IPSECME Working Group
Internet-Draft
Updates: 3948 7296 (if approved)
Intended status: Standards Track
Expires: 26 November 2026

A. Antony
S. Klassert
secunet

25 May 2026

Multiple UDP Source Ports for ESP in UDP Encapsulation
draft-antony-ipsecme-udp-encap-multiport-00

Abstract

This document specifies a mechanism to improve network path distribution and host receive-queue load distribution for IPsec traffic using ESP in UDP encapsulation [RFC3948]. Using the per-resource Child SA mechanism of [RFC9611], peers negotiate multiple Child SAs each bound to a distinct UDP source port. The resulting variation in UDP source port enables receive-side scaling (RSS) and equal-cost multi-path (ECMP) load balancing, supporting efficient per-CPU IPsec processing. This document specifies the IKEv2 negotiation, NAT traversal behavior, and operational requirements for this mechanism.

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 https://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 26 November 2026.

Copyright Notice

Copyright (c) 2026 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 (https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

In high-speed IPsec deployments, endpoints exchange traffic at multi-gigabit rates and must distribute cryptographic processing across multiple CPU cores. ESP in UDP encapsulation [RFC3948] is widely deployed in cloud environments and across NAT gateways. However, when ESP is encapsulated in UDP using port 4500 for both source and destination, all traffic between a given pair of peers shares a single 4-tuple (src-IP, dst-IP, src-port=4500, dst-port=4500). This eliminates the 4-tuple diversity required for effective NIC receive-side scaling (RSS) and ECMP path selection.

This document specifies a mechanism whereby IKEv2 peers establish multiple Child Security Associations (SAs), each bound to a distinct UDP source port, using the per-resource Child SA mechanism of [RFC9611]. Each per-resource Child SA is created via a CREATE_CHILD_SA exchange sent from a new ephemeral UDP source port. The resulting UDP flows, with varying source ports, enable NIC hardware and network infrastructure to distribute IPsec traffic across RSS queues and ECMP paths. A Fallback SA on the standard port pair (4500 to 4500) is always maintained per [RFC9611]. This mechanism is defined for ESP [RFC4303] in UDP encapsulation [RFC3948]; its applicability to EESP [I-D.ietf-ipsecme-eesp][I-D.ietf-ipsecme-eesp-ikev2] is discussed in Section 9.

Varying the UDP source port without IKEv2 coordination is insufficient. Without a negotiated binding between a UDP source port and a specific Child SA, the responder cannot distinguish an intentional port change from a NAT remapping event, which would trigger IKE SA roaming procedures per [RFC7296] Section 2.23. NAT keepalives ([RFC3948] Section 6) must be maintained per active port pair; without IKEv2 signaling, the IKEd has no record of which port pairs exist. NIC and kernel queue-steering rules require both peers to agree on the port-to-resource binding; without negotiation, consistent steering configuration across peers is not achievable. This document specifies the IKEv2 exchanges and behavioral rules that establish deterministic port-to-SA bindings, providing the coordination that unilateral port variation cannot.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Terminology

This document uses the following terms from IKEv2 [RFC7296]: Child SA, CREATE_CHILD_SA exchange, IKE_AUTH exchange, INFORMATIONAL exchange.

This document uses the following terms from [RFC3948]: UDP-encapsulated ESP, Non-ESP Marker.

This document uses the following terms defined in [RFC9611]: per-resource Child SA, Resource, SA_RESOURCE_INFO, TS_MAX_QUEUE.

Fallback SA The standard UDP-encapsulated ESP Child SA using UDP source port 4500 and destination port 4500, established during IKE_AUTH. It remains active for the lifetime of the IKE SA.

Per-Resource Child SA A Child SA established via CREATE_CHILD_SA from an Ephemeral Source Port, bound to that port for data-plane entropy and traffic-steering purposes. In this document, the resource is a CPU core or NIC receive queue.

Ephemeral Source Port A UDP source port selected by the IKEd for a per-resource Child SA, distinct from port 4500 and from the source ports of all other active per-resource Child SAs.

