IPSECME Working Group
Internet-Draft
Intended status: Standards Track
Expires: 20 March 2026

S. Klassert
A. Antony
secunet
T. Brunner
codelabs GmbH
V. Smyslov
ELVIS-PLUS

16 September 2025

IKEv2 negotiation for Enhanced Encapsulating Security Payload (EESP) draft-ietf-ipsecme-eesp-ikev2-01

Abstract

This document specfies how to negotiate the use of the Enhanced Encapsulating Security Payload (EESP) protocol using the Internet Key Exchange protocol version 2 (IKEv2). The EESP protocol, which is defined in draft-klassert-ipsecme-eesp, provides the same security services as Encapsulating Security Payload (ESP), but has richer functionality and provides better performance in specific circumstances. This document specifies negotiation of version 0 of EESP.

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 20 March 2026.

Copyright Notice

Table of Contents

1. Introduction

The Enhanced Encapsulating Security Payload (EESP), specified in [I-D.klassert-ipsecme-eesp], introduces enhancements to the Encapsulating Security Payload (ESP) defined in [RFC4303]. These improvements address evolving requirements in modern IPsec deployments. EESP offers increased flexibility for hardware offloads at the packet level. It supports carrying inner packet flow identifiers for the use with ECMP, RSS hardware, and IPsec peers prior to decryption. EESP also enables the establishment of Sub SAs with independent keys and sequence number spaces. Additionally, it supports the use of 64-bit sequence numbers transmitted in each packet or the omission of sequence numbers when the Replay Protection service is not needed or cannot be achieved (e.g. in some multicast scenarios). EESP packets carry a version number, enabling easier support for future extensions.

This document specifies the negotiation of EESP Security Associations (SAs) within the Internet Key Exchange Protocol Version 2 (IKEv2) protocol [RFC7296]. It details the creation, rekeying, and deletion of EESP SAs, as well as the negotiation of EESP specific transforms and properties.

The extensions defined here enables EESP SAs to coexist with ESP SAs, while introducing new capabilities to enhance IPsec’s performance and versatility in modern use cases.

This document does not obsolete or update any existing RFCs. While stateless implementations of EESP are referenced, their negotiation, which is similar to [PSP], is outside the scope of this document.

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

It is assumed that readers are familiar with IKEv2 [RFC7296], IPsec architecture [RFC4301] and ESP [RFC4303]. This document uses a notation and conventions from IKEv2 [RFC7296].

This document uses the following terms defined in [RFC2992]: Equal-cost multi-path (ECMP)

2. EESP Overview

2.1. EESP Version

Each EESP packet carries the EESP Base Header version, which is specified in Section [XXX] of [I-D.klassert-ipsecme-eesp]. EESP version determines the format of the EESP packet and its processing rules. The initial version specified in [I-D.klassert-ipsecme-eesp] is version 0.

2.2. EESP Sub SA

Existing mechanisms for establishing Child SAs, as described in [RFC7296], yield pair of SAs. High-speed IP traffic is often asymmetric. Creating multiple pairs of Child SAs, e.g. for multiple resources ([RFC9611]), or for DSCP to carry asymmetric traffic, is inefficient.

To deal with this limitations EESP introduces the concept of Sub SAs. An EESP Sub SA is an unidirectional SA derived from the same-direction EESP SA of the pair of SAs negotiated using IKEv2. Each Sub SA derives its own keying material and maintains independent sequence number spaces and IV spaces from the parent Child SA; all other characteristics of Sub SAs (algorithms, traffic selectors etc.) are identical to the EESP SA they belong to.

Each Sub SA is identified by a Sub SA Id, which is a number in the range from zero up to the maximum Sub SA ID indicated by the receiver of the Sub SA during EESP SA negotiation via IKEv2 (see Section 3.5). The Sub SA ID is carried in the Session ID field of the EESP base header (see Section [XXX] of [I-D.klassert-ipsecme-eesp]). The Sub SA ID MUST be unique for all Sub SAs in the context of an EESP SA.

