Minimal ESP
draft-ietf-lwig-minimal-esp-01

Abstract

   This document describes a minimal implementation of the IP
   Encapsulation Security Payload (ESP) defined in RFC 4303. Its
   purpose is to enable implementation of ESP with a minimal set of
   options to remain compatible with ESP as described in RFC 4303. A
   minimal version of ESP is not intended to become a replacement of the
   RFC 4303 ESP, but instead to enable a limited implementation to
   interoperate with implementations of RFC 4303 ESP.

   This document describes what is required from RFC 4303 ESP as well as
   various ways to optimize compliance with RFC 4303 ESP.

   This document does not update or modify RFC 4303, but provides a
   compact description of how to implement the minimal version of the
   protocol. If this document and RFC 4303 conflicts then RFC 4303 is
   the authoritative description.

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 May 1, 2021.

Copyright Notice

Table of Contents

1. Requirements Notation

   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.

2. Introduction

   ESP [RFC4303] is part of the IPsec protocol suite [RFC4301]. IPsec
   is used to provide confidentiality, data origin authentication,
   connectionless integrity, an anti-replay service (a form of partial
   sequence integrity) and limited traffic flow confidentiality.

Figure 1 describes an ESP Packet. Currently ESP is implemented in
the kernel of major multi purpose Operating Systems (OS). The ESP
and IPsec suite is usually implemented in a complete way to fit
multiple purpose usage of these OS. However, completeness of the
IPsec suite as well as multi purpose scope of these OS is often
performed at the expense of resources, or a lack of performance. As
a result, constraint devices are likely to have their own
implementation of ESP optimized and adapted to their specificities.
With the adoption of IPsec by IoT devices with minimal IKEv2
[RFC7815] and ESP Header Compression (EHC) with
[I-D.mglt-ipsecme-diet-esp] or
[I-D.mglt-ipsecme-ikev2-diet-esp-extension], it becomes crucial that
ESP implementation designed for constraint devices remain inter-
operable with the standard ESP implementation to avoid a fragmented
usage of ESP. This document describes the the minimal properties and
ESP implementation needs to meet.

