IPSECME Working Group
Internet Draft
Intended status: Proposed Standard
Expires: December 2013

P. Sathyanarayan(Ed.)
S. Hanna
S. Melam
Juniper Networks
Y. Nir
Check Point
D. Migault
Francetelecom - Orange
K. Pentikousis
EICT

October 21, 2013

Auto Discovery VPN Protocol
draft-sathyanarayan-ipsecme-advpn-03

Abstract

This document defines a protocol for dynamically establishing and tearing down IPsec tunnels as needed without requiring non-scalable configuration.

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at
http://www.ietf.org/shadow.html

This Internet-Draft will expire on April 21, 2014

Copyright Notice

Copyright (c) 2013 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.

Table of Contents

1. Introduction

IPsec [IPSECARCH] is currently being deployed in more diverse network environments which exhibit significantly larger numbers of hosts than we have seen before. For example, IPsec is currently used in a broad set of scenarios ranging from remote office VPN deployments to mobile device access to corporate and other sensitive network resources to securing critical telecommunications infrastructure in cellular networks. Large deployments of IPsec may involve thousands of gateways and endpoints with constantly changing traffic patterns. In order to enable efficient and secure traffic flow in such environments, we need to be able to establish tunnels dynamically, as needed. As a result, static IPsec configuration based on presets is no longer deemed adequate. Users expect to be able to connect remotely and securely without compromising their communications quality of experience. In other words, a more dynamic method of establishing and tearing down Security Associations (SAs) [IPSECARCH] than what is currently possible with current standards is desired. This is discussed in [ADVPNreq] where it is shown that, for a variety of use cases, static configuration does not scale for large systems and that a standardized solution is needed where equipment from different vendors may be involved.

Motivated by the problem defined in [ADVPNreq], this document proposes a protocol that can demonstratively scale in large IPsec deployments while ensuring that routing stretch is minimized and network resources are used more optimally. The proposed protocol extends [IKEV2] to meet the requirements spelled out in [ADVPNreq], providing a standard way to dynamically establish and tear down IPsec tunnels, as needed, without requiring non-scalable configuration. The protocol introduces the concept of a "shortcut" which can be used by compliant IPsec gateways to optimize the traffic path between two peers. The protocol has provisions for adhering to established policies and is applicable to single- and multi-domain environments. Shortcuts can be established and torn down dynamically and, as we show in Appendix A, the proposed solution is applicable to a variety of use cases and scenarios, pertaining to both wired and wireless networks.

The remainder of this document is organized as follows. Section 2 presents our design considerations and discusses the salient protocol characteristics we are after. Section 3 specifies the Auto Discovery VPN Protocol (ADVPN), while Section 4 examines the implications of ADVPN on IPsec policy. Security considerations are discussed in Section 5.

Further, this document includes three appendices: Appendix A details several ADVPN use cases, while Appendix B explains how the proposed protocol meets the requirements set in [ADVPNreq]. Finally, Appendix C provides an illustrative example of how the PROTECTED_DOMAIN response can be created.

1.1. Conventions Used In This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [MUSTSHOULD].

1.2. Terminology

This section defines the terms used throughout this document. The terms introduced in [ADVPNreq] apply here as well. For reader convenience, we repeat in particular the following terms:

Endpoint - A device that implements IPsec for its own traffic but does not act as a gateway.

Gateway - A network device that implements IPsec to protect traffic flowing through the device.

In addition, this document defines the following terms:

Peer - A host (gateway or endpoint) with an IPsec Security Association.

Shortcut - An IPsec Security Association established dynamically (at the suggestion of a shortcut suggester).

SHORTCUT exchange - A new IKEv2 exchange that carries all data needed to establish the shortcut IPsec Security Association.

Shortcut suggester - A peer that initiates a SHORTCUT exchange.

Shortcut partners - The pair of peers that received a SHORTCUT exchange suggesting that they should establish a shortcut.

Shortcut initiator - A peer directed by a SHORTCUT exchange to act as the IKEv2 initiator while establishing a shortcut.

Shortcut responder - A peer directed by a SHORTCUT exchange to act as the IKEv2 responder while establishing a shortcut.

2. Design Considerations

The protocol described in this document aims at operational environments with possibly tens of thousands (or more) peers. Peers may belong to the same administrative domain, or to different administrative domains which have already established trust relationships. In this kind of network environment one wants to minimize configuration effort overall and, to the degree possible, eliminate manual labor in administering route optimization. This may not only result in better network resource utilization, but also in increased network resilience as reliance on a few centrally-located gateways is reduced. In addition, the automation introduced by the protocol described herein enables administrators to optimize IPsec traffic flows at time scales that are simply not possible with today's tools. In general, the protocol should allow for self-optimization as permitted by established domain policies.

Since IPsec traffic may originate or terminate behind NATs and other policy-enforcing gateways, we aim for a protocol that can work well in this environment. In addition, peers are not expected to be stationary. Given the widespread deployment of wireless networks and the proliferation of mobile devices with multiple interfaces it is reasonable to anticipate that some hosts can join the ADVPN from a range of different access points and this is taken into account in our protocol design.

A central aim of the protocol is to enable peers to setup IPsec tunnels without the need for continuous manual configuration. In addition, the establishment of new tunnels should not inadvertently affect other peers, i.e. it should not call for manual configuration elsewhere in the VPN. Moreover, the establishment of new IPsec tunnels should be easily controlled and managed by the administrator. When new tunnels are operational as well as when they are terminated the administrator should be fully aware of it.

With these considerations in mind, we design a system that can function purely on the basis of local optimizations and policies. In short, the ADVPN protocol enables each individual gateway to act as a shortcut suggester (as per administrator configuration), i.e. to recommend a "shortcut" to appropriate peers with which it has previously established IPsec security associations. These peers, which we refer to as the shortcut partners, can accept, reject or ignore this recommendation, according to their own policies. If the partner(s) reject the recommendation, the partner response indicates the reasons for the rejection, so the shortcut suggester can properly optimize its VPN topology. In addition, responses may also carry informational data that may be handled by the shortcut suggester in various ways. This foundation bestows scaling properties to the Auto Discovery VPN protocol, as described in the following sections.

It is important to highlight that the protocol introduced in this document does not require peers to have a comprehensive understanding of the global network topology. Each peer can act in accordance with its own policy. Taken as a whole, this system optimizes the graph of IPsec Security Associations to match the current traffic flow (subject to policy constraints) and then continuously reoptimizes the IPsec tunnel graph as traffic flows and policies change over time. Appendix A provides illustrative examples of such a (re)optimization.

3. Auto Discovery VPN Protocol

The Auto Discovery VPN protocol (ADVPN) enables an IPsec gateway to suggest the establishment of a shortcut, i.e. a direct IPsec tunnel between two of its peers. For example, the shortcut could be used to establish a more optimal path for data delivery.

Whenever an IPsec gateway decides that a shortcut between two of its peers would be beneficial, it initiates a SHORTCUT exchange with both peers, including all information needed to establish the shortcut in the exchange. The peers can reject the SHORTCUT exchange but they can also use the information contained in the exchange to attempt to establish a direct SA between them. We refer to these peers as the shortcut partners. We refer to the gateway offering the shortcut suggestion to both partners as the shortcut suggester.

A shortcut MAY be torn down when it is no longer receiving adequate traffic (as determined by the shortcut partners) or when the timeout for

the shortcut expires. Of course, the shortcut partners MAY decide to explicitly terminate the shortcut at any time.

Note that this protocol works in an exemplary manner in typical hub-and-spoke topologies but it is also well-suited for arbitrary topologies. For example, consider the case of two endpoints exchanging an adequate amount of traffic (as determined by the shortcut suggester) and connected through a series of gateways, all of which support the Auto Discovery VPN protocol. As detailed in Appendix A (Sections A.2. and A.3. , the protocol enables the step-by-step optimization of the traffic flow between two endpoints through the use of shortcut tunnels. The protocol effectively enables direct and secure communication between the two endpoints without any manual configuration involved in setting up the respective tunnels.

3.1. Prerequisites

The Auto Discovery VPN protocol MUST only be used with IKEv2.

Before the Auto Discovery VPN protocol can be used, all participants (i.e. the shortcut suggester and the shortcut partners) must indicate support for this protocol by sending ADVPN_SUPPORTED notification payload as described in Section 3.5. Any IKEv2 peer that sends this notification is indicating that it supports the protocol defined in this draft.

Shortcut partners and shortcut suggesters MUST NOT send any of the messages defined in the remainder of this specification unless the intended recipient of the message has sent the ADVPN_SUPPORTED notification payload during the IKEv2 exchange. Any party that supports this protocol will send this notification payload in the first IKE_AUTH request sent in the IKE exchange. However, it may delay sending this payload until later, for example, if it has a policy that restricts the set of peers with which it is willing to establish a shortcut.

3.2. Shortcut Initiation

Once the use of the Auto Discovery VPN protocol is enabled, an IPsec gateway which acts as a shortcut suggester can decide that two of its peers (which have indicated support for the ADVPN protocol) should establish a direct IPsec Security Association. Note that the decision-making process for selecting the two peers is outside the scope of this document. As an illustrative example, however, one could consider the observation of excessive transit traffic load between said peers. Another reason could be the realization that certain quality of service (QoS)

requirements would be better served through a shortcut. For instance, some of the traffic between the two peers may be delay-sensitive and would benefit from a more direct route. Alternatively, gateway-, policy- and operation-related reasons, such as overload, scheduled maintenance, energy-saving and so on, could also trigger the initiation of a shortcut recommendation. The reasoning behind the trigger that initiates a shortcut exchange to selected peers is beyond the scope of this document.

Once an IPsec gateway has decided that two peers should establish a direct SA, it acts as a shortcut suggester and uses its already established IKEv2 SAs with these peers to initiate a SHORTCUT exchange to each of the shortcut partners. Each SHORTCUT exchange includes most or all of the information needed to allow the shortcut partners to establish their own SA, such as, the IP address, port number and identity of the other partner, an indication of which partner should be the IKEv2 initiator and which should be the responder, and even an optional Pre-Shared Key, which can facilitate partner authentication with each other.

The shortcut suggester MAY also include Traffic Selectors in the SHORTCUT exchange to indicate which traffic should be sent over the shortcut. This allows traffic for certain destinations to use the ADVPN shortcut while traffic for other destinations continues to flow through the gateway (i.e. the shortcut suggester). Further, it allows traffic destined for certain port numbers (e.g. associated to high-volume, delay-sensitive traffic such as video conferencing applications) to follow the path defined by the shortcut, while other types of traffic carrying, for example, sensitive information that ought to be logged or analyzed, continue to be routed through the gateway.

The shortcut partners MAY decline to act on the SHORTCUT exchange. Although the decision to do so is outside the scope of this document, one could consider, for example, that there may be implementation-specific or operational reasons for rejecting the newly received shortcut suggestion. For instance, the shortcut partners may be low on resources or they may have recently tried to establish this shortcut and failed. Another reason for not accepting the shortcut recommendation could be that doing so would violate local policy. For instance, one of the shortcut partners may accept shortcuts only within its organization.

The shortcut partner(s) MAY ignore the SHORTCUT exchange, but it MUST provide a reason for such refusal to the shortcut suggester by including the ADVPN_STATUS notification in the SHROTCUT exchange. We note that a shortcut suggester SHOULD NOT reinitiate a SHORTCUT exchange just because the shortcut partners have not set up the requested shortcut tunnel. An ADVPN_STATUS notification MAY carry a timeout value as an indication by

either of the partners to the shortcut suggester so that the latter defers the re-initiation of a SHORTCUT exchange for this partner pair for the specified amount of time.

The shortcut suggester can indicate, during the SHORTCUT initiation exchange, which shortcut partner should act as the initiator and which as the responder. For example, if only one of the two peers is behind a NAT, the shortcut suggester can indicate the peer behind the NAT as the initiator. Once this decision is made, the shortcut suggester initiates the SHORTCUT exchange with the chosen shortcut responder first. Once the responder's response is received and it indicates acceptance of the suggestion, the shortcut suggester can proceed with the notification to the partner that will act as the shortcut initiator. If, on the other hand, the responder rejects the suggestion, the shortcut suggester MAY change the roles of the partners or terminate the process.

