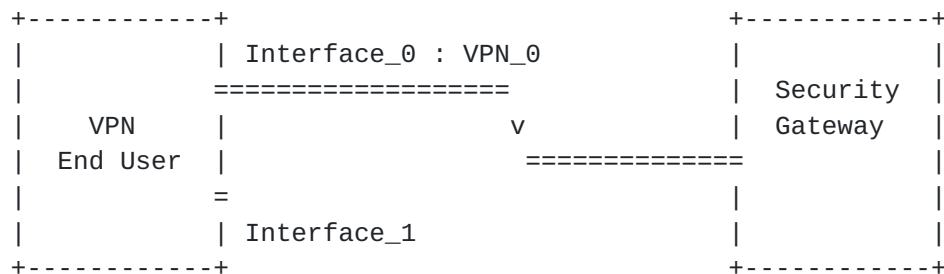


Figure 4: VPN End User Establishing VPN_0

The exchange is completely described in [RFC4555]. First the negotiates the IKE_SA. In the figure below peers also proceed to NAT detection because of the use of MOBIKE.

```

Initiator                                Responder
-----
(IP_I1:500 -> IP_R1:500)
HDR, SAi1, KEi, Ni,
  N(NAT_DETECTION_SOURCE_IP),
  N(NAT_DETECTION_DESTINATION_IP) -->

<-- (IP_R1:500 -> IP_I1:500)
    HDR, SAR1, KEr, Nr,
      N(NAT_DETECTION_SOURCE_IP),
      N(NAT_DETECTION_DESTINATION_IP)

```

The initiators and the responder proceed to the authentication exchange, advertise they support MOBIKE and the Clone IKE SA extension - with the MOBIKE_SUPPORTED and the CLONE_IKE_SA_SUPPORTED Notify Payloads - and negotiate the 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_I1:4500 -> IP_R1:4500)
HDR, SK { IDi, CERT, AUTH,
          CP(CFG_REQUEST),
          SAI2, TSi, TSr,
          N(CLONE_IKE_SA_SUPPORTED)
          N(MOBIKE_SUPPORTED),
          N(ADDITIONAL_IP*_ADDRESS)+ } -->

<-- (IP_R1:4500 -> IP_I1:4500)
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 set 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_SA between the VPN End User and the Security Gateway. Currently both IKE_SA are set using Interface 0 of the VPN End User.

Initiator	Responder

<pre> (IP_I1:4500 -> IP_R1:4500) HDR, SK { N(CLONE_IKE_SA), SA, Ni, KEi} --> </pre>	<pre> <-- (IP_R1:4500 -> IP_I1:4500) HDR, SK { N(CLONE_IKE_SA), SA, Nr, KEr} </pre>

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_I1:4500 -> IP_R1:4500)
HDR(new), SK(new) { [CP(CFG_REQUEST)],
                    SAI2, TSi, TSr } -->