IKEd The IKEv2 implementation on a host responsible for IKE SA and Child SA lifecycle management.

TBD1 The IKEv2 Notify Message Status Type defined in this document that signals support for the UDP Ephemeral Source Port mechanism. A peer including TBD1 in IKE_AUTH implicitly signals support for the per-resource Child SA mechanism of [RFC9611]. See Section 10.

2. Problem Statement

ESP in UDP encapsulation [RFC3948] deploys ESP packets in UDP with source port 4500 and destination port 4500. Because all IPsec traffic between two peers shares this single 4-tuple, no port entropy is present in the outer UDP header.

Modern NIC hardware uses the outer UDP 4-tuple for RSS queue assignment. Without source port entropy, all IPsec traffic between two peers is directed to a single NIC RSS queue and processed by a single CPU core, creating a throughput bottleneck even when multiple cores are available.

Native ESP carries the SPI at a fixed header offset and can serve as an ntuple steering key for per-resource flow distribution. EESP [I-D.ietf-ipsecme-eesp] can carry explicit resource identifiers. However, support for ESP SPI and EESP resource identifier filtering in current network devices is limited. UDP source and destination port ntuple filtering scales well and is broadly supported across current NIC drivers and network equipment, making ESP in UDP encapsulation the practical foundation for per-resource flow steering.

Multi-path networks using ECMP similarly rely on flow 5-tuple entropy to spread traffic across links. A single UDP flow between two peers concentrates all traffic on one ECMP path, underutilizing available bandwidth.

The IPv6 flow label [RFC6438] addresses load distribution for tunnel traffic in IPv6 environments. It does not apply to ESP-in-UDP deployments, which are used specifically where NAT traversal is required. NAT devices do not preserve the IPv6 flow label, and many such deployments remain on IPv4.

Varying the UDP source port per CPU or per NIC queue resolves both problems. Each per-resource Child SA has a distinct UDP source port, providing the entropy needed for RSS and ECMP distribution without modifying the inner ESP payload or changing traffic selectors. Each per-resource Child SA also maintains an independent ESP sequence number counter and replay window, eliminating cross-CPU synchronization of cryptographic state.

3. Solution Overview

Two IKEv2 peers first establish a standard IKE SA and a Fallback SA using UDP-encapsulated ESP on port 4500. Both peers signal support for this mechanism by including TBD1 (see Section 6.1) in the IKE_AUTH exchange.

When per-resource Child SAs are desired, the initiator sends a CREATE_CHILD_SA exchange from a new ephemeral UDP source port, including SA_RESOURCE_INFO per [RFC9611]. The responder treats the resulting Child SA as a per-resource Child SA bound to that port tuple. The responder MUST send the CREATE_CHILD_SA response back to the same source port and IP address from which the request was received, using its own port 4500 as the source. All other IKE communication continues on the main port pair (4500 to 4500).

The initiator MAY request additional per-resource Child SAs via further CREATE_CHILD_SA exchanges. If the responder is unwilling to create more per-resource Child SAs for the Traffic Selector pair, it returns TS_MAX_QUEUE per [RFC9611]. The Fallback SA remains active throughout.

The initiator MUST NOT send CREATE_CHILD_SA from an Ephemeral Source Port unless both peers have exchanged TBD1 in the IKE_AUTH exchange. Without this exchange, a CREATE_CHILD_SA from a non-4500 source port would be misinterpreted by the responder as a NAT mapping change per [RFC7296] Section 2.23, updating the IKE SA peer port and disrupting all subsequent IKE communication.

4. Updates to RFC3948 and RFC7296

4.1. Update to RFC3948

[RFC3948] Section 2.1 requires that the UDP Source Port and Destination Port of ESP-in-UDP packets "MUST be the same as that used by IKE traffic."

This document updates that requirement as follows. When two IKEv2 peers have enabled the mechanism defined in this document by exchanging TBD1 in the IKE_AUTH exchange, ESP-in-UDP packets belonging to a per-resource Child SA MAY use a UDP source port different from the source port used for IKE traffic. The UDP source port for such packets MUST be the Ephemeral Source Port bound to that per-resource Child SA as negotiated in Section 6.