If a peer receives an EESP packet with a Sub SA ID greater than the maximum value it announced during EESP SA setup, that peer MAY drop this packet without further processing.

Use of Sub SAs is optional in EESP and can be negotiated using IKEv2.

2.3. EESP Sequence Numbers

Unlike ESP, the EESPv0 header transmits 64-bit sequence numbers if replay protection is used. In addition, the Sequence Number field in the EESPv0 header is optional and can be omitted from the packet if replay protection is not needed. Note that while possible, disabling replay protection is generally NOT RECOMMENDED and should only be done in case of multicast scenarios or if the upper level protocol provides this service. See Section 7 for details.

3. EESP SA Negotiation in IKEv2

Current EESP specification [I-D.klassert-ipsecme-eesp] defines version 0 of the EESP protocol. Consequently, this document limits its scope to only deal with EESPv0. If other EESP versions are defined in future, their negotiation using IKEv2 should be covered by separate documents.

EESP Security Associations (SAs) are negotiated in IKEv2 similarly to ESP SAs - as Child SAs in the IKE_AUTH or the CREATE_CHILD_SA exchanges. For this purpose a new Security Protocol Identifier EESPv0 (<TBD1>) is defined. This protocol identifier is placed in the Protocol ID field of the Proposal Substructure in the SA Payload when peers negotiate EESP version 0. It is possible for the initiator to include both ESP and EESPv0 proposals in the SA payload to negotiate either ESP or EESP.

3.1. EESP Specific Transform Types and Transform IDs

3.1.1. Sub SA Key Derivation Function Transform

This document defines a new Sub SA Key Derivation Function (SSKDF) transform type, that is used to negotiate a key derivation function for Sub SAs as described in Section 2.2.

This document creates a new IKEv2 IANA registry for the Key Derivation Functions transform IDs. The initially defined Transform IDs are listed in the table below.

Value	Algorithm
0	NONE
1	SSKDF_HKDF_SHA2_256
2	SSKDF_HKDF_SHA2_384
3	SSKDF_HKDF_SHA2_512
4	SSKDF_AES256_CMAC

Table 1: Sub SA Key Derivation Functions

These algorithms are defined as follows:

* SSKDF_HKDF_SHA2_256, SSKDF_HKDF_SHA2_384 and SSKDF_HKDF_SHA2_512 use HKDF-Expand defined in [RFC5869] with the indicated hash functions, that is, SHA-256, SHA-384 or SHA-512, respectively, with corresponding key sizes of 32, 48 and 64 octets. SSKDF is then defined as:

  SSKDF(K, S, L) = HKDF-Expand(K, S, L)

* SSKDF_AES256_CMAC is currently undefined

Other key derivation functions may be added after the publication of this document. Readers should refer to [IKEv2-IANA] for the latest values.

The type of the Sub SA Key Derivation Function transform is <TBA2>.

3.1.2. New Transform IDs for Sequence Numbers Transform Type

This document defines two new Transform IDs for the Sequence Numbers transform type: ’64-bit Sequential Numbers’ (<TBD5>) and ’None’ (<TBD6>).

To enable the presence of sequence numbers in the EESP header and enabling replay protection, the initiator MUST propose SN = (64-bit Sequential Numbers) in the Proposal Substructure inside the Security Association (SA) payload. When the responder selects 64-bit Sequential Numbers, the Sequence Number field MUST be included into the EESP header and peers MUST perform replay protection.

To disable sequence numbering, and thus replay protection based on sequence numbers, the initiator MUST propose SN=None (<TBD6>). When the responder selects None, the Sequence Number field is omitted from the EESP header.

3.2. Transforms Consistency

IKEv2 limits transform types that can appear in the Proposal substructure based on its Protocol ID field (see Section 3.3.3 of [RFC7296]). For EESPv0 the following transform types are allowed:

Protocol	Mandatory Types	Optional Types
EESPv0	ENCR, SN	KE, SSKDF

Table 2

For the ENCR transform type only those transform IDs that define use of AEAD cipher mode are allowed in case of EESPv0. Transform IDs that define pure encryption MUST NOT be used in the context of EESPv0.