For each field of the ESP packet represented in Figure 1 this
document provides recommendations and guidance for minimal
implementations. The primary purpose of Minimal ESP is to remain
interoperable with other nodes implementing RFC 4303 ESP, while
limiting the standard complexity of the implementation.

```
0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ----
|               Security Parameters Index (SPI)                 | ^Int.
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |Cov-
|                      Sequence Number                          | |ered
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | ----
|                    Payload Data* (variable)                   | |   ^
~                                                               ~ |   |
|                                                               | |Conf.
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |Cov-
|               |         Padding (0-255 bytes)                 | |ered*
+-+-+-+-+-+-+-+-+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |   |
|                               | Pad Length   | Next Header    | v   v
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ------
|          Integrity Check Value-ICV   (variable)              |
~                                                               ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
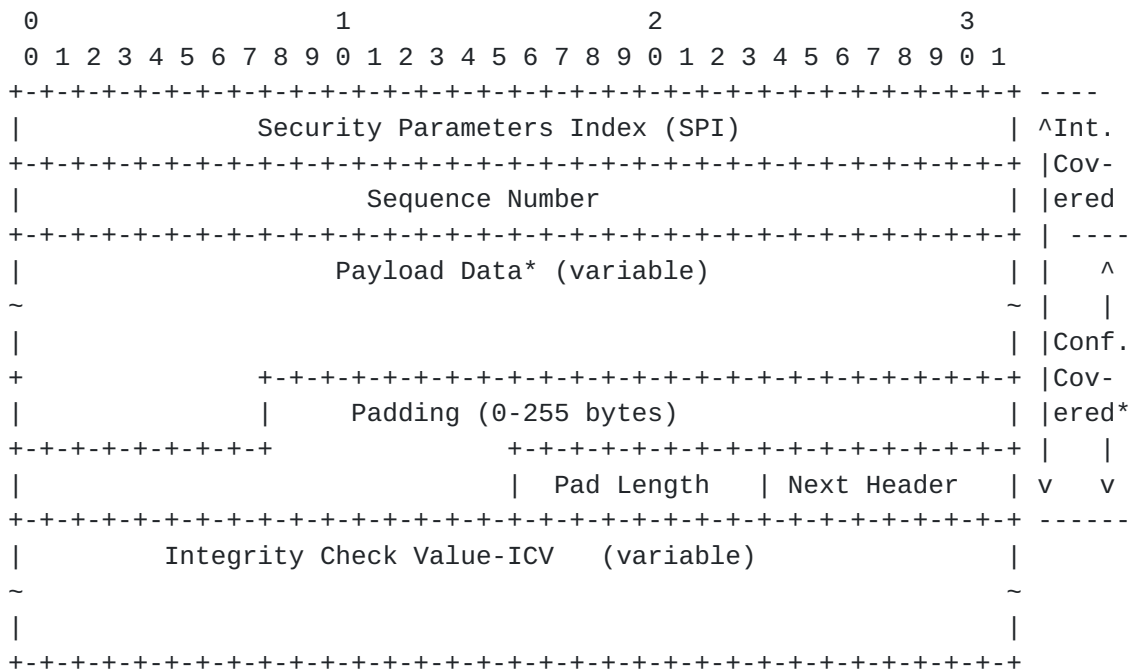
Figure 1: ESP Packet Description

3. Security Parameter Index (SPI) (32 bit)

   According to the [RFC4303], the SPI is a mandatory 32 bits field and
   is not allowed to be removed.

   The SPI has a local significance to index the Security Association
   (SA). From [RFC4301] section 4.1, nodes supporting only unicast
   communications can index their SA only using the SPI. On the other
   hand, nodes supporting multicast communications must also use the IP
   addresses and thus SA lookup needs to be performed using the longest
   match.

   For nodes supporting only unicast communications, it is RECOMMENDED
   to index SA with the SPI only. The index MAY be based on the full 32
   bits of SPI or a subset of these bits. Some other local constraints
   on the node may require a combination of the SPI as well as other
   parameters to index the SA.

   Values 0-255 MUST NOT be used. As per section 2.1 of [RFC4303],
   values 1-255 are reserved and 0 is only allowed to be used internal
   and it MUST NOT be send on the wire.

   It is RECOMMENDED to index each inbound session with a SPI randomly
   generate over 32 bits. Upon the generation of a SPI the peer checks
   the SPI is not already used and does not fall in the 0-255 range. If
   the SPI has an acceptable value, it is used to index the inbound
   session, otherwise the SPI is re-generated until an acceptable value
   is found. A random generation provides a stateless way to generate
   the SPIs, while keeping the probability of collision between SPIs
   relatively low.

   However, for some constrained nodes, generating and handling 32 bit
   random SPI may consume too much resource, in which case SPI can be
   generated using predictable functions or end up in a using a subset
   of the possible values for SPI. In fact, the SPI does not
   necessarily need to be randomly generated. A node provisioned with
   keys by a third party - e.g. that does not generate them - and that
   uses a transform that does not needs random data may not have such
   random generators. However, non random SPI and restricting their
   possible values MAY lead to privacy and security concerns. As a
   result, this alternative should be considered for devices that would
   be strongly impacted by the generation of a random SPI and after
   understanding the privacy and security impact of generating non
   random SPI.

   When a constrained node limits the number of possible SPIs this limit
   should both consider the number of inbound SAs - possibly per IP
   addresses - as well as the ability for the node to rekey. SPI can

typically be used to proceed to clean key update and the SPI value
may be used to indicate which key is being used. This can typically
be implemented by a SPI being encoded with the SAD entry on a subset
of bytes (for example 3 bytes), while the remaining byte is left to
indicate the rekey index.

Note that SPI value is used only for inbound traffic, as such the SPI
negotiated with IKEv2 [RFC7296] or [RFC7815] by a peer, is the value
used by the remote peer when its sends traffic. As SPI are only used
for inbound traffic by the peer, this allows each peer to manage the
set of SPIs used for its inbound traffic.

The use of a limited number of SPIs or non random SPIs come with
security or privacy drawbacks. Typically, a passive attacker may
derive information such as the number of constraint devices
connecting the remote peer, and in conjunction with data rate, the
attacker may eventually determine the application the constraint
device is associated to. If the SPIs are set by a manufacturer or by
some software application, the SPI may leak in an obvious way the
type of sensor, the application involved or the model of the
constraint device. When identification of the application or the
hardware is associated to privacy, the SPI MUST be randomly
generated. However, one needs to realize that in this case this is
likely to be sufficient and a thorough privacy analysis is required.
More specifically, traffic pattern may leak sufficient information in
itself. In other words, privacy leakage is a complex and the use of
random SPI is unlikely to be sufficient.

As the general recommendation is to randomly generate the SPI,
constraint devices that will use a (very) limited number of SPIs are
expected to be very constraint devices with very limited
capabilities, where the use of randomly generated SPI may prevent
them to implement IPsec. In this case the ability to provision non
random SPI enables these devices to secure their communications.