If the shortcut partner identified as the initiator in the SHORTCUT exchange decides to establish the shortcut suggested by the exchange, it will attempt to establish an IKEv2 exchange with its designated shortcut partner (the "shortcut responder") and then to establish an IPsec security association between the two. Ordinarily, the Initiator in an IKE_AUTH exchange MAY include an IDr payload. In an IKE_AUTH exchange established because of a SHORTCUT, both the IDi and IDr payloads are mandatory, and their content MUST agree with the ID payloads in the SHORTCUT exchange. Once the SA between the two partners is established, both shortcut partners SHOULD send to the shortcut suggester a SHORTCUT response, indicating that the shortcut tunnel has been established. Details of how this is done are specified in the descriptions of specific Shortcut Types in Section 3.4.

If the shortcut partners are able to establish an IPsec security association, they can use the Traffic Selectors for this SA to determine which traffic should be sent through this tunnel. Shortcut partners MUST ensure that the Traffic Selectors negotiated for the shortcut tunnel are a subset of the Traffic Selectors they have in place for their SA with the shortcut suggester. Since there may be an overlap between the Traffic Selectors for the shortcut SA and for the SA with the shortcut suggester, preference SHOULD be given in this case to sending traffic over the shortcut SA, as described in Section 4.

If a VPN gateway is performing address translation (NAT) for traffic coming from one peer and going to another peer, then in MUST NOT suggest a shortcut between them. Such traffic would have different addresses when flowing directly between the peers, and there is no guarantee that such

flows work with the IPsec policy and with the routing in the remote network.

3.3. Shortcut Termination

After establishing an IPsec Security Association triggered by a SHORTCUT exchange, as described in the following subsection, either of the shortcut partners may decide to terminate the shortcut. This may occur at any point of time and for a variety of reasons (outside the scope of this document), such as, for example, due to lack of traffic using the shortcut, local policy, shortage of resources, or other reasons. However, the shortcut SA SHOULD NOT be terminated simply because the SA with the shortcut suggester was terminated due to inactivity. On the contrary, dropping the SA with the shortcut suggester while maintaining the shortcut SA may be quite a normal occurrence if the only traffic flowing through the shortcut suggester has now been diverted into the shortcut.

Note that either shortcut partner may terminate a shortcut by closing the corresponding IKE SA (and therefore all child IPsec SAs) by sending an IKEv2 Delete payload to the other shortcut partner, thus indicating that the IKE SA should be deleted.

3.4. Peer Address Identification Payload

The Peer Address Identification Payload, denoted as IDa, contains the address of the peer gateway. It is formatted in the same manner as the IDi and IDr payloads which are defined in Section 3.5 of RFC 5996. The ID type in the IDa payload MUST be from one of the following types:

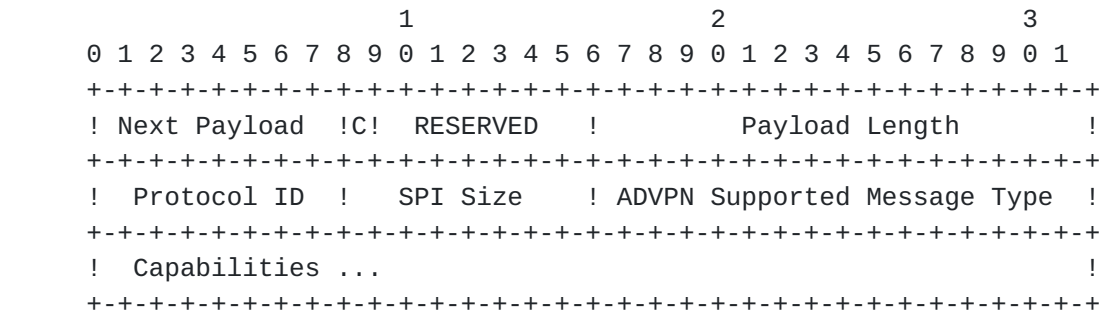
- ID_IPV4_ADDR, indicating an IPv4 address
- ID_IPV6_ADDR, indicating an IPv6 address
- ID_FQDN, indicating a fully qualified domain name; this is allowed if and only if the peer has indicated the capability "FQDN Resolver" in its ADVPN_SUPPORTED notification. See Section 3.5.

This payload is currently defined only for the SHORTCUT exchange. It MUST NOT be sent to any peer that has not indicated support for SHORTCUT exchanges by sending the ADVPN_SUPPORTED notification. The Critical bit MUST be set.

[RFC EDITOR PLEASE REMOVE THIS PARAGRAPH] For development and interoperability testing while this document is still a draft and IANA actions have not taken place, implementations can use the private-use value of 247 for the payload type of the IDa payload.

3.5. ADVPN_SUPPORTED Notification

The Notify payload for announcing support of ADVPN is included in the Initial Exchange and is formatted as follows:



The fields corresponding to the first 4 octets are defined as described in RFC 5996. The remaining fields are defined as follows:

- Protocol ID (1 octet) MUST be zero, as specified in Section 3.10 of RFC 5996.
- SPI Size (1 octet) MUST be zero, in conformance with Section 3.10 of RFC 5996.
- ADVPN Supported Message Type (2 octets) - MUST be xxxxx. [RFC EDITOR NOTE: value assigned by IANA for ADVPN_SUPPORTED]
- The Capabilities field can be two or more octets long, indicating the capabilities that this implementation supports. See the Table below for the capabilities specified in this document. The first of these capabilities MUST indicate the protocol version range (0x01-0x08), and at least one MUST list the features range (0x09-0xff). All version capabilities MUST precede all feature capabilities.

Value	Name	Comment
0x00	pad	Used to pad the notification to any desired length. MAY be sent multiple times at the end of the list, and MUST be ignored on receipt

0x01	v1	The version described in this document. MUST be sent first by implementations compliant with this document.
0x02..0x08	RES1	Reserved for future versions
0x09	Suggester	Can act as shortcut suggester in a SHORTCUT exchange
0x0A	Shortcut Partner	Can act as shortcut partner in a SHORTCUT
0x0B	FQDN Resolver	Can resolve peer locators given as FQDN. Relevant only for the shortcut partner.
0x0C	Trusted Suggester	This peer can act as a trusted suggester as described in Section 4.
0x0D..0x7F	RES2	Reserved for future extensions
0x80..0xDF	RES3	Reserved
0xE0..0xFF	RES4	Reserved for private use

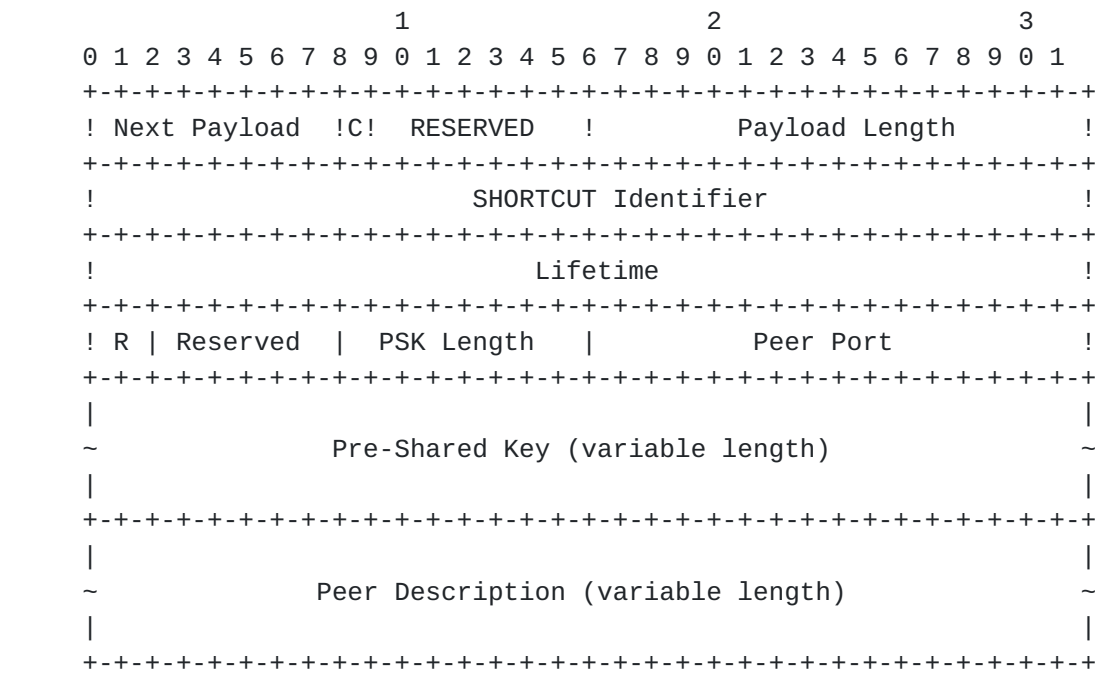
The IKE exchange Initiator MAY send multiple version capabilities. The Responder MUST send exactly one version capability, and that capability represents the version of the specification to be used.

A receiver MUST ignore any capabilities it does not recognize. Extension documents SHOULD consider the effects of the peer not recognizing such capabilities. If such extensions are critical for the operation of the protocol, a new version number may also be needed.

[RFC EDITOR PLEASE REMOVE THIS PARAGRAPH] For development and interoperability testing while this document is still a draft and IANA actions have not taken place, implementations can use the private-use value of 47831 as the ADVPN_SUPPORTED Notify type.

3.6. ADVPN_INFO Payload

The ADVPN_INFO payload is used in the SHORTCUT exchange to send information from the suggester to a shortcut partner about the shortcut.



The fields corresponding to the first 4 octets are defined as described in RFC 5996. The remaining fields are defined as follows:

- SHORTCUT Identifier (4 octets) - a 32-bit identifier for this shortcut. The same value MUST be used for both shortcut peers and it MUST be unique per suggester and partners pair. The value SHOULD be unique per suggester, i.e. there should not exist in the VPN two SHORTCUTs initiated by the same suggester with the same identifier.
- Lifetime (4 octets) - a 32-bit integer that indicates the maximum number of seconds that this shortcut recommendation should last. After this period of time lapses, the shortcut partners SHOULD tear down the shortcut SA. If the field is 0, the shortcut suggestion MAY last indefinitely. The shortcut partners MAY use a smaller timeout value than given here based on their policies.
- R - Role (2 bits) - this field indicates to the partner its designated role in the upcoming exchange (i.e. shortcut initiator or responder). Role assignment is decided by the shortcut suggester and, as mentioned earlier, it is outside the scope of this document. Value 00 is reserved and MUST NOT be used by implementations compliant to this specification. Value 01 indicates that the receiving peer MUST act as shortcut responder. Value 10 indicates that the peer receiving this payload MUST act as the shortcut initiator. Value 11 indicates that the receiving peer SHOULD NOT initiate the exchange immediately and MAY initiate the exchange at later stage. The waiting time before the peer initiates the exchange could be several minutes.

. PSK Length (1 octet) indicates the length in octets of the Pre-Shared Key. It MUST be set to zero if certificate authentication is to be used between the shortcut peers.
. Peer Port (16 bit) - is set to zero when none of the shortcut partners are behind a NAT. The suggester has IKEv2 and IPsec channels with both shortcut partners, so it is aware whether partners are behind a NAT or not. If one of the shortcut partners is behind a NAT, the Peer Port MUST be a non-zero value. This value is used for UDP encapsulation as defined in [RFC3948] between the partner that has received the shortcut payload and the peer shortcut partner.

  If the peer shortcut partner is behind a NAT, the Peer Port designates the port by which the NAT identifies the peer within its private network. The global IP address of the NAT is provided by the Peer Address Identification Payload (IDa). When the shortcut partner receiving the shortcut payload sends an IKEv2 or IPsec ESP packet to the peer shortcut partner, it MUST UDP encapsulate the packet with IP source set to its local IP address, source port set to 4500, IP destination set to the IP provided by the Peer Address Identification Payload, and destination port set to Peer Port. When the packet reaches the NAT gateway, the IP destination and destination port are translated by the NAT to private IP address and 4500. Similarly, IKEv2 and IPsec ESP packets sent by the peer shortcut partner are UDP encapsulated with IP source set to the private IP address of the peer shortcut partner, source port set to 4500. IDa determines the IP destination payload sent to the peer shortcut partner and the counterpart shortcut payload Peer Port field sent to the peer shortcut partner defines the destination port. This field MAY be 4500 or another value depending on whether the shortcut partner of the shortcut payload (described in this section) is also behind a NAT.