Note, that '64-bit Sequential Numbers' and 'None' transform IDs are unspecified for ESP and MUST NOT be used in ESP proposals. On the other hand, currently defined transform IDs for the Sequence Numbers transform type (32-bit Sequential Numbers and Partially Transmitted 64-bit Sequential Numbers) are unspecified for EESPv0 and MUST NOT be used in EESPv0 proposals.

Implementations MUST ignore transforms containing invalid values for the current proposal (as if they are unrecognized, in accordance with Section 3.3.6 of [RFC7296]).

The use of the 'None' Transform ID for the SN transform if further limited by the ENCR transform. In particular, if the selected ENCR transform defines use of implicit IV (as transforms defined in [RFC8750]), then the value 'None' MUST NOT be selected for the SN transform.

3.3. Example of SA Payload Negotiating EESP

Below is the example of SA payload for EESP negotiation.

```
SA Payload
   |
   +--- Proposal #1 ( Proto ID = EESPv0(<TBD1>), SPI size = 4,
   |     |            5 transforms,      SPI = 0x052357bb )
   |     |
   |     +-- Transform ENCR ( Name = ENCR_AES_GCM_16 )
   |     |     +-- Attribute ( Key Length = 256 )
   |     +-- Transform ENCR ( Name = ENCR_AES_GCM_16 )
   |     |     +-- Attribute ( Key Length = 128 )
   |     +-- Transform SSKDF ( Name = SSKDF_HKDF_SHA2_256 )
   |     +-- Transform SSKDF ( Name = SSKDF_HKDF_SHA2_512 )
   |     +-- Transform SN ( Name = 64-bit Sequential Numbers )
```

Figure 1: EESPv0 SA proposal

3.4. Use of Notifications in the Process of EESP Negotiation

IKEv2 Notify Message Status Type USE_WESP_MODE, [RFC5840], is not supported when negotiating EESP SA, because the WESP functionality is part of EESP protocol. If this notification is received it MUST be ignored.

The ESP_TFC_PADDING_NOT_SUPPORTED, [RFC7296], notification is not supported in EESP, instead use IP-TFS, USE_AGGFRAG, [RFC9347]. If this notification is received it MUST be ignored.

3.5. Announcing Maximum Sub SA ID

In the process of establishing the EESP SA, each peer MAY inform the other side about the maximum value of Sub SA ID that it can accept as a receiver. The other side MUST choose IDs for its outgoing Sub SAs in the range from zero to this value (inclusive). Thus, announcing the maximum value for Sub SA ID effectively limits the number of Sub SAs the sending side is ready to handle as a Sub SA receiver.

Note that this is not a negotiation: each side can indicate its own value for the maximum Sub SA ID. In addition, sending side is not required to consume all possible Sub SA IDs up to the indicated maximum value - it can create fewer Sub SAs. In any case, when creating Sub SAs as a sender an endpoint has to consider that Sub SA IDs MUST NOT repeat for a given EESP SA and MUST NOT exceed the value sent by the peer in this notification. The actual number of Sub SAs can be different in different directions.

A new notify status type EESP_MAX_SUB_SA_ID (<TBD3>) is defined by this document. The format of the Notify payload for this notification is shown below.

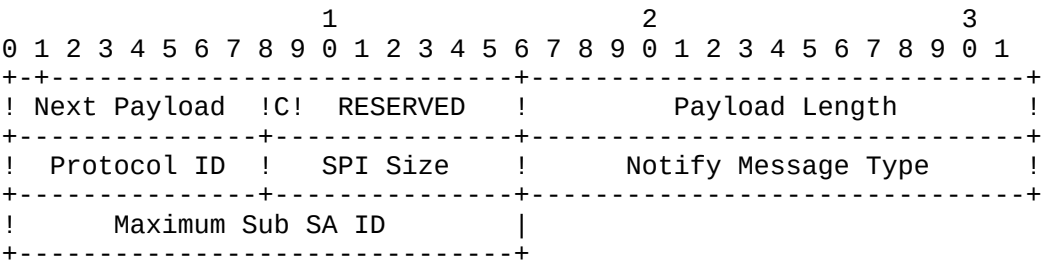

Figure 2: Maximum Sub SA Notification

* Protocol ID (1 octet) - MUST be 0. MUST be ignored if not 0.