These devices, due to there limitations, are expected to provide
limited information and how the use of non random SPI impacts privacy
requires further analysis. Typically temperature sensors, wind
sensors, used outdoor do not leak privacy sensitive information.
When used indoor, privacy leakage outside the local network may be
limited.

As far as security is concerned, revealing the type of application or
model of the constraint device could be used to identify the
vulnerabilities the constraint device is subject to. This is
especially sensitive for constraint devices where patches or software
updates will be challenging to operate. As a result, these devices
may remain vulnerable for relatively long period. In addition,
predictable SPIs enable an attacker to forge packets with a valid

SPI. Such packet will not be rejected due to an SPI mismatch, but
instead after the signature check which requires more resource and
thus make DoS more efficient, especially for devices powered by
batteries.

4. **Sequence Number(SN) (32 bit)**

According to [RFC4303], the Sequence Number (SN) is a mandatory 32
bits field in the packet.

The SN is set by the sender so the receiver can implement anti-replay
protection. The SN is derived from any strictly increasing function
that guarantees: if packet B is sent after packet A, then SN of
packet B is strictly greater then the SN of packet A.

Some constraint devices may establish communication with specific
devices, like a specific gateway, or nodes similar to them. As a
result, the sender may know whereas the receiver implements anti-
replay protection or not. Even though the sender may know the
receiver does not implement anti replay protection, the sender MUST
implement a always increasing function to generate the SN.

Usually, SN is generated by incrementing a counter for each packet
sent. A constraint device may avoid maintaining this context and use
another source that is known to always increase. Typically,
constraint nodes using 802.15.4 Time Slotted Channel Hopping (TSCH),
whose communication is heavily dependent on time, can take advantage
of their clock to generate the SN. This would guarantee a strictly
increasing function, and avoid storing any additional values or
context related to the SN. When the use of a clock is considered,
one should take care that packets associated to a given SA are not
sent with the same time value. Note however that standard receivers
are generally configured with incrementing counters and, if not
appropriately configured, the use of a significantly larger SN may
result in the packet out of the receiver's windows and that packet
being discarded.

For inbound traffic, it is RECOMMENDED that any receiver provide a
anti-replay protection, and the size of the window depends on the
ability of the network to deliver packet out of order. As a result,
in environment where out of order packets is not possible the window
size can be set to one. However, while RECOMMENDED, there is no
requirements to implement an anti replay protection mechanism
implemented by IPsec. A node MAY drop anti-replay protection
provided by IPsec, and instead implement its own internal mechanism.

SN can be encoded over 32 bits or 64 bits - known as Extended
Sequence Number (ESN). As per [RFC4303], the support ESN is not

mandatory. The determination of the use of ESN is based on the
largest possible value a SN can take over a session. When SN is
incremented for each packet, the number of packets sent over the life
time of a session may be considered. However, when the SN is
incremented differently - such as when time is used - the maximum
value SN needs to be considered instead. Note that the limit of
messages being sent is primary determined by the security associated
to the key rather than the SN. The security of the key used to
encrypt decreases with each message being sent and a node MUST ensure
the limit is not reached - even though the SN would permit it. In a
constrainted environment, it is likely that the implementation of a
rekey mechanism is preferred over the use of ESN.

[5](#). Padding

The purpose of padding is to respect the 32 bit alignment of ESP or
block size expected by an encryption transform - such as AES-CBC for
example. ESP MUST have at least one padding byte Pad Length that
indicates the padding length. ESP padding bytes are generated by a
succession of unsigned bytes starting with 1, 2, 3 with the last byte
set to Pad Length, where Pad Length designates the length of the
padding bytes.

Checking the padding structure is not mandatory, so the constraint
device may not proceed to such checks, however, in order to
interoperate with existing ESP implementations, it MUST build the
padding bytes as recommended by ESP.

In some situation the padding bytes may take a fix value. This would
typically be the case when the Data Payload is of fix size.

ESP [RFC4303] also provides Traffic Flow Confidentiality (TFC) as a
way to perform padding to hide traffic characteristics, which differs
from respecting a 32 bit alignment. TFC is not mandatory and MUST be
negotiated with the SA management protocol. TFC has not yet being
widely adopted for standard ESP traffic. One possible reason is that
it requires to shape the traffic according to one traffic pattern
that needs to be maintained. This is likely to require extra
processing as well as providing a "well recognized" traffic shape
which could end up being counterproductive. As such TFC is not
expected to be supported by a minimal ESP implementation.

As a result, TFC cannot be enabled with minimal ESP, and
communication protection that were rely on TFC will be more sensitive
to traffic shaping. This could expose the application as well as the
devices used to a passive monitoring attacker. Such information
could be used by the attacker in case a vulnerability is disclosed on
the specific device. In addition, some application use - such as

health applications - may also reveal important privacy oriented