  If the peer shortcut partner is not behind a NAT, but the partner receiving this shortcut payload is behind a NAT, then the Peer Port value is set to 4500. When the shortcut partner sends an IKEv2 or an IPsec ESP packet, it MUST UDP encapsulate the packet with IP source set to its private IP, the port source set to 4500, IP destination set to the IP provided by the Peer Address Identification Payload (IDa) and the destination port 4500. When the packet reaches the NAT gateway, IP source and port are translated by the NAT with the global IP address and the port used to identify the shortcut partner. These two values are provided to the peer shortcut partner, by the Peer Address Identification Payload (IDa) and shortcut payload sent by the suggester to the peer shortcut partner.

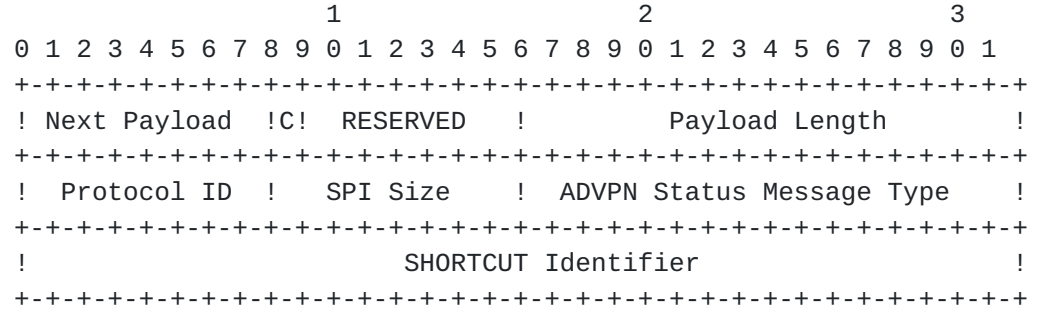
If the Peer Port is set to zero and the partner is behind a NAT, this is obviously a misconfiguration. The partner MAY return a RCODE set to SHORTCUT_PARTNER_UNREACHABLE. If the partner initiates the IKEv2 negotiation, it MUST ignore the Peer Port value and proceed to UDP encapsulation. The negotiation MAY be successful only if the peer shortcut partner is not behind a NAT. Similarly if the partner is not initiating the IKEv2 negotiation, the negotiation MAY be successful if the shortcut partner has received a properly configured shortcut payload.

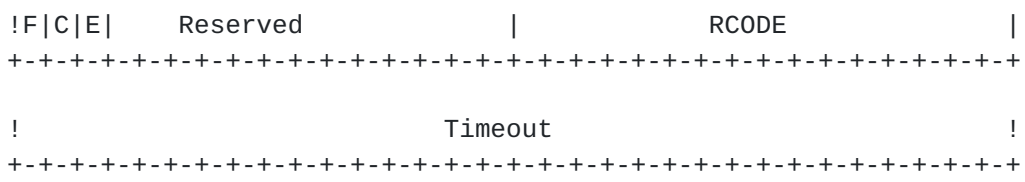
. Pre-Shared Key - An octet string used as the PSK in the IKE_AUTH exchange between the shortcut partners. This field MUST be the same in the ADVPN INFO payloads sent to the two shortcut partners. It MUST be randomly generated using a good random source, and it SHOULD be long enough to meet the security requirements of the deployment. In practice, this means that the length of the randomly generated data should be at least 16 octets long.
. Peer Description (variable length) - The length of this field is calculated by subtracting the combined lengths of the other fields from the value of the Payload Length field. It contains a description of the other peer, in a format that is simply a name, suitable for logging, and encoded in UTF-8.

[RFC EDITOR PLEASE REMOVE THIS PARAGRAPH] For development and interoperability testing while this document is still a draft and IANA actions have not taken place, implementations can use the private-use value of 248 as the ADVPN_INFO payload type.

3.7. ADVPN_STATUS Notification

The ADVPN_STATUS payload is used by a shortcut partner for conveying to the suggester and to the other SHORTCUT peer the status of the shortcut.





The fields corresponding to the first 4 octets are defined as described in RFC 5996. The remaining fields are defined as follows:

- Protocol ID (1 octet) MUST be zero, as specified in Section 3.10 of RFC 5996.
- SPI Size (1 octet) MUST be zero, in conformance with Section 3.10 of RFC 5996.
- ADVPN Status Message Type (2 octets) - MUST be yyyyyy. [RFC EDITOR NOTE: value assigned by IANA for ADVPN_STATUS]
- SHORTCUT Identifier - the 32-bit field from the SHORTCUT_INFO notification.
- F (1 bit) - Fatal. Set if this notification means that the SHORTCUT no longer exists.
- C (1 bit) - Critical. Set if the peer must understand this RCODE, or else delete the shortcut if the F bit is not set.
- E (1 bit) - Error. Indicates an error condition (rather than a policy change).
- RCODE (2 octets) - a 16-bit field as described in Section 3.8.1.
- Timeout - this 32-bit field indicates the maximum number of seconds that the shortcut service is not available by the peer.

The SHORTCUT_STATUS notification may be sent to the suggester in the SHORTCUT exchange response message. If either the suggester or a partner needs to communicate a change in status after the original SHORTCUT exchange is over, the same can be communicated in an INFORMATIONAL request. Additionally, the same notification is used in the IKE exchange between the shortcut partners, so that they can match that exchange and the resulting Security Associations to the shortcut. Since there may be more than one active shortcut between a pair of shortcut partners, the notification is inserted into the exchanges that create child SAs, either the IKE_AUTH exchange or the CREATE_CHILD_SA exchange.

[RFC EDITOR PLEASE REMOVE THIS PARAGRAPH] For development and interoperability testing while this document is still a draft and IANA actions have not taken place, implementations can use the private-use value of 47833 as the ADVPN_STATUS notify type.

3.8. The SHORTCUT Exchange

This new exchange type defines how the suggestions are conveyed. It is designed for sending the SHORTCUT data. The suggester initiates the SHORTCUT exchange and each of the shortcut partners responds to the notification. The shortcut partner acts as the [RFC5996] Responder. The exchange is constructed as follows:

```
   HDR, SK {IDa, ADVPN_INFO, IDi,
          IDr[, TSi][, TSr][, VID]} -->
                                  <== HDR, SK {N(ADVPN_STATUS)}
```

The IDa payload, defined in Section 3.4. provides the location of the other shortcut peer and it may not necessarily have the same data as the IDi or IDr payloads.

The ADVPN_INFO payload contains PSK and any other information needed for establishing the SHORTCUT IKE SA, as well as the optional time out information.

The IDi Identification Payload contains the identity of the shortcut initiator. The shortcut initiator MUST use this identifier when establishing the shortcut and the shortcut responder MUST verify that this identifier was used.

The IDr Identification Payload contains the identity of the shortcut responder. The shortcut initiator MUST use this payload in the subsequent IKE_AUTH exchange with the shortcut responder.

The TSi and TSr Traffic Selector Payloads (when present) contain, respectively, traffic selectors the intended Initiator and intended Responder in the IKE_AUTH exchange between the shortcut partners. The content is as specified in Section 3.13 of RFC 5996 [IKEV2].

The responder checks that everything in the request is acceptable according to local policy. If not, it MUST return an error RCODE. If the shortcut is acceptable, then it either tries to establish the shortcut IKE SA, and reports the results in the SHORTCUT response, or it immediately responds with a SHORTCUT_ACK RCODE, and follows up with more detailed status reports in future Informational exchanges.

In the subsequent IKE_AUTH exchange between the two partners, the initiator MUST use the IDi and IDr payloads as specified in the exchange described in this section, and MUST use a subset (not necessarily a proper

subset) of the traffic selector payload data, if such data has been included in the SHORTCUT exchange.

When shortcut are performed within a single administrative domain, IKE PAD are likely to be configured to trust all partners associated to a given domain. This will most probably be reflected by a common field in the certificate, and the ID is determined by the certificate. On the other end, PSK and a random ID MUST be added in the PAD of the partners. Creation of a new entry in the PAD is done since the suggester is trusted. If not the partner MUST return a SHORTCUT Response with UNMATCHED SHORTCUT PAD.

3.8.1. Content of the IDi and IDr payloads

The identification payloads in the SHORTCUT request message are later used in the IKE_AUTH exchange between the shortcut partners. They are required to follow the rules in RFC 5996 for authentication of the IKE SA. However, those rules are vague and left to specific profiles, specific implementations and specific administrative decisions.

Since AD-VPN is intended to work with multiple implementations and multiple administrative domains, we believe it is necessary to specify the content of these payloads more strictly.

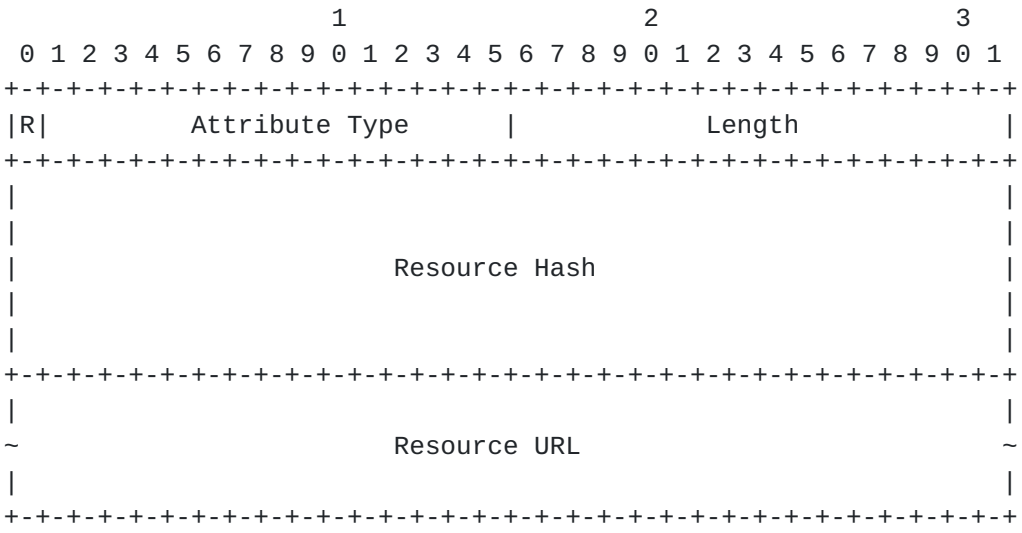
If a shortcut partner is supposed to authenticate using a certificate (and, therefore, the PSK Length field in the ADVPN_INFO payload is set to zero), then the ID payload matching this partner MUST match the certificate that it has. In this case, the ID payload MUST be from one of the following types:

- ID_IPV4_ADDR or ID_IPV6_ADDR. This ID type MAY be used only if the certificate contains an alternate name extension of type iPAddress, and MUST contain the same IP address as the extension.
- ID_FQDN. This ID type MAY be used only if the certificate contains an alternate name extension of type dNSName, and MUST contain the same FQDN as the extension.
- ID_RFC822_ADDR. This ID type MAY be used only if the certificate contains an alternate name extension of type rfc822Name, and MUST contain the same NAI as the extension.
- ID_DER_ASN1_DN. This ID type can always be used, and if used, MUST contain an exact copy of the subject field of the certificate.

If the partners are using a PSK to authenticate, the ID payloads need not reflect any real name of the partners, as that information is conveyed in the ADVPN_INFO payload. The ID payloads do, however, need to be unique so as to allow a quick lookup of the ID payload. To ensure this, the ID payload MUST be of type ID_KEY_ID, and the content MUST be unique. To ensure uniqueness across administrative domains, the content of the ID payload in such cases MUST be randomly or pseudo-randomly generated and MUST have 128 bits.

3.9. PROTECTED_DOMAIN Attribute Type

This is a new attribute for the CFG payload. In a request, its size MUST be a constant zero. In a response, it MUST be at least 20 octets long, having the following format:



PROTECTED_DOMAIN Configuration Attribute Format

In the above diagram, Attribute Type is ZZZ, [RFC EDITOR NOTE: value assigned by IANA for PROTECTED_DOMAIN], Resource Hash is the SHA-1 hash of the resource, and Resource URL is an HTTP URL pointing at the resource.