* SPI Size (1 octet) - MUST be 0. MUST be ignored if not 0.

* Notify Status Message Type (2 octets) - set to EESP_MAX_SUB_SA_ID (<TBD3>).

* Maximum Sub SA ID (2 octets, integer in network byte order) -- specifies the maximum value for the EESP Sub SA ID the sender of this notification is expecting to receive

The maximum number of Sub SAs the sender of this notification can handle as a receiver can be calculated as the value of the Maximum Sub SA ID field plus 1. For example, value 0 in the Maximum Sub SA ID field means that only one Sub SA (with Subs SA ID = 0) can be handled.

If a peer doesn't have any restrictions on the number of the incoming Sub SAs, then it MAY omit sending this notification. As a consequence, if this notification was not received by a peer, that peer can assume that it can create as many outgoing Sub SAs as it needs (provided that Sub SA IDs not repeat).

If no SSKDF transform was negotiated, this notification MUST be ignored by peers.

3.6. Announcing Maximum Crypt Offset

Each peer MAY inform the other side about the maximum offset they accept in the EESP 'Crypt Offset' option. The other side MUST NOT use a Crypt Offset exceeding this value (inclusive).

Note that this is not a negotiation: each side can indicate its own value for the maximum Crypt Offset. If a valid EESP packet is received where the Crypt Offset exceeds the announced maximum, it MUST be dropped, and the Child SA SHOULD be deleted.

A new notify status type EESP_MAX_CRYPT_OFFSET (<TBD4>) is defined by this document. The format of the Notify payload for this notification is shown below.

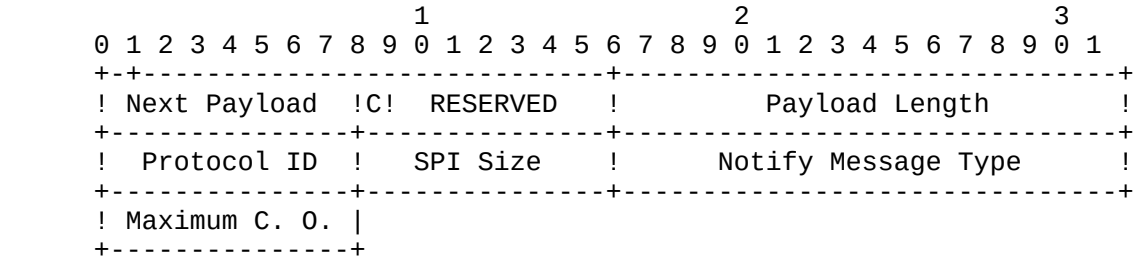

Figure 3: Maximum Crypt Offset Notification

* Protocol ID (1 octet) - MUST be 0. MUST be ignored if not 0.

* SPI Size (1 octet) - MUST be 0. MUST be ignored if not 0.

* Notify Status Message Type (2 octets) - set to EESP_MAX_CRYPT_OFFSET (<TBD4>).

* Maximum Crypt Offset (1 octet) -- specifies the maximum value for the CryptOffset field in the EESP Crypt Offset option the sender of this notification is accepting (measured in 4-octet units). Note that the field in the option is only 6 bits wide.

If a peer doesn't allow the use of the Crypt Offset option, instead of sending the value 0, the notification SHOULD be omitted entirely. That is, if this notification was not received by a peer, that peer MUST not use a Crypt Offset when sending EESP packets. If a packet with Crypt Offset option is still received, it MUST be dropped, and the Child SA SHOULD be deleted.