This relaxation applies only to per-resource Child SAs negotiated per this document. The Fallback SA and all other Child SAs MUST continue to use the same port as IKE traffic, as required by [RFC3948].

4.2. Update to RFC7296

[RFC7296] Section 2.23 requires that "The peer MUST also send all subsequent IKEv2 traffic on UDP port 4500."

[RFC7296] Section 2.11 already requires that a responder MUST accept IKEv2 requests regardless of the UDP source port and reply to the address and port from which the request was received. The responder-side behavior required by this document therefore needs no change to existing implementations.

This document updates the initiator-side requirement of Section 2.23. When the mechanism defined in this document is in use, CREATE_CHILD_SA exchanges used to negotiate per-resource Child SAs MAY be sent from an Ephemeral Source Port other than 4500. The responder MUST reply to the same Ephemeral Source Port from its own port 4500.

All other IKEv2 traffic, including INFORMATIONAL exchanges, the IKE SA, and all exchanges not related to per-resource Child SA negotiation, MUST continue to use port 4500 as required by [RFC7296].

5. Fallback SA

The Fallback SA is the initial Child SA established during the IKE_AUTH exchange using UDP source port 4500 and destination port 4500, following [RFC3948] and [RFC7296]. It serves the role of the shared Child SA described in [RFC9611]: a single SA usable by all resources while per-resource Child SAs are being negotiated or when no per-resource Child SA exists for a given resource.

The Fallback SA MUST remain active for the lifetime of the IKE SA. It MUST NOT be deleted while per-resource Child SAs are active. IKE control messages, rekeying exchanges, and deletion messages for per-resource Child SAs MUST be sent using the Fallback SA’s port pair (4500 to 4500).

6. Per-Resource Child SA Negotiation

6.1. Capability Announcement

Support for the UDP Ephemeral Source Port mechanism defined in this document is signaled by including the TBD1 notification in the IKE_AUTH exchange. Both peers MUST include TBD1 to enable the mechanism. If either peer omits TBD1 from IKE_AUTH, the initiator MUST NOT send CREATE_CHILD_SA from an Ephemeral Source Port; both peers MUST use the Fallback SA for all traffic.

TBD1 has no notification data.

6.2. Creating Per-Resource Child SAs

To create a per-resource Child SA, the initiator IKEd opens a new UDP socket bound to an Ephemeral Source Port and sends a CREATE_CHILD_SA exchange from that port to the responder’s port 4500. The CREATE_CHILD_SA exchange MUST include an SA_RESOURCE_INFO notification per [RFC9611].

The Ephemeral Source Port MUST be selected from the dynamic port range (49152-65535) per [RFC6056] and MUST NOT be a well-known port (0-1023). The port MUST be distinct from port 4500 and from the source ports of all currently active per-resource Child SAs. The port SHOULD be selected randomly within the dynamic range per [RFC6056]. Because the port value is exchanged in the IKE handshake and bound to an SA known to both peers, randomization does not provide confidentiality; it prevents predictable allocation patterns that expose implementation state.

The IKEd MUST retain the socket binding to the Ephemeral Source Port for the lifetime of the SA, preventing the operating system from assigning that port to other applications.

The initiator SHOULD create one per-resource Child SA per CPU core or NIC receive queue available for IPsec processing, up to the limit indicated by TS_MAX_QUEUE ([RFC9611]). Creating additional per-resource Child SAs beyond available resources provides no benefit and increases IKE state on both peers.

6.3. Responder Behavior

Upon receiving a CREATE_CHILD_SA containing SA_RESOURCE_INFO from a new UDP source port, and having exchanged TBD1 in IKE_AUTH, the responder MUST:

1. Respond to the initiator’s Ephemeral Source Port from its own port 4500.

2. Install the Child SA with the IP and port tuple (initiator-IP, responder-IP, Ephemeral-Source-Port,

    1. as the UDP binding.