The corresponding entry for this attribute as in the table in section 3.15.1 of RFC 5996 is as follows:

Attribute Type	Value	Multi-Valued	Length

PROTECTED_DOMAIN	1	NO	0, or 20+ octets

The format of the resource itself is described in Section 4.1.

3.10. SHORTCUT Response Codes (RCODE)

This section provides more information on the use of the response codes (RCODE). RCODE is a 16-bit field, used by the shortcut partners to indicate the status on the SHORTCUT notification received.

The RCODEs we consider in this document are the following:

Value	Description
0	SHORTCUT_ACK
1	SHORTCUT_OK
2	SHORTCUT_PARTNER_UNREACHABLE
3	TEMPORARILY_DISABLING_SHORTCUT
4	SHORTCUT_PARTNER_UNREACHABLE
5	IKEv2_NEGOTIATION_FAILED
6	UNMATCHED_SHORTCUT_SPD
7	UNMATCHED_SHORTCUT_PAD

3.10.1. SHORTCUT_ACK

The RCODE value for SHORTCUT_ACK is: 0 (zero)

This RCODE indicates that the receiving partner has accepted the shortcut, but has yet to establish the shortcut IKE SA. This RCODE MUST be used only in the response for a SHORTCUT exchange.

3.10.2. SHORTCUT_OK

The RCODE value for SHORTCUT_OK is: 1

This RCODE indicates that the shortcut has been successfully established between the shortcut partners.

3.10.3. SHORTCUT_PARTNER_UNREACHABLE

The RCODE value for SHORTCUT_PARTNER_UNREACHABLE is: 2

This RCODE indicates that the attempt to establish the recommended shortcut has failed because the partner peer was unreachable. This may happen, for example, if the partner peers are behind separate NATs, or a firewall drops packets between the shortcut partners. It may also be that the partner peer is only available through a specific interface. In addition, the partner peer may have been temporarily disconnected or its shortcut service has been temporarily disabled as explained in subsection 3.10.4.

It is the responsibility of the shortcut suggester to determine the reason of the observed unreachability as well as what policy to apply. However, the shortcut suggester SHOULD NOT initiate another SHORTCUT exchange to the shortcut partners before the Timeout indicated in the shortcut Data. If the Timeout value is not present, then it is up to the shortcut suggester to decide when a new SHORTCUT exchange should be initiated.

3.10.4. TEMPORARILY_DISABLING_SHORTCUT

The RCODE for TEMPORARILY_DISABLING_SHORTCUT is: 3

This RCODE indicates that the shortcut recommendation is refused by the shortcut peer because it has deactivated the shortcut service. In other words, this RCODE indicates that any attempt to establish shortcuts will be refused independently of the SHORTCUT exchange sent. For example, the shortcut service could be disabled when the shortcut peer is overloaded.

If the shortcut initiator generates this response code, then it SHOULD NOT initiate the shortcut negotiation.

When receiving an ADVPN_STATUS notification with this response code, the shortcut suggester SHOULD NOT initiate any other SHORTCUT exchange before the Timeout indicated in the shortcut Data. If the Timeout value is not present, then it is up to the shortcut suggester to decide when a new SHORTCUT exchange should be initiated.

3.10.5. IKEV2_NEGOTIATION_FAILED

The Shortcut Type for IKEV2_NEGOTIATION_FAILED is: 4

This RCODE indicates that the IKEv2 negotiation between the two partner peers did not complete successfully. That is, the shortcut recommendation was accepted and acted upon, but the IKEv2 negotiation failed. This RCODE does not provide information on the reasons the shortcut establishment failed, and thus other more specific RCODEs (see below) SHOULD be preferred by implementations when this is possible.

3.10.6. UNMATCHED_SHORTCUT_SPD

The RCODE for UNMATCHED_SHORTCUT_SPD is: 5

This RCODE indicates an error resulting from the analysis of the SHORTCUT exchange. Before establishing a shortcut, the shortcut initiator MUST check that the shortcut partner's IP address matches its Security Policy Database (SPD). If a mismatch occurs with the shortcut initiator's SPD, the shortcut initiator MUST NOT initiate the shortcut. In this case, the initiator MUST use the UNMATCHED_SHORTCUT_SPD RCODE in its ADVPN_STATUS notification.

If the mismatch occurs with the shortcut responder, it MUST send to the shortcut suggester the UNMATCHED_SHORTCUT_SPD RCODE in its ADVPN_STATUS notification. Eventually the shortcut initiator will start an IKEv2 negotiation. The shortcut responder SHOULD terminate the IKEV2 negotiation with a TS_UNACCEPTABLE. At this stage the shortcut cannot be established and the shortcut initiator MUST respond to the shortcut suggester with the IKEv2_NEGOTIATION_FAILED Shortcut Type in its ADVPN_STATUS Notify Payload.

When receiving ADVPN_STATUS with this RCODE, the shortcut suggester SHOULD NOT reinitiate a SHORTCUT exchange with the shortcut partners with the same Traffic Selectors in short order.

3.10.7. UNMATCHED_SHORTCUT_PAD

The Shortcut Type for UNMATCHED_SHORTCUT_PAD is: 6

This RCODE indicates an error resulting from the analysis of the SHORTCUT exchange. Before establishing a shortcut, the shortcut initiator MUST

check that the shortcut partner's IP address and Identities IDi/IDr match its Peer Authentication Database (PAD). If a mismatch occurs with the shortcut initiator's PAD, the shortcut initiator MUST NOT initiate the establishment of the recommended shortcut. The initiator then sends the UNMATCHED_SHORTCUT_PAD RCODE in its ADVPN_STATUS notification

If the mismatch occurs with the shortcut responder, it MUST send to the shortcut suggester the UNMATCHED_SHORTCUT_PAD RCODE in its ADVPN_STATUS notification. Eventually the shortcut initiator will start an IKEv2 negotiation. The shortcut responder SHOULD terminate the IKEV2 negotiation with a TS_UNACCEPTABLE. Thus, the shortcut cannot be established and the shortcut initiator MUST return the shortcut suggester the IKEv2_NEGOTIATION_FAILED Shortcut Type in its ADVPN_STATUS.

When receiving ADVPN_STATUS with this RCODE, the shortcut suggester SHOULD NOT reinitiate a SHORTCUT exchange with the shortcut partners with the same Traffic Selectors in short order.

4. Trusted Suggester

Advertising this capability means that the sender supports sending its entire protected domain through the PROTECTED_DOMAIN attribute in the CFG payload.

The intended way to use this is that a spoke node, whether a remote access client or a gateway, is configured only with credentials for a single hub gateway. On the first Initial exchange, the hub gateway advertises the TRUSTED_SUGGESTER capability. In the IKE_AUTH exchange or in a later Informational exchange, the spoke sends a CFG_REQUEST, asking for the PROTECTED_DOMAIN. The reply (see Section 3.8.1. gives the spoke a URL for a resource that contains the entire protected domain of the hub, which is the entire protected domain of the VPN. The CFG_REPLY also contains a hash of the resource, which has an important security benefit. Since there are attacks against HTTP, and no obvious way to secure HTTPS in this context, using a protected exchange to send a hash of the resource makes sure that the retrieved resource is in fact the intended one.

The list of IP addresses and ranges returned in the resource is used to populate the SPD, and provides an initial configuration with all VPN traffic going through the hub gateway. Normal ADVPN protocol operation can later optimize the traffic, but this mechanism ensures that bootstrapping does not require extensive manual configuration.

The procedure described above begins with the bare minimum of configuration. Some nodes will begin with some initial configuration, in

which case it is up to them to harmonize their static configuration with the data from the trusted suggester. It is perfectly valid to use only a subset of the protected domain in populating the SPD. Appendix C provides an illustrative example of the use of the PROTECTED_DOMAIN attribute.

4.1. Format of Protected Domain Resource

The protected domain resource is formatted much like a CFG reply. It is a series of Configuration Attributes (see section 3.15.1 in RFC 5996), but the only attribute types allowed are INTERNAL_IP4_SUBNET (13) and INTERNAL_IP6_SUBNET (15). The last Configuration Attribute has the R bit set to mark it as the last one.

For an example, the following resource indicates a protected domain comprised of all three test networks from [RFC5737] and [RFC3849]:

```
0x0D 0x08 0xC0 0x00 0x02 0x00 0xff 0xff 0xff 0x00
   - 8-octet IP4_SUBNET: 192.0.2.0 (255.255.255.0)
0x0D 0x08 0xC6 0x33 0x64 0x00 0xff 0xff 0xff 0x00
   - 8-octet IP4_SUBNET: 198.51.100.0 (255.255.255.0)
0x0D 0x08 0xCB 0x00 0xD1 0x00 0xff 0xff 0xff 0x00
   - 8-octet IP4_SUBNET: 203.0.113.0 (255.255.255.0)
0x8F 0x11 0x20 0x01 0x0D 0xB8 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x20
   - Last 17-octet IP6_SUBNET: 2001:DB8::/32
```

4.2. Lifetime of The Data In Protected Domain Resource

As the Protected Domain Resource is delivered over HTTP/1.1 or above, the Cache-Control ([HTTPCache]) directives can be used to assign a lifetime. The IPsec node MUST expire the entries from the SPD as soon as the data would expire from the cache, or MUST fetch fresh data beforehand.

5. IPsec Policy

This section discusses the implications of the use of the ADVPN Protocol on IPsec policy.

5.1. Security Policy Database (SPD)

The SHORTCUT exchange described in Section 3.2. conveys policy in the sense of Section 4.4.1 of RFC 4301 ([IPSECARCH]). In the terms of that document, these are SPD elements. Assuming these elements are accepted, they update the existing security policy of the receiver.

The entries specified in a SHORTCUT exchange are inserted into the SPD immediately before the entry that they are updating, so that these new entries take precedence over existing ones.

RFC 4301 does not specify time limits for SPD entries. In that sense, this document updates RFC 4301. SPD entries now come in two flavors: static entries, which have no expiration time and are defined by an administrator, and dynamic entries which have an expiration time as specified in the SHORTCUT exchange.

For example, suppose a static entry exists for the remote subnet 192.0.2.0/23, and the local subnet is 192.0.1.0/24. Initially, the entry looks like this:

Local=192.0.1.0/24,Remote=192.0.2.0/23,PROT,Peer=vpngw.example.com

Assume now that a SHORTCUT exchange is received which describes gateway foo.example.com, and remote subnet 192.0.2.0/24. The database will look as follows:

Local=192.0.1.0/24,Remote=192.0.2.0/24,PROT,Peer=foo.example.com

Local=192.0.1.0/24,Remote=192.0.2.0/23,PROT,Peer=vpngw.example.com

Because of the rules of processing as specified in Section 5.1 of RFC 4301, the earlier entry takes precedence, and overrides the second entry for subnet 192.0.2.0/24. The second entry still applies to 192.0.3.0/24.

5.1.1. Security Policy Database Cache (SPD Cache)

The SPD Cache also needs to be updated. With the above entry, a cache entry was created reflecting the matching SA. If no change to the cache is

made, the IPsec stack will continue to use the existing SA despite the change in policy. Since implementations of the SPD cache vary widely, we do not specify the exact way to handle this change, but discuss below some implementation suggestions.

One way to handle this would be to narrow the existing SPD Cache entry so as to cover only the selectors which are not affected by the SHORTCUT. This has the advantage of forcing the SPD cache entry to a failure match with the negotiated SA. Whether this is a problem depends on the implementation of these databases. It is likely not a good idea to also narrow the existing SAs. While it should be fine for outbound SAs, it will cause the IPsec stack to drop validly encrypted packets on inbound processing.

Another way to handle this would be to simply delete the SPD cache entry, forcing a re-evaluation of the SPD for the next packets. This causes an even more serious discrepancy between the Security Association Database (SAD) and SPD Cache. This should only be done if it is possible to match existing SAs to new SPD cache entries, which, again, depends on the implementation details.

The one foolproof way is to erase both SPD cache entries and SAs, sending the appropriate DELETE payloads to the peer. This is perfectly compliant and perfectly functional, but will create more work for the IKE daemon.