4. Key Derivation for Sub SAs

When an EESP SA is using Sub SAs, each Sub SA (including the one with Session ID 0) uses separate keys. This allows each Sub SA to use its own independent Sequence Number and IV space.

In order to derive these keys, a Sub SA Key Derivation Function (SSKDF) MUST be negotiated as part of the proposal of the EESP SA using Transform Type <TBD2>. This SSKDF is independent of the PRF negotiated for IKEv2.

If no Sub SAs are to be used for an EESP SA, Transform Type <TBD2> SHOULD be omitted in the proposal, but it MAY be NONE. If it's omitted or NONE is selected by the responder, Sub SAs cannot be created by either peer and the key derivation for the in- and outbound EESP SAs of the Child SA are done as described in section 2.17 of [RFC7296].

If an SSKDF is selected as part of the proposal, instead of directly taking keys for the Sub SAs from KEYMAT, as described in section 2.17 of [RFC7296], only one 'root' key is taken for each EESP SA of the Child SA. Their length is determined by the key size of the negotiated SSKDF. The root key for the EESP SA carrying data from the initiator to the responder is taken before that for the SA going from the responder to the initiator.

The root key and SSKDF are configured as properties of an EESP SA, which derives the keys for individual Sub SAs as specified in [I-D.klassert-ipsecme-eesp].

Because individual Sub SAs can't be rekeyed, the complete EESP Child SA MUST be rekeyed when either a cryptographic limit or a time-based limit is reached for any individual Sub SA.

5. IANA Considerations

5.1. Changes in the Existing IKEv2 Registries

5.1.1. IKEv2 Security Protocol Identifiers registry

This document defines new Protocol ID in the "IKEv2 Security Protocol Identifiers" registry:

Protocol ID	Protocol	Reference
<TBD1>	EESPv0	[this document]

Table 3

5.1.2. IKEv2 Transform Type Values

This document defines a new transform type in the "Transform Type Values" registry:

Type	Description	Used In	Reference
<TBD2>	Sub SA Key Derivation Function (SSKDF)	(EESPv0)	[this document]

Table 4

Valid Transform IDs are defined in a new registry listed in Table 7.

This document also modifies the "Used In" column of existing "Encryption Algorithm (ENCR)" transform type by adding EESPv0 as allowed protocol for this transform and adding a rederence to this document.

5.1.3. IKEv2 Notify Message Status Types registry.

Value	Notify Message Status Type	Reference
<TBD3>	EESP_MAX_SUB_SA_ID	[this document]
<TBD4>	EESP_MAX_CRYPT_OFFSET	[this document]

Table 5

5.1.4. Sequence Number

This document defines two new values in the IKEv2 "Transform Type 5

* Sequence Numbers Properties Transform IDs" registry:

Value	Name	Reference
<TBD5>	64-bit Sequential Numbers	[this document]
<TBD6>	None	[this document]

Table 6

5.2. New IKEv2 Registries

A new set of registries is created for EESP on IKEv2 parameters page [IKEv2-IANA]. The terms Reserved, Expert Review and Private Use are to be applied as defined in [RFC8126].

5.2.1. Transform Type <TBD2> - Sub SA Key Derivation Function Transform IDs

This documents creates the new IKEv2 registry "Transform Type <TBD2> - Sub SA Key Derivation Function Transform IDs". The initial values of this registry are:

Number	Name	Reference
0	NONE	[this document]
1	SSKDF_HKDF_SHA2_256	[this document]
2	SSKDF_HKDF_SHA2_384	[this document]
3	SSKDF_HKDF_SHA2_512	[this document]
4	SSKDF_AES256_CMAC	[TBD]
5-1023	Unassigned	[this document]
1024-65535	Private use	[this document]

Table 7: "Transform Type <TBD2>" Registry

Changes and additions to the unassigned range of this registry are by the Expert Review Policy [RFC8126].