<-- (IP_R1:4500 -> IP_I1:4500)
    HDR(new), SK(new) { [CP(CFG_REPLY)],
                        SAr2, TSi, TSr}

```

The resulting configuration is depicted in figure 5. VPN_0 and VPN_1 have been created, but both are using the same Interface: Interface_0.

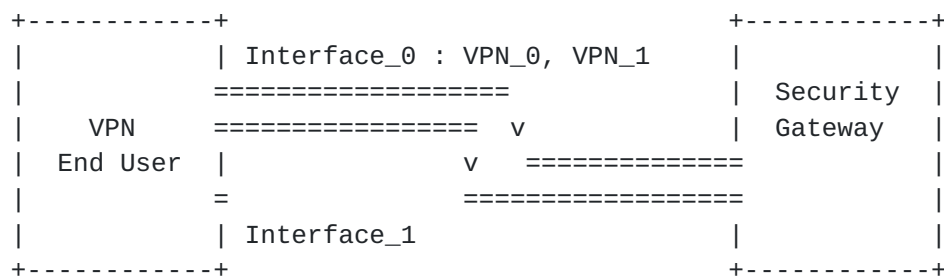


Figure 5: VPN End User Establishing VPN_0 and VPN_1

[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 are using 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_I2:4500 -> IP_R1:4500)
HDR(new), SK(new) { N(NAT_DETECTION_SOURCE_IP),
                    N(NAT_DETECTION_DESTINATION_IP) }

<-- (IP_R2:4500 -> IP_I1:4500)
    HDR(new), SK(new) {
        N(NAT_DETECTION_SOURCE_IP),
        N(NAT_DETECTION_DESTINATION_IP) }

(This worked, and the initiator requests the peer to switch to new
addresses.)

(IP_I2:4500 -> IP_R1:4500)
HDR(new), SK(new) { N(UPDATE_SA_ADDRESSES),
                    N(NAT_DETECTION_SOURCE_IP),
                    N(NAT_DETECTION_DESTINATION_IP),
                    N(COOKIE2) } -->

<-- (IP_R1:4500 -> IP_I2:4500)
    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 6.

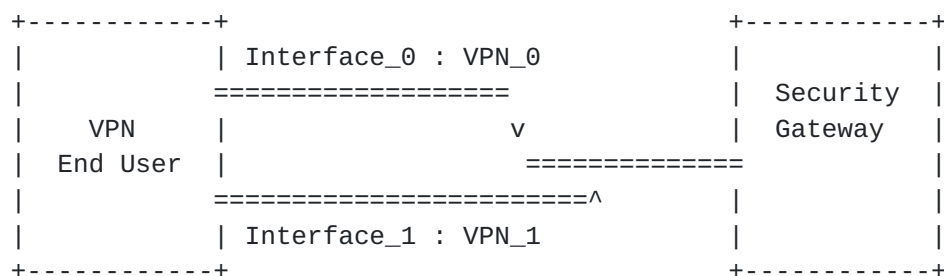


Figure 6: VPN End User with Multiple Interfaces

B.5. Reduced Exchange

The previous sections detail the various exchanges between the VPN End User and the Security Gateway. This section shows an example where the number of exchanges are limited, thus limiting the delay to set up a multiple interface VPN communication.

Initiator

Responder

```

-----

(IP_I1:500 -> IP_R1:500)
HDR, SAI1, KEI, NI,
  N(NAT_DETECTION_SOURCE_IP),
  N(NAT_DETECTION_DESTINATION_IP)  -->

      <-- (IP_R1:500 -> IP_I1:500)
      HDR, SAR1, KER, NR,
        N(NAT_DETECTION_SOURCE_IP),
        N(NAT_DETECTION_DESTINATION_IP)

(IP_I1:4500 -> IP_R1:4500)
HDR, SK { IDI, CERT, AUTH,
  CP(CFG_REQUEST),
  SAI2, TSI, TSR,
  N(CLONE_IKE_SA_SUPPORTED),
  N(MOBIKE_SUPPORTED),
  N(ADDITIONAL_IP*_ADDRESS)+,
  N(CLONE_IKE_SA),
  SA, NI, KEI}  -->

      <-- (IP_R1:4500 -> IP_I1:4500)
      HDR, SK { IDR, CERT, AUTH,
        CP(CFG_REPLY),
        SAR2, TSI, TSR,
        N(CLONE_IKE_SA_SUPPORTED),
        N(MOBIKE_SUPPORTED),
        N(ADDITIONAL_IP*_ADDRESS)+},
        N(CLONE_IKE_SA),
        SA, NR, KER}

      <-- (IP_R1:4500 -> IP_I2:4500)
      HDR(new), SK(new)
        { [CP(REQUEST)],
          SAI2, TSI, TSR,
          N(UPDATE_SA_ADDRESSES)}

(IP_I2:4500 -> IP_R1:4500)  -->
HDR(new), SK(new) { [CP(CFG_REPLY)],
  SAR2, TSI, TSR}

```

Author's Address

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