5.2. Peer Authentication Database (PAD)

This database will also be updated with a temporary entry when a SHORTCUT exchange is received. The entry includes the name, IP address and a specification of either PSK or certificate authentication. This entry MUST also expire when the SHORTCUT expires.

It is conceivable that peers will appear in both static and dynamic entries. It is also possible that the same peer will be mentioned in multiple SHORTCUT exchanges, each with a different expiration time. An implementation of this specification MUST track all such entries. Two entries will be considered to represent the same entity if either they share both ID and certificate, or if they share ID and IP address.

If all entries matching a particular entity expire, then the implementation MUST delete all IKE and child SAs associated with that entity.

6. Security Considerations

No lifetime is specified for the Pre-Shared Key (PSK) so the shortcut suggester SHOULD generate the PSK value with plenty of entropy. See [RANDOMNESS] for advice on generating random numbers for cryptographic purposes. The shortcut partners may rekey as needed and may even use the PSK value for reauthentication, although it is not clear that there is much value in doing so. If one of the shortcut partners decides that the PSK is too old (recognizing that it is only used for authentication), it may simply tear down the shortcut SA. Eventually, the shortcut suggester will set up the shortcut again, if it is needed.

To head off this situation, the shortcut suggester may periodically initiate a new SHORTCUT exchange to each of the two shortcut partners. If a shortcut partner receives a SHORTCUT exchange suggesting a shortcut that already exists with new parameters, the shortcut partner SHOULD establish a new shortcut SA with the peer partner using the new parameters and then tear down the old shortcut SA.

7. IANA Considerations

IANA is requested to allocate a new payload type from the "IKEv2 Payload Types" registry with the name "Identification - Peer Address", i.e. "IDa", and this document as the corresponding reference document.

IANA is requested to allocate a new Configuration Payload Attribute Type with name "PROTECTED_DOMAIN", not multivalued, with a length of "0 or more octets".

IANA is requested to allocate a new payload type from the "IKEv2 Payload Types" registry with the name "ADVPN Information", i.e. "ADVPN_INFO", and this document as the corresponding reference document.

IANA is requested to allocate a new exchange type from the "IKEv2 Exchange Types" registry with name "SHORTCUT" and this document as reference.

IANA is requested to assign two code points from the "IKEv2 Notify Message Types - Status Types" registry, as follows. The reference document for all three shall be this document:

. SHORTCUT_SUPPORTED

. SHORTCUT_STATUS

IANA is requested to set up a new registry called "IKEv2 ADVPN Capabilities". The value is 8-bit, and the range is partitioned as follows:

. 0x00 reserved for padding
. 0x01-0x08 used for version indication and negotiation
. 0x09-0x07F used for supported features
. 0x80-0xDF reserved
. 0xE0-0xFF reserved for private use

The policy for allocations from the version range shall be "Standards Action". The policy for allocation from the features range shall be "Expert Review". The initial allocation for this registry is as defined in the table in Section 3.5. with the additional column of reference specification, which shall be set to this document.

IANA is requested to set up a new registry called "IKEv2 AD-VPN Response Codes". The value is 16-bit, and the range is partitioned as follows:

. 0x0000-0xBFFF Used for RCODEs
. 0xC000-0xFFFF Reserved for private use

The policy for allocations from this registry shall be FCFS. The initial allocation is given in the table in Section 3.8.1.

8. References

8.1. Normative References

[IKEV2] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

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

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

[RFC1034] Mockapetris, P., "Domain Names - Concepts and Facilities", RFC 1034, November 1987.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[HTTPCache] Fielding, R., Nottingham, M., and Reschke, J., "Hypertext Transfer Protocol (HTTP/1.1): Caching", draft-ietf-httpbis-p6-cache-23 (work in progress), July 2013.

8.2. Informative References

[ADVPNreq] Manral, V., "Auto Discovery VPN Problem Statement and Requirements",RFC 7018.

[RANDOMNESS] Eastlake, 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[RFC3492] Costello, A., "Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)", RFC 3492, March 2003.

[MEDIATION] Brunner, T., "IKEv2 Mediation Extension draft-burner-ikev2-mediation-00"

[RFC5737] Arkko, J., Cotton, M., and Vegoda, L., "IPv4 Address Blocks Reserved for Documentation", RFC 5737, January 2010.

[RFC3849] Huston, G., Lord, A., and Smith, P., "IPv6 Address Prefix Reserved for Documentation", RFC 3849, July, 2004.

9. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

The authors of this draft would like to acknowledge the following people who have contributed to or provided substantial input on the preparation of this document or predecessors to it: Scott McKinnon, Vishwas Manral, Valery Smyslov, Michael Richardson and Yaron Sheffer.

Appendix A. ADVPN Example Use Cases

This appendix presents a few example situations where the ADVPN protocol may be useful and illustrates how it works.

A.1. Branch Office Videoconference

In this example, users have initiated a videoconference between two branch offices of SmithCo located in Ashby and Bedford. Each branch office has an IPsec gateway that is configured to send all traffic to the IPsec gateway at the main SmithCo office in Paris. Figure 6 illustrates this initial situation, showing these three IPsec gateways and the IPsec SAs in place when the videoconference starts.

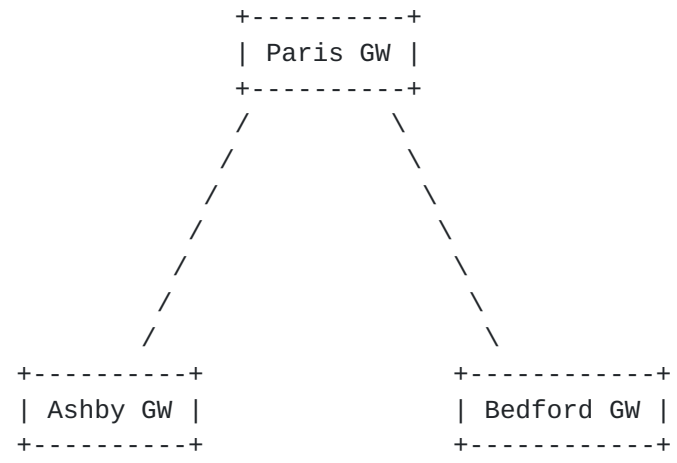


Figure 6: Initial SmithCo IPsec SAs

All of these gateways support SHORTCUT exchanges and have been configured to use them within SmithCo. Therefore, they all sent the ADVPN_SUPPORTED notification payload described in Section 3.4 to each other in their initial IKE exchanges. This means that they are all aware that SHORTCUT exchange may be used on the IPsec SAs illustrated in Figure 6.

Once the videoconference begins, the Paris GW notices a large amount of videoconference traffic between the Ashby GW and the Bedford GW. The Paris GW has been configured to permit videoconference traffic to trigger a shortcut between two branch gateways so it initiates a SHORTCUT exchange to the Ashby and Bedford GWs, suggesting that they establish a shortcut. In this instance, it identifies the Ashby GW as the shortcut initiator by

setting the I bit in the exchange sent to that gateway and leaving that bit cleared in the exchange sent to Bedford GW.

Because all gateways in SmithCo have certificates from the same CA and have been configured to trust that CA to issue certificates, there is no need to use a PSK. The Paris GW simply sets the IDi Identification Payload field of the SHORTCUT exchange to the subject DN of the Ashby GW and the IDr Identification Payload field of the SHORTCUT exchange to the subject DN of the Bedford GW.

The Paris GW sets the TSi Traffic Selector Payload and TSr Traffic Selector Payload fields in the SHORTCUT exchange to indicate that the Ashby GW should only use this shortcut for videoconferencing traffic destined for the network behind the Bedford GW and vice versa.

After receiving the SHORTCUT exchange, the Ashby GW establishes an IKEv2 exchange with the Bedford GW and then establishes an IPsec security association between the two. Figure 7 shows the SAs in use after the shortcut has been established.

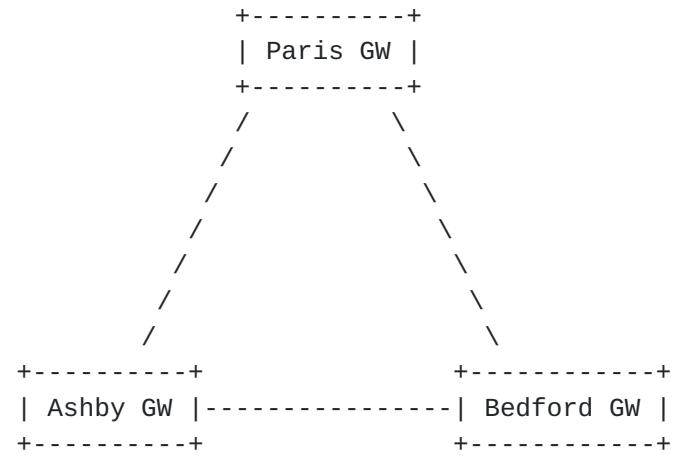


Figure 7: SmithCo IPsec SAs with the shortcut established

After the timeout period specified by the Paris GW, the shortcut between Ashby GW and Bedford GW will be terminated. If the videoconference is finished before that time, the shortcut may also be terminated due to inadequate traffic, at the discretion of the Ashby GW and Bedford GW.

A.2. Optimization for Videoconference with Partner

In this example, SmithCo has added a partner JonesCo and established an IPsec SA between Paris GW (the main SmithCo office) and Tokyo GW (the main JonesCo office).

Users have initiated a videoconference between the Ashby branch office of SmithCo and the Concord branch office of JonesCo. Each branch office has an IPsec gateway that is configured to send all traffic to the IPsec gateway at the main office for that company. Figure 8 illustrates this initial situation, showing these four IPsec gateways and the IPsec SAs in place when the videoconference starts.

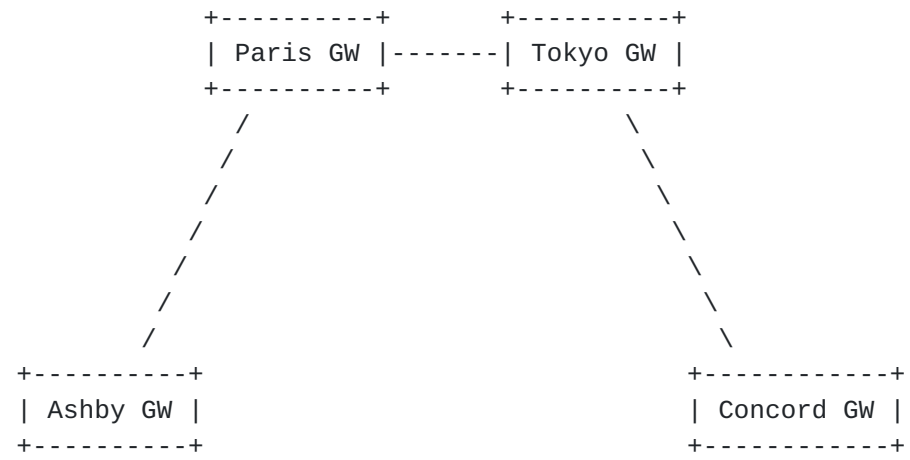


Figure 8: Initial IPsec SAs within SmithCo and JonesCo

All gateways in this example support SHORTCUT exchange. Therefore, they all sent the ADVPN_SUPPORTED notification payload described in Section 3.4 to each other in their initial IKE exchanges. This means that they are all aware that SHORTCUT exchange may be used on the IPsec SAs illustrated in Figure 8. Further, these gateways have been configured to use SHORTCUT exchange to optimize routing for video traffic within their organizations and among SmithCo and JonesCo gateways.

Once the videoconference begins, the Paris GW notices a large amount of videoconference traffic transiting the Paris GW between the Ashby GW and the Tokyo GW. Therefore, the Paris GW initiates a SHORTCUT exchange to the Ashby GW and the Tokyo GW, suggesting that they establish a shortcut. We will not cover all the details of this process because most are similar to the previous example. However, assume that the Ashby GW and the Tokyo GW have certificates from different CAs and may not be configured to trust each other's CA. Therefore, the Paris GW generates a PSK and sends it to

both the Ashby GW and the Tokyo GW. The Ashby GW and the Tokyo GW use this PSK to establish a shortcut SA, as shown in Figure 9. Because of the Traffic Selectors sent by the Paris GW in the SHORTCUT exchange, this shortcut SA may only be used for video traffic between the Ashby GW and JonesCo.