3. NOT update the IKE SA’s IP address or port based on this message. Per-resource Child SA creation from a new source port MUST NOT be interpreted as IKE SA roaming or NAT mapping change.

6.4. Implementation Considerations

The IKEd MUST open a socket bound to the Ephemeral Source Port only when initiating a CREATE_CHILD_SA exchange from that port. The socket MUST NOT be opened speculatively or in advance of the exchange.

During the CREATE_CHILD_SA exchange, the IKEd MUST only accept IKEv2 messages received on the Ephemeral Source Port socket that carry the IKE SA cookies (initiator and responder SPIs) of the IKE SA under

which the Child SA is being negotiated. Messages with unknown or mismatched IKE SA cookies MUST be silently discarded. This prevents an attacker from injecting IKEv2 messages via the ephemeral port.

After the CREATE_CHILD_SA exchange completes, the IKEd MUST retain the socket binding to prevent the operating system from assigning the port to another application, but MUST NOT process further IKEv2 messages received on the ephemeral port. All subsequent IKE traffic for the Child SA uses the Fallback SA's port pair (4500 to 4500).

6.5. Path Validation

Completion of the CREATE_CHILD_SA exchange does not establish that the data path for a per-resource Child SA is viable. A NAT gateway may silently drop ESP traffic on the new port pair even when the IKE exchange succeeded. Forwarding traffic on an unconfirmed path will result in blackholing.

The responder MUST install only the inbound SA upon completing the CREATE_CHILD_SA exchange. Installation of the outbound SA MUST be deferred until data-plane reachability is confirmed.

Data-plane reachability is confirmed when the responder receives the first ESP packet on the new inbound SA. The SAD MAY enforce a soft limit of one incoming packet on the inbound SA; when this limit triggers, the kernel signals the IKEd (e.g., via an XFRM acquire event), which then installs the outbound SA.

Alternatively, the initiator MAY send an encrypted ESP ping ([I-D.ietf-ipsecme-encrypted-esp-ping]) immediately after the CREATE_CHILD_SA exchange completes, providing explicit confirmation of data-plane reachability to the responder.

Until the outbound SA is installed, the responder MUST use the Fallback SA for traffic destined to the initiator.

6.6. NIC Queue Steering

When a per-resource Child SA is established, each peer programs its NIC or kernel packet classifier to steer incoming ESP traffic for that UDP port pair to the target CPU or queue.

Because the same Ephemeral Source Port appears in different header fields on each side, the steering rules are asymmetric:

* On the initiator: incoming ESP traffic from the responder arrives with dst-port = Ephemeral-Source-Port. Steer on dst-port = Ephemeral-Source-Port.

* On the responder: incoming ESP traffic from the initiator arrives with src-port = Ephemeral-Source-Port. Steer on src-port = Ephemeral-Source-Port.

Example using ethtool ntuple rules, where the Ephemeral Source Port is 50001 and queue index is 20:

```
On initiator:
  ethtool --config-ntuple eth0 flow-type udp4 \
    src-port 4500 dst-port 50001 action 20

On responder:
  ethtool --config-ntuple eth0 flow-type udp4 \
    src-port 50001 dst-port 4500 action 20
```

Figure 1: NIC Steering Rules (Ephemeral Source Port 50001)

7. NAT Traversal Considerations

The design requires that only the initiator selects the Ephemeral Source Port for a per-resource Child SA. If both peers were to independently choose their own ephemeral ports, the responder would install the Child SA bound to the initiator's private address before any traffic has flowed. When a NAT is present, the responder does not yet know the NAT-translated address and port for the new flow: no mapping exists until the initiator sends the first packet. The responder may also have no route to the initiator's private address and cannot send traffic until the NAT mapping is established. By requiring the initiator to select the port and send first, the NAT mapping is created before the responder installs the outbound SA, avoiding this failure mode.