5.2.2. Guidance for Designated Experts

In all cases of Expert Review Policy described here, the Designated Expert (DE) is expected to ascertain the existence of suitable documentation (a specification) as described in [RFC8126] and to verify that the document is permanently and publicly available. The DE is also expected to check the clarity of purpose and use of the requested code points. Last, the DE must verify that any specification produced outside the IETF does not conflict with work that is active or already published within the IETF.

6. Implementation Status

[Note to RFC Editor: Please remove this section and the reference to [RFC7942] before publication.]

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

Authors are requested to add a note to the RFC Editor at the top of this section, advising the Editor to remove the entire section before publication, as well as the reference to [RFC7942].

7. Security Considerations

EESP option Crypt Offset [I-D.klassert-ipsecme-eesp] section [XXX] allows exposing transport headers for telemetry. It is indented use of within data center.

When an EESP receiver implementation uses Stateless Decryption, it may not rely on single Security Policy Database (SPD) as specified in the IPsec Architecture document [RFC4301], section 4.4.1. However, the receiver MUST validate the negotiated Security Policy through other means to ensure compliance with the intended security requirements. For by adding Security Policy to the socket or route entry. Also comply with ICMP processing specified in section 6 of [RFC4301].

If the replay protection service is disabled, an attacker can re-play packets with a different source address. Such an attacker could disrupt the connection by replaying a single packet with a different source address or port number. In this case the receiver SHOULD NOT dynamically modify ports or addresses without using IKEv2 Mobility [RFC4555].

Additional security relevant aspects of using the IPsec protocol are discussed in the Security Architecture document [RFC4301].

8. Acknowledgments

TBD

9. Normative References

[I-D.klassert-ipsecme-eesp]
Klassert, S., Antony, A., and C. Hopps, "Enhanced Encapsulating Security Payload (EESP)", Work in Progress, Internet-Draft, draft-klassert-ipsecme-eesp-02, 26 February 2025, <https://datatracker.ietf.org/doc/html/draft-klassert-ipsecme-eesp-02>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

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

[RFC5840] Grewal, K., Montenegro, G., and M. Bhatia, "Wrapped Encapsulating Security Payload (ESP) for Traffic Visibility", RFC 5840, DOI 10.17487/RFC5840, April 2010, <https://www.rfc-editor.org/info/rfc5840>.

[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>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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>.

10. Informative References

[IKEv2-IANA] IANA, "IKEv2 Parameters", <https://www.iana.org/assignments/ikev2-parameters/ikev2-parameters.xhtml>.

[PSP] Google, "PSP Architecture Specification", <https://github.com/google/psp/blob/main/doc/PSP_Arch_Spec.pdf>.

[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>.

[RFC2992] Hopps, C., "Analysis of an Equal-Cost Multi-Path Algorithm", RFC 2992, DOI 10.17487/RFC2992, November 2000, <https://www.rfc-editor.org/info/rfc2992>.

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

[RFC5869] Krawczyk, H. and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", RFC 5869, DOI 10.17487/RFC5869, May 2010, <https://www.rfc-editor.org/info/rfc5869>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/info/rfc7942>.

[RFC8750] Migault, D., Guggemos, T., and Y. Nir, "Implicit Initialization Vector (IV) for Counter-Based Ciphers in Encapsulating Security Payload (ESP)", RFC 8750, DOI 10.17487/RFC8750, March 2020, <https://www.rfc-editor.org/info/rfc8750>.

[RFC9347] Hopps, C., "Aggregation and Fragmentation Mode for Encapsulating Security Payload (ESP) and Its Use for IP Traffic Flow Security (IP-TFS)", RFC 9347, DOI 10.17487/RFC9347, January 2023, <https://www.rfc-editor.org/info/rfc9347>.

[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>.

Appendix A. Additional Stuff

TBD

Authors' Addresses

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

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

Tobias Brunner
codelabs GmbH
Email: tobias@codelabs.ch

Valery Smyslov
ELVIS-PLUS
Email: svan@elvis.ru