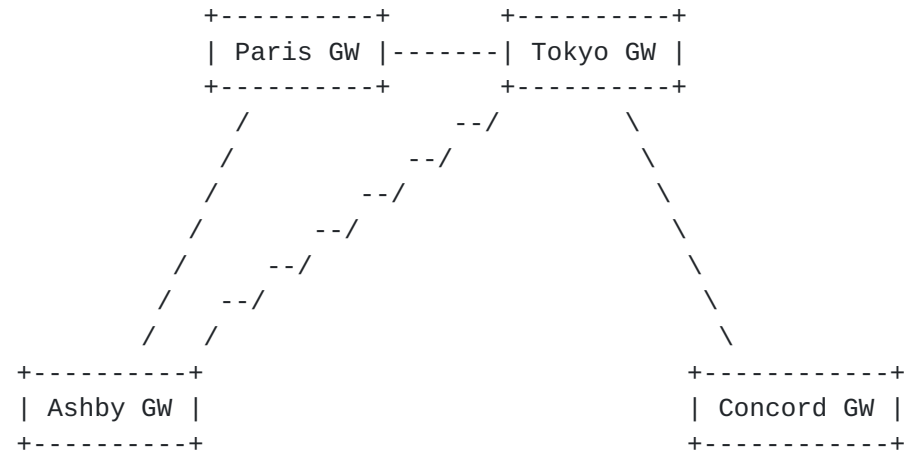


Figure 9: SmithCo and JonesCo with First Shortcut

After this first shortcut SA has been established, Tokyo GW notices large volumes of video traffic between Ashby GW and Concord GW. Therefore, Tokyo GW initiates a SHORTCUT exchange to the Ashby GW and the Concord GW, suggesting that they establish a shortcut. We do not cover all details of this process because they are mostly similar to the previous example. Again, the Ashby GW and the Concord GW probably have certificates from different CAs so the Tokyo GW generates a PSK and sends it to both the Ashby GW and the Concord GW. The Ashby GW and the Concord GW use this PSK to establish a shortcut SA, as shown in Figure 10. Because of the Traffic Selectors sent by the Tokyo GW in the SHORTCUT exchange, this shortcut SA may only be used for video traffic between the Ashby GW and the Concord GW.

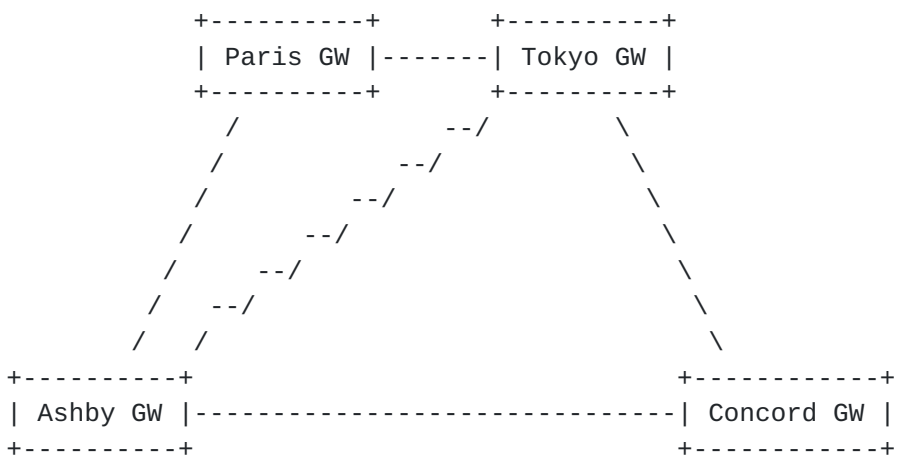


Figure 10: SmithCo and JonesCo with Second Shortcut

After some period, the Ashby GW or the Tokyo GW may realize that no traffic is flowing over the SA between them and therefore decide to terminate this SA. This will result in the SA configuration shown in Figure 11.

Note that this optimal SA configuration has been reached without needing to have any special configuration or global knowledge and it involves multiple domains. The only requirement is a policy on the Paris GW and the Tokyo GW indicating that video traffic between SmithCo and JonesCo should be optimized by creating shortcut SAs.

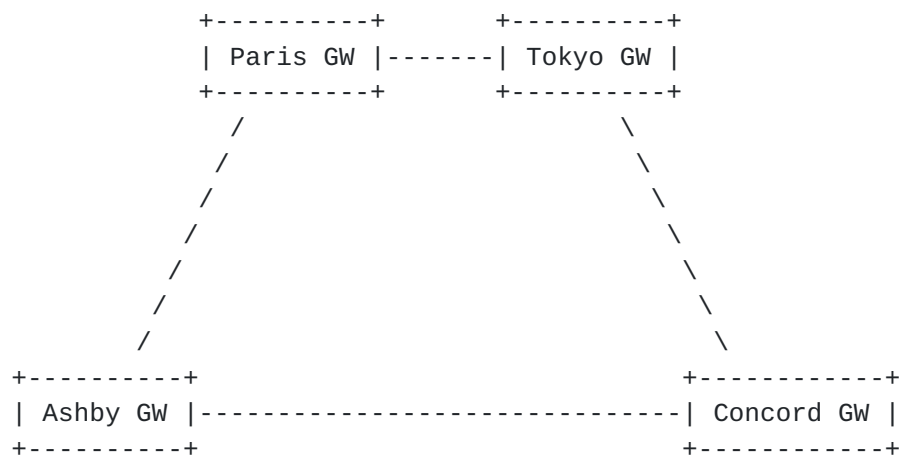


Figure 11: SmithCo and JonesCo in Final Configuration

The shortcut between the Ashby GW and the Concord GW will remain up until its timeout is reached or traffic levels on this SA drop off because the videoconference has finished.

A.3. Heterogeneous Wireless Networks Traffic

As wireless networks increase their access capacities, denser deployments will become the norm. In addition, we observe an increasing number of cases where operators, for various reasons that are outside of the scope of this document, opt for network deployments that use a variety of coverage sites. In practice, this means that, for instance, macro cells are complemented by smaller cells (pico cells, femto cells, etc.) that boost capacity and improve end-user experience. Today's cellular networks can provide access rates in the order of tens of Mb/s with high quality of service guarantees, and can thus be used as connections where small and medium enterprises can base their VPNs. Within this context, the operator may use different gateways for securing subscriber VPN traffic.

Consider, for example, the case illustrated in Figure 12 where two colleagues from different departments of the same company use multimedia conferencing to collaborate with some customers. Dotted lines in the Figure indicate IP connectivity, while dashed lines indicate an established SA. All gateways and endpoints in the Figure support the protocol described in this document, i.e., they have indicated so to each other as described in Section 3.4. One of them, Peer 1 has joined the teleconference while on the go, but will be arriving at the company office prior to the conclusion of the teleconference. As Peer 1 roams in the

mobile network, changing cell sites as it travels towards the office, the multimedia traffic flows through the Macro GW.

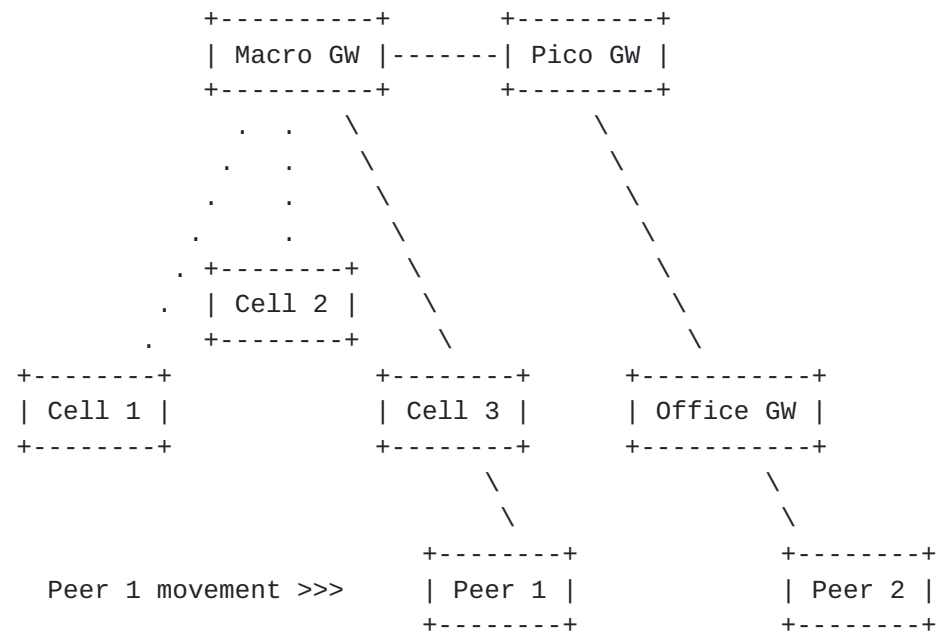


Figure 12: Initial IPsec SAs within the HetNet

Note that both the Macro GW and the Pico GW are in the realm of the mobile operator, while the Office GW is in the realm of the company.

The company and the mobile operator have an already established trust relationship. Moreover, for end-user experience reasons as well as traffic flow optimization both the company network administrators and the mobile operator have policies that favor traffic routes that are contained in the local company network.

Once Peer 1 enters the area of the company campus the wireless network small-cell deployment covering the company buildings is the preferred means of connecting to the network, both from the perspective of the company and the mobile operator. At this stage in our scenario, the fact that Peer 1 is in the coverage area of the Pico GW is recognized by the Macro GW, which initiates (as a shortcut suggester) the procedure described in Section 3. As a result, the first step in the route optimization is performed and Peer 1 sets up the shortcut with the Pico GW, which becomes its shortcut partner.

Figure 13 illustrates the newly established shortcut as well as the fact that Peer 1 continues to use the same radio interface as before, i.e. this scenario does not involve vertical handovers.

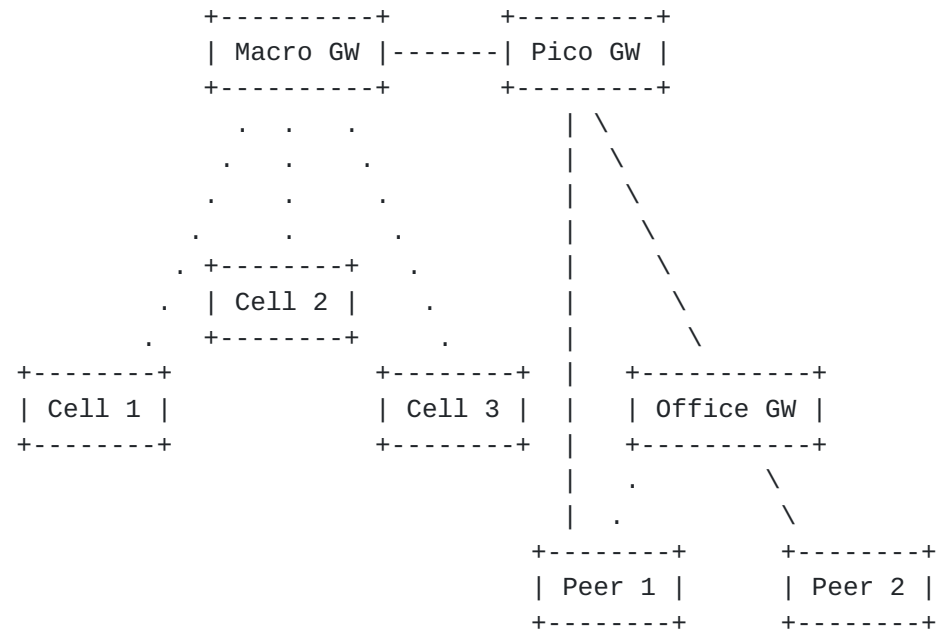


Figure 13: First route optimization within the HetNet

Once Peer 1 moves within the company premises and establishes the shortcut with the operator Pico GW more route optimization opportunities arise, and the ADVPN protocol can implement them without requiring any additional manual configuration neither by the operator nor by the company administrator.

At this stage, we assume that the Pico GW can determine the fact that Peer 1 could become a shortcut partner of the Office GW. Similarly to what was mentioned above, the Pico GW initiates the shortcut (i.e. acts as a shortcut suggester) indicating to Peer 1 and the Office GW that they should establish an SA with each other. The partners agree to these recommendations, as per their respective local policies, and proceed with the establishment. At the end of this process, the configuration is as illustrated in Figure 14.