7.1. Initiator Behind NAT

When the initiator A is behind a NAT gateway N, and A creates a per-resource Child SA from Ephemeral Source Port P:

```
A:P --> NAT --> N:Q --> B:4500   (initiator to responder)
B:4500 --> N:Q --> A:P           (responder to initiator)
```

Figure 2: Initiator-Behind-NAT Port Mapping

The NAT gateway creates a new mapping for source port P, translating it to external port Q. The responder B receives CREATE_CHILD_SA from N:Q and responds to N:Q. The per-resource Child SA's port binding at the responder is (N:Q, B:4500). No special handling is required; the standard procedure of Section 6.2 applies.

7.2. No NAT

When there is no NAT between peers, per-resource Child SA creation proceeds as described in Section 6.2. IP and port tuples are used directly for NIC steering and SAD lookups.

The source and destination ports are symmetric in the ESP flow, as illustrated for Ephemeral Source Port 50001:

```
A:50001 --> B:4500   (A to B ESP traffic)
B:4500  --> A:50001  (B to A ESP traffic)
```

Figure 3: Port Tuples without NAT

7.3. Bidirectional NAT

Some NAT deployments (e.g., certain cloud environments) allow mapping creation from either direction. In such environments, the responder MAY initiate per-resource Child SA creation using its own Ephemeral Source Port, with the NAT gateway creating the necessary mapping. The procedure is identical to the initiator case and no special handling is required.

7.4. Responder-Initiated SA Blocked by NAT

When the responder B initiates a per-resource Child SA from a new Ephemeral Source Port and the NAT gateway does not support mapping creation in the B-to-A direction, the CREATE_CHILD_SA request is silently dropped. After retransmission attempts are exhausted per [RFC7296] Section 2.1, B MUST abandon the attempt.

A dropped CREATE_CHILD_SA leaves the IKE Message ID sequence in an inconsistent state. B MUST recover by sending an INFORMATIONAL exchange over the main IKE SA (UDP port 4500 to 4500), containing both an IKEV2_MESSAGE_ID_SYNC notification ([RFC6311] Section 5.1) and a Delete payload ([RFC7296] Section 3.11) carrying the SPI that B proposed in the failed CREATE_CHILD_SA.

```
INF( N(IKEV2_MESSAGE_ID_SYNC),
     D(ESP, SPI) )
```

Figure 4: INFORMATIONAL for Abandoned Per-Resource Child SA

Multiple SPIs MAY be carried in a single Delete payload when several per-resource Child SA attempts are abandoned.

On receiving this INFORMATIONAL, A processes IKEV2_MESSAGE_ID_SYNC per [RFC6311] and processes the Delete payload per [RFC7296] Section 3.11. If A has installed a Child SA for the indicated SPI, A MUST delete it. If the SPI is unknown to A, A silently ignores it per [RFC7296] Section 3.11.

B MUST be prepared to receive a delayed CREATE_CHILD_SA response even after sending this INFORMATIONAL. If such a response arrives and B installs the Child SA, B MUST delete it immediately.

B MAY retry per-resource Child SA creation from a different Ephemeral Source Port, as individual ports may be selectively blocked by NAT policy. B SHOULD cease responder-initiated per-resource Child SA creation after repeated consecutive failures and rely on A to create additional per-resource Child SAs.

7.5. NAT Mapping Change

NAT mapping changes affecting per-resource Child SAs fall into two cases.

When the peer’s IP address changes (e.g., after network roaming), MOBIKE [RFC4555] or the [RFC7296] Section 2.23 address-change procedure detects the change on the Fallback SA’s port pair (4500 to 4500). Per-resource Child SAs have no independent IKE channel and rely entirely on the Fallback SA for detection. Upon completing a MOBIKE UPDATE_SA_ADDRESSES exchange, the IKEd MUST delete all per-resource Child SAs associated with the affected IKE SA and SHOULD recreate them via CREATE_CHILD_SA exchanges from the new source address, following Section 6.2. Path validation (Section 6.5) MUST be performed for each new per-resource Child SA before its outbound SA is installed. Until recreation is complete, the Fallback SA MUST be used for all traffic.