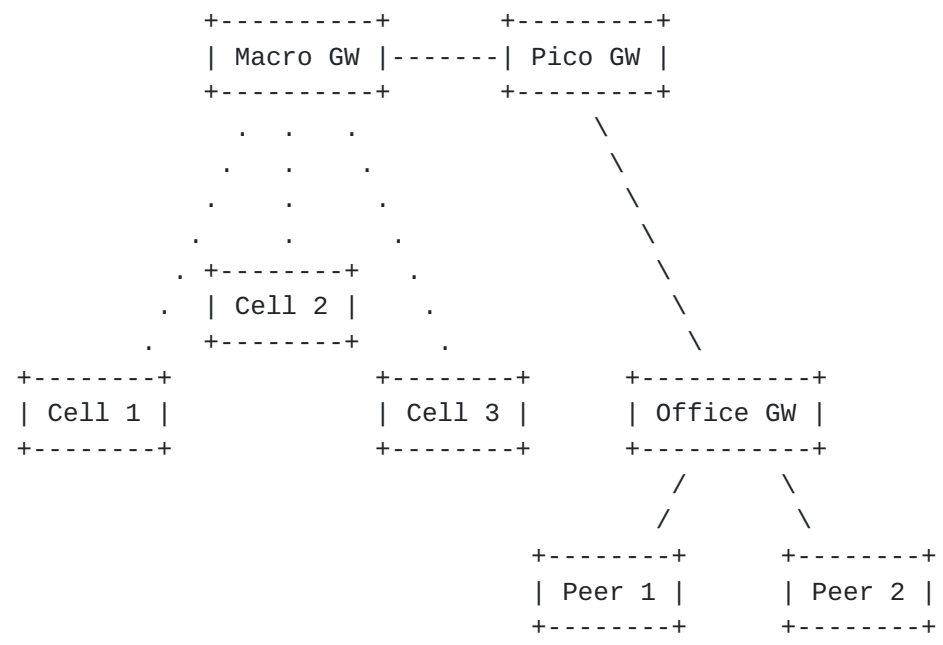


Figure 14: Second route optimization within the HetNet

After this optimization all IPsec traffic is contained within the local small-cell wireless network. Note that the company network may include several pico cells, all of which can establish SAs with the Office GW.

In principle, the protocol can be used to proceed with a further traffic optimization. Namely, Peer 1 and Peer 2 can establish a direct shortcut between each other, i.e. become shortcut partners and thus avoid routing through the Office GW. This is a decision that the Office GW may take based on local connectivity information. In this case, after following the same procedure described earlier, the two Peers will establish an SA, as illustrated in Figure 15.

As Figure 15 shows, traffic may still flow through the Office routers but Peer 1 and Peer 2 do not need to maintain an SA with the Office GW (if there is no other traffic).

Finally, note that, in principle, the Office GW could determine that since no traffic is flowing through its SA with the Pico GW, the respective SA could be temporarily terminated and initiated later on when the need arises. This final configuration is illustrated in Figure 16.

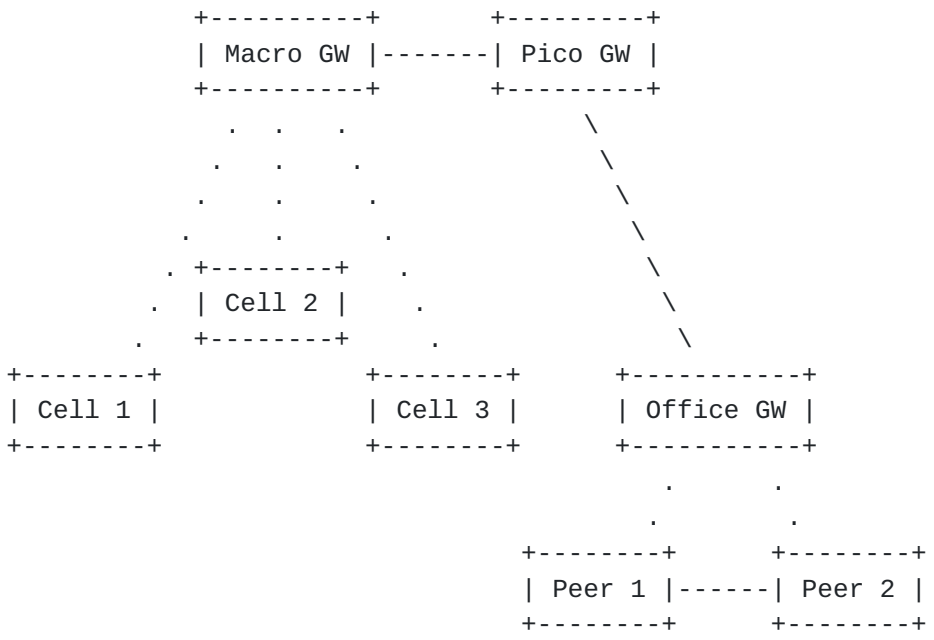


Figure 15: Third route optimization within the HetNet

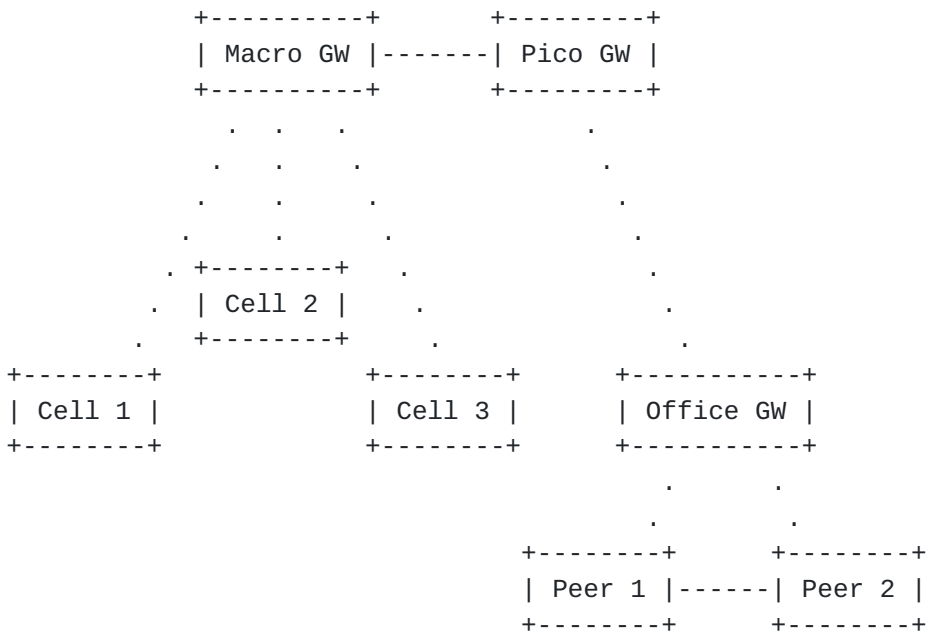


Figure 16: Final configuration

Appendix B. Comparison Against ADVPN Requirements

This section compares the ADVPN protocol specified in this document against requirements set by [ADVPNreq] (Section 4).

Requirement #1 :

This section details modifications when an endpoint, a gateway, a spoke and a hub is added or removed or changed.

End points establish a tunnel with a gateway to communicate with another endpoint. The gateway may use the ADVPN protocol to optimize communication and either set up endpoint-to-endpoint communication if both endpoints are attached to the "initial gateway", or to point to a "closer alternative gateway". The ADVPN protocol described in this document, impacts either the two endpoints or the endpoint and the "closer alternative gateway". Hubs or gateways other than the "initial gateway" or the "closer alternative gateway" IPsec configuration are not impacted.

An ADVPN is changed means that its IP address is modified. Updating the outer IP address is the purpose of MOBIKE and involves the two peers connected with their outer IP addresses.

Similarly, removing an endpoint only impacts the IPsec configuration of the gateways or the other endpoint it is communicating with. It is up to local policy that the "initial gateway" decides to keep the IPsec configuration of the endpoint or to remove it once the endpoint has moved to the "alternative gateway that is closer". In the case the "initial gateway" does not remove the SAs associated to the endpoint, the endpoint is considered attached simultaneously to two gateways.

The use of ADVPN with an endpoint that is added, removed or changed results in local IPsec configuration modifications. Only gateways that the endpoint is attached to are modified. Other gateways, spokes and hub are not impacted.

Gateways may accept traffic from another gateway. The traffic may be the one associated to an endpoint or to a gateway. In the first case, the gateway is considered as the "closer alternative gateway" as discussed above. The second case occurs if the "initial gateway" tunnels traffic from an "alternative gateway" to a "closer alternative gateway". It may then use ADVPN so traffic directly goes from the "alternative gateway" to the "closer alternative gateway". The IPsec configuration is then

updated on both the "alternative gateways" and the "closer alternative gateway".

Similarly, when the "closer alternative gateway" is removed, only gateways and endpoints attached to these gateways are impacted.

The use of ADVPN with a gateway that is added or removed results in local IPsec configuration modifications. Only gateways attached to are modified. Others gateways, spokes and hub are not impacted.

Spokes are between endpoints and gateways. Unlike end points, they have a complete network, and they are attached to a hub. If a spoke-to-spoke communication is set with ADVPN, then IPsec policies of the two spokes are updated. The hub may not modify its IPsec policies. Similarly, when a spoke is removed, the IPsec policies of the other spokes are updated.

The use of ADVPN with a spoke that is added or removed results in local IPsec configuration modifications. Only spokes attached to the one being removed are modified. Other gateways, spokes and hubs are not impacted.

Anytime a shortcut is established, new security policies are created on the shortcut initiator and the shortcut responders. ADVPN avoids these security policies to be created manually. In addition, it uses PSK authentication, which is, reduces latency and round trip times over other authentication methods like EAP-SIM.

Additionally, PROTECTED_DOMAIN capability can provide the initial protected domain subnet information to all its endpoints, from a trusted suggester. The trusted suggester provides periodic update on protected domain subnet information its endpoints. This periodic update, avoids requirement of any manual configuration change required, whenever new endpoint is added or existing endpoint is removed/updated, within given ADVPN protected domain.

Requirement #2 :

The solution specified in this document does not require any manual intervention for establishing a direct tunnel between endpoints. As described in Requirement #1 above and in Section 4. , SPD and SAD entries get automatically updated without any manual intervention. If an IP address of a shortcut partner has changed, MOBIKE can help in updating SPD entries automatically. If an IP address change happens after a reboot of a shortcut partner, then the peer shortcut partner will detect this condition using IKEv2 keep-alive and can divert the traffic back to the "initial-gateway". Once rebooted, the shortcut

partner will establish IPsec tunnel with the "initial-gateway". At this stage, the "initial-gateway" will send SHORTCUT exchange to the shortcut partners, to establish shortcut tunnel with new IP address of shortcut partners.

Requirement #3 :

This draft enables shortcut partners to establish a secure channel between them automatically. This will allow other tunneling and routing protocols to establish direct tunnels or exchange route updates. However, how a routing protocol module is aware of this new shortcut tunnel (or how it exchanges route updates), with shortcut partners, using shortcut tunnel or how other tunneling protocols establish direct tunnel between shortcut partners, is specific to the vendor implementation. Thus it is out of scope of this specification.

Requirement #4 :

While this document describes the syntax of SHORTCUT messages, it makes no mandates about the policy for initiating shortcuts, nor about the policy for accepting or rejecting shortcuts. Some endpoints may agree to accept shortcuts from any peer, as long as the traffic selectors are a subset of those that the SPD says should go to that peer. Others may filter the shortcuts based on IKE ID, so that they do not open tunnels to endpoints outside their administrative domain. Future documents may profile such behavior.

Requirement #5 :

When a spoke becomes compromised it may compromise inbound/outbound communications associated with it. A compromised spoke may want to use ADVPN in order to corrupt additional traffic that go through other gateways and spokes. The ADVPN protocol provides facilities to create shortcuts, however the shortcuts for given traffic is always triggered by an endpoint dealing with that traffic. As a result, a compromised host does not affect the security of other unrelated peers.

Requirement #6 :

This document addresses seamless session handoffs when endpoints roam around different policy boundaries. A detailed explanation about this is given in Section A.3.

Requirement #7 :

When a shortcut between different gateways is created for a given endpoint-to-endpoint session, the endpoint-to-endpoint communication is not impacted by the shortcut. In other words, this is transparent to the endpoints. More precisely, a new shortcut partner is created on the two alternate gateways, spokes or hubs. This modifies the communication path, but not the session itself.

Requirement #8 :

This document does not explicitly detail all NAT scenarios, in this version at least, but does provide two mechanisms that address this.

When the suggester proposes a shortcut to the shortcut peers, the suggester has performed IKE AUTH and can detect the shortcut peers are behind a NAT. This can be done with multiple ways including the NAT_DETECTION_SOURCE_IP / NAT_DETECTION_DESTINATION_IP Notify Payload exchange, the UDP encapsulation and use of port 4500. In most cases, when the shortcut peer is behind a NAT, inbound IKEv2 and IPsec traffic are sent through a specific port.

The ADVPN protocol described in this document enables the suggester to specify each shortcut peer whether the other peer is behind a NAT or not by setting the NAT bit in the ADVPN_INFO Notification. When this bit is set, it indicates UDP encapsulation MUST be used fro IKEv2. In addition, the ADVPN_INFO Notification also specifies the UDP Port on which the shortcut peer is reachable.

Another advantage of the ADVPN protocol is that the SHORTCUT exchange are sent to the shortcut peer by the suggester, and the suggester can determine whether a shortcut can be established or not. If the shortcut cannot be established, for example if the shortcut peers are both behind a NAT, then the suggester MAY forgo the establishment of the shortcut and thus avoid communication disruption due to NATs.

To address scenario where both the shortcut partners are behind NAT device, SHORTCUT exchange includes each other's peer UDP port number, that the shortcut suggester is receiving IKE and IPSec traffic. This information should help the shortcut partner to reach each other in certain types of NAT deployments.

Similarly the ADVPN protocol has designed optional exchanges that MAY in the future be designed to address specific NAT issues. For example, [MEDIATION] MAY be added for double NAT and hole punching.

Note that ADVPN is essentially based on the use of tunnel mode which makes TS selectors independent from the IP addresses of the outer header. Thus, the use of the tunnel mode makes ADVPN more resilient to NAT compared to transport mode.

Requirement #9 :

This document does not create a MIB. However, it does define several events that can be reportable:

* The gateway suggests a shortcut

* The peer accepts or rejects a shortcut (the former involves a change in policy)

* A shortcut times out (again involves a change in policy)

Requirement #10 :

The document is independent of administrative domains. One of the properties that may be associated with administrative domains is a set of one or more trust anchors used to issue certificates for VPN gateways and endpoints. To avoid the need for cross-trusting these anchors, this document offers the option of using dynamically-generated PSKs.

Requirement #11 :

While this document describes the syntax of SHORTCUT messages, it makes no mandates about the policy for initiating shortcuts, or the policy for accepting and rejecting shortcuts. Some endpoints may agree to accept shortcuts from any peer, as long as the Traffic Selectors are a subset of those that the SPD says should go to that peer. Others may filter the shortcuts based on IKE ID, so that they do not open tunnels to endpoints outside their administrative domain. Future documents may profile such behavior.

Requirement #12 :

The Traffic Selectors in the SHORTCUT message can be used to specify both multicast routing protocols, such as IGMP, and multicast traffic through the use of multicast addresses in selectors. With this, the SHORTCUT tunnels can be used to pass multicast and multicast routing traffic.

Requirement #13 :

This document defines several events that can be logged and monitored:

* The gateway suggests a shortcut

* The peer accepts or rejects a shortcut (the former involves a change in policy)

* A shortcut times out (again involving a change in policy)

A status report listing active shortcuts for a particular gateway is also possible and recommended for implementations.

Requirement #14 :

L3VPNs all use some kind of transport-layer protocol. GRE uses protocol number 43, IP-in-IP uses 4, and so on. Selectors for these protocols can easily be specified using the TS payloads included in SHORTCUTs. The additional information that may be needed to set up a tunnel for each of these protocols is outside the scope of this document.

Requirement #15 :

QoS policy is outside the scope of this document. However, the mandate of RFC 5996 to allow multiple parallel SAs for different classes of QoS applies to peers that a VPN box learns about through SHORTCUT messages. This means that QoS policy can still be enforced. If there are any additional requirements to be addressed with respect to QoS, the SHORTCUT message structure can be extended to support identified QoS attributes that should be exchanged.

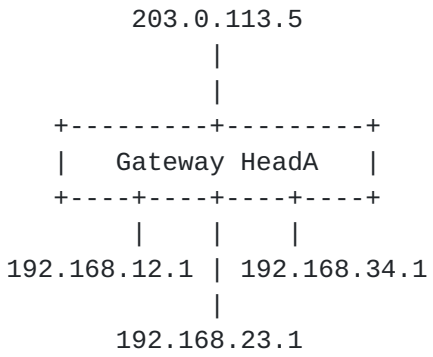
Requirements #16

ADVPN does not make spokes, hubs or gateway single points of failure. By design, ADVPN provides two types of resiliency: Topological resiliency by creating shortcuts. These shortcuts provide alternate path, and make communications resilient in case a hub or spoke fails. In addition, the use of tunnel mode between gateways makes possible the use of MOBIKE that provides end point resiliency.

Appendix C. PROTECTED_DOMAIN Example

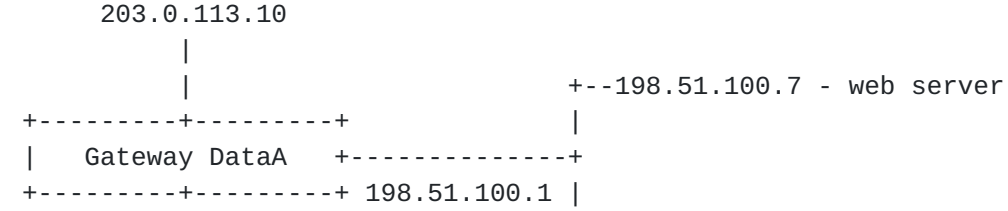
This appendix contains an example of how the PROTECTED_DOMAIN response is created. As this example requires multiple subnets, we will use the non-routable addresses from RFC 1918 in addition to the documentation subnets from RFC 5737.

Assume that Company A has two major locations. First, at the company headquarters there are three non-routable subnets: 192.168.12.0/24, 192.168.23.0/24, and 192.168.34.0/24. At this location, the VPN gateway has four interfaces: one towards each of the three internal subnets, and one external interface that connects to the Internet with IP address 203.0.113.5, as illustrated below.



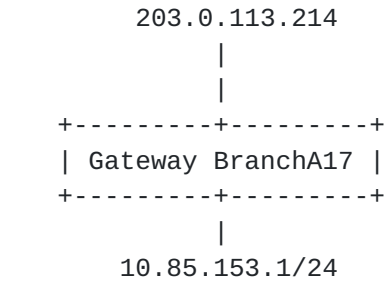
Traffic to the Internet and to partner sites gets NAT-ted, but non-routable addresses are allowed within the internal VPN.

Company A has also a datacenter, at a location different from the headquarters, which is implemented as a non-routable subnet: 192.168.45.0/24. In addition, there is also a routable subnet with some front-end servers: 198.51.100.0/24, known as the DMZ. The gateway has an external IP address 203.0.113.10. We would like access to the DMZ to go through the VPN for communications from within the company, but it can go either through the VPN or outside the VPN for traffic from partners.

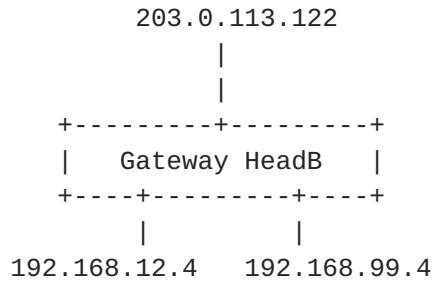


```
              |                        +--192.51.100.5 - SMTP server
        192.168.45.1
```

In addition to the two main locations introduced above, Company A also has many smaller locations. Each of those has one /24 non-routable subnet. The initial configuration of each of those small gateways is such that the internal subnet is different from that in all the other small gateways.



Company A also has a supplier, Company B, that has their own gateway with some routable and some non-routable addresses. They have a VPN tunnel configured with Gateway DataA. Although there is some overlap in the protected domains, the only non-routable addresses that go through this VPN are the 192.168.23.0/24 subnet that is behind HeadA, and the 192.168.99.0/24 that is behind HeadB. Others are either blocked, or NAT-ted behind the address of HeadB



Only the branch office gateways and Gateway HeadA need the PROTECTED_DOMAIN (see Section 3.9.) configuration attribute. Gateway HeadB has a static policy, and Gateway DataA will not send to it a PROTECTED_DOMAIN according to the established policy. The field contains the union of all the sets of addresses for which the DataA server is willing to forward traffic.

For the BranchA17 gateway, the initial configuration is very simple:

. One peer is defined: Gateway DataA (203.0.113.10)
. Either a CA certificate and a DN for DataA, or a PSK
. An internal network: 10.85.153.0/24
. Gateway DataA is a trusted suggester

In total, there are nine other branch office gateways other than BranchA17, and of course there is Gateway HeadA and Gateway HeadB. Only Gateway DataA knows about all of them. So the PROTECTED_DOMAIN it sends to other gateways from the same administrative domain is as follows:

. 192.168.12.0/24 (from behind HeadA)
. 192.168.23.0/24 (from behind HeadA)
. 192.168.34.0/24 (from behind HeadA)
. 192.168.45.0/24 (from behind DataA itself)
. 198.51.100.0/24 (DMZ behind DataA itself)
. 192.168.99.0/24 (from behind HeadB)
. 10.85.101.0/24 (from behind BranchA01)
. 10.85.104.0/24 (from behind BranchA02)
. 10.85.125.0/24 (from behind BranchA03)
. 10.85.131.0/24 (from behind BranchA04)
. 10.85.139.0/24 (from behind BranchA11)
. 10.85.143.0/24 (from behind BranchA12)
. 10.85.150.0/24 (from behind BranchA16)
. 10.85.153.0/24 (from behind BranchA17)
. 10.85.159.0/24 (from behind BranchA18)
. 10.85.162.0/24 (from behind BranchA19)

Every time a new branch gateway is added, its protected domain is added to the SPD of DataA, and this also updates the contents of the PROTECTED_DOMAIN that it sends. As described in Section 4.2. , the content expires in BranchA17 after a while, so the cache expiry time is also the time it takes for such a change to propagate to the other branch offices.

When BranchA17 receives this PROTECTED_DOMAIN, it removes its own protected domain and anything else for which it has a static configuration, and adds the rest into the SPD as traffic that is protected and tunneled to the trusted suggester (Gateway DataA).

A more complex scenario would send a different PROTECTED_DOMAIN also to the partner gateway HeadB. For example, it could send the following PROTECTED_DOMAIN:

. 192.168.23.0/24 (This is a network behind Gateway HeadA)
. 192.168.45.1/32 (This is a single address at Gateway DataA)

The reason for having this configuration is that the IP addresses behind branch office gateways are not guaranteed to be unique outside of the administrative domain. So Gateway DataA NATs these addresses behind its own IP address, which then has to be part of the protected domain. Note that under this specification, such traffic cannot be "shortcutted", because we don't have a way to tell the BranchA17 gateway to NAT these packets when sending them through the shortcut tunnel (TBD)

Authors' Addresses

Praveen Sathyanarayan
Juniper Networks, Inc.
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
USA

Email: praveenys@juniper.net

Steve Hanna
Juniper Networks, Inc.
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
USA

Email: shanna@juniper.net

Suresh Nagavenkata Melam
Juniper Networks, Inc.
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
USA

Email: nmelam@juniper.net

Yoav Nir
Check Point Software Technologies Ltd.
5 Hasolelim st.

Tel Aviv 6789735
Israel

Email: ynir@checkpoint.com

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

Email: mglt.ietf@gmail.com

Kostas Pentikousis
EICT GmbH
Torgauer Strasse 12-15
10829 Berlin
Germany

Email: k.pentikousis@eict.de