When only an ephemeral port mapping changes (the IP address remains the same but the NAT gateway remaps a specific ephemeral port), the Fallback SA is unaffected and MOBIKE does not fire. Detection relies on NAT keepalive failure for that port pair (Section 8.1), DPD (Section 8.2), or path validation (Section 6.5) timeout on the affected per-resource Child SA. Upon detecting the failure, the IKEd SHOULD delete the affected per-resource Child SA and recreate it via a new CREATE_CHILD_SA exchange.

7.6. NAT Mapping Loss

A NAT gateway reboot or mapping table reset silently invalidates all per-resource Child SA port mappings. The Fallback SA is more resilient: IKE keepalives on the 4500 to 4500 port pair will naturally re-establish the NAT mapping on the first exchange after the reboot. Per-resource Child SAs on ephemeral ports have no independent keepalive that recreates their NAT mapping. Once a mapping is lost, inbound ESP traffic for those SAs is silently dropped.

The IKEd SHOULD detect the failure via the DPD procedure described in Section 8.2 or via path validation (Section 6.5), delete the affected per-resource Child SAs, and create replacements via CREATE_CHILD_SA exchanges sent from the Fallback SA's port pair (4500 to 4500). The first such exchange will re-establish the NAT mapping for the new Ephemeral Source Port.

8. Operational Considerations

8.1. NAT Keepalives

When NAT traversal keepalives are required ([RFC3948] Section 6), a one-byte NAT keepalive packet MUST be sent for every active UDP source and destination port pair, not only for the Fallback SA's port pair (4500 to 4500).

If N per-resource Child SAs and one Fallback SA are active, N+1 independent keepalive flows MUST be maintained, one per unique (src-IP, dst-IP, src-port, dst-port) tuple.

8.2. Dead Peer Detection and Liveness

Liveness checking MAY be performed per per-resource Child SA port pair, or only on the Fallback SA port pair (4500 to 4500), as a local policy choice.

If a liveness failure is detected on a per-resource Child SA path, only that SA and its associated port pair SHOULD be considered failed. The IKEd SHOULD delete the failed per-resource Child SA and MAY create a replacement.

If a liveness failure is detected on the Fallback SA, all per-resource Child SAs associated with the same IKE SA SHOULD be considered failed, and the IKE SA teardown procedure ([RFC7296] Section 1.4) applies.

8.3. Child SA Rekeying

Rekeying of per-resource Child SAs MUST be initiated via the main IKE SA, using port pair 4500 to 4500. This ensures rekeying messages are not affected by per-resource Child SA path failures.

The rekeyed Child SA MUST reuse the same Ephemeral Source Port as the SA being rekeyed, preserving the UDP binding and NIC queue steering configuration.

8.4. Deletion

Delete exchanges for per-resource Child SAs MUST be sent via the main IKE SA port pair (4500 to 4500), ensuring delivery even when the per-resource Child SA path is no longer viable.

9. EESP Considerations

This mechanism applies equally to EESP [I-D.ietf-ipsecme-eesp][I-D.ietf-ipsecme-eesp-ikev2] when Sub SAs are not in use. Each per-resource Child SA is a separate EESP Child SA with its own SPI negotiated via CREATE_CHILD_SA, and [RFC9611] applies identically to the ESP case.

When EESP Sub SAs are in use (an SSKDF transform is negotiated), the mechanism defined in this document does not apply. Sub SAs are derived from a parent EESP SA and have no independent SPIs or IKEv2 lifecycle; they do not participate in CREATE_CHILD_SA exchanges and cannot be bound to an Ephemeral Source Port.

Note: if a future revision of EESP Sub SA negotiation includes support for resource binding and UDP source port assignment, the per-resource distribution function provided by this document could be subsumed into the base Sub SA mechanism, eliminating the need for separate CREATE_CHILD_SA exchanges per resource.

10. IANA Considerations

This document requests IANA to assign a value for TBD1 in the "IKEv2 Notify Message Status Types" registry:

Value	Notify Message Status Type	Reference
TBD1	UDP_EPHEMERAL_SOURCE_PORT	This document

Table 1

11. Security Considerations

Per-resource Child SAs have independent key material, inheriting the security properties of ESP-in-UDP [RFC3948]. The Ephemeral Source Port provides entropy in the outer UDP header but carries no cryptographic material.

The path validation requirement (see Section 6.5) ensures that traffic is not forwarded on an SA whose data path has not been confirmed. Bypassing path validation risks traffic blackholing when paths are blocked by NAT or firewall policy.

The abandoned-SA recovery procedure in Section 7.4 uses a standard Delete payload over the main IKE SA. Implementations MUST handle a delayed CREATE_CHILD_SA response arriving after the recovery INFORMATIONAL has been sent, as specified in that section.

UDP source port variation increases the set of flows observable by on-path devices. ESP encryption and integrity protection prevent payload manipulation, but per-flow traffic analysis based on port patterns remains possible. The varying source port is a performance mechanism; it MUST NOT be relied upon as a security mechanism.

12. Acknowledgments

This document evolved from discussions at several IETF meetings and from review of [I-D.xu-ipsecme-esp-in-udp-lb]. The authors thank the IPSECME working group participants for their input and feedback, with particular thanks to Valery Smyslov, Tero Kivinen, Paul Wouters, and Paul Bottorff.

13. Normative References

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

[RFC3948] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, DOI 10.17487/RFC3948, January 2005, <https://www.rfc-editor.org/info/rfc3948>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

[RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-Protocol Port Randomization", BCP 156, RFC 6056, DOI 10.17487/RFC6056, January 2011, <https://www.rfc-editor.org/info/rfc6056>.

[RFC6311] Singh, R., Ed., Kalyani, G., Nir, Y., Sheffer, Y., and D. Zhang, "Protocol Support for High Availability of IKEv2/IPsec", RFC 6311, DOI 10.17487/RFC6311, July 2011, <https://www.rfc-editor.org/info/rfc6311>.

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

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC9611] Antony, A., Brunner, T., Klassert, S., and P. Wouters, "Internet Key Exchange Protocol Version 2 (IKEv2) Support for Per-Resource Child Security Associations (SAs)", RFC 9611, DOI 10.17487/RFC9611, July 2024, <https://www.rfc-editor.org/info/rfc9611>.

14. Informative References

[I-D.ietf-ipsecme-eesp]
Klassert, S., Antony, A., and C. Hopps, "Enhanced Encapsulating Security Payload (EESP)", Work in Progress, Internet-Draft, draft-ietf-ipsecme-eesp-03, 2 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-ipsecme-eesp-03>.

[I-D.ietf-ipsecme-eesp-ikev2]
Klassert, S., Antony, A., Brunner, T., and V. Smyslov, "IKEv2 negotiation for Enhanced Encapsulating Security Payload (EESP)", Work in Progress, Internet-Draft, draft-ietf-ipsecme-eesp-ikev2-02, 2 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-ipsecme-eesp-ikev2-02>.

[I-D.ietf-ipsecme-encrypted-esp-ping]
Antony, A. and S. Klassert, "Encrypted ESP Echo Protocol", Work in Progress, Internet-Draft, draft-ietf-ipsecme-encrypted-esp-ping-03, 4 May 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-ipsecme-encrypted-esp-ping-03>.

[I-D.xu-ipsecme-esp-in-udp-lb]
Xu, X., Hegde, S., Pismenny, B., Zhang, D., Xia, L., and M. Puttaswamy, "Encapsulating IPsec ESP in UDP for Load-balancing", Work in Progress, Internet-Draft, draft-xu-ipsecme-esp-in-udp-lb-15, 26 February 2026, <https://datatracker.ietf.org/doc/html/draft-xu-ipsecme-esp-in-udp-lb-15>.

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

[RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <https://www.rfc-editor.org/info/rfc6438>.

Authors' Addresses

Antony Antony
secunet Security Networks AG
Email: antony.antony@secunet.com

Steffen Klassert
secunet Security Networks AG
Email: steffen.klassert@secunet.com