informations.

Some constrained nodes that have limited battery life time may also
prefer avoiding sending extra padding bytes. However the same nodes
may also be very specific to an application and device. As a result,
they are also likely to be the main target for traffic shaping. In
most cases, the payload carried by these nodes is quite small, and
the standard padding mechanism may also be used as an alternative to
TFC, with a sufficient trade off between the require energy to send
additional payload and the exposure to traffic shaping attacks. In
addition, the information leaked by the traffic shaping may also be
addressed by the application level. For example, it is preferred to
have a sensor sending some information at regular time interval,
rather when an specific event is happening. Typically a sensor
monitoring the temperature, or a door is expected to send regularly
the information - i.e. the temperature of the room or whether the
door is closed or open) instead of only sending the information when
the temperature has raised or when the door is being opened.

6. Next Header (8 bit)

According to [RFC4303], the Next Header is a mandatory 8 bits field
in the packet. Next header is intended to specify the data contained
in the payload as well as dummy packet. In addition, the Next Header
may also carry an indication on how to process the packet
[I-D.nikander-esp-beet-mode].

The ability to generate and receive dummy packet is required by
[RFC4303]. For interoperability, it is RECOMMENDED a minimal ESP
implementation discards dummy packets. Note that such recommendation
only applies for nodes receiving packets, and that nodes designed to
only send data may not implement this capability.

As the generation of dummy packets is subject to local management and
based on a per-SA basis, a minimal ESP implementation may not
generate such dummy packet. More especially, in constraint
environment sending dummy packets may have too much impact on the
device life time, and so may be avoided. On the other hand,
constrained nodes may be dedicated to specific applications, in which
case, traffic pattern may expose the application or the type of node.
For these nodes, not sending dummy packet may have some privacy
implication that needs to be measured. However, for the same reasons
exposed in Section 5 traffic shaping at the IPsec layer may also
introduce some traffic pattern, and on constraint devices the
application is probably the most appropriated layer to limit the risk
of leaking information by traffic shaping.

In some cases, devices are dedicated to a single application or a single transport protocol, in which case, the Next Header has a fix value.

Specific processing indications have not been standardized yet [I-D.nikander-esp-beet-mode] and is expected to result from an agreement between the peers. As a result, it is not expected to be part of a minimal implementation of ESP.

7. ICV

The ICV depends on the crypto-suite used. Currently [RFC8221] only recommends crypto-suites with an ICV which makes the ICV a mandatory field.

As detailed in Section 8 we recommend to use authentication, the ICV field is expected to be present that is to say with a size different from zero. This makes it a mandatory field which size is defined by the security recommendations only.

8. Cryptographic Suites

The cryptographic suites implemented are an important component of ESP. The recommended suites to use are expected to evolve over time and implementers SHOULD follow the recommendations provided by [RFC8221] and updates. Recommendations are provided for standard nodes as well as constrained nodes.

This section lists some of the criteria that may be considered. The list is not expected to be exhaustive and may also evolve overtime. As a result, the list is provided as indicative:

1. Security: Security is the criteria that should be considered first for the selection of encryption algorithm transform. The security of encryption algorithm transforms is expected to evolve over time, and it is of primary importance to follow up-to-date security guidances and recommendations. The chosen encryption algorithm transforms MUST NOT be known vulnerable or weak (see [RFC8221] for outdated ciphers). ESP can be used to authenticate only or to encrypt the communication. In the later case, authenticated encryption must always be considered [RFC8221].

2. Resilience to nonce re-use: Some transforms -including AES-GCM - are very sensitive to nonce collision with a given key. While the generation of the nonce may prevent such collision during a session, the mechanisms are unlikely to provide such protection across reboot. This causes an issue for devices that are configured with a key. When the key is likely to be re-used

across reboots, it is RECOMMENDED to consider transforms that are
nonce misuse resistant such as AES-GCM-SIV for example[RFC8452]

3. Interoperability: Interoperability considers the encryption
    algorithm transforms shared with the other nodes. Note that it
    is not because an encryption algorithm transform is widely
    deployed that is secured. As a result, security SHOULD NOT be
    weaken for interoperability. [RFC8221] and successors consider
    the life cycle of encryption algorithm transforms sufficiently
    long to provide interoperability. Constraint devices may have
    limited interoperability requirements which makes possible to
    reduces the number of encryption algorithm transforms to
    implement.

4. Power Consumption and Cipher Suite Complexity: Complexity of the
    encryption algorithm transform or the energy associated to it are
    especially considered when devices have limited resources or are
    using some batteries, in which case the battery determines the
    life of the device. The choice of a cryptographic function may
    consider re-using specific libraries or to take advantage of
    hardware acceleration provided by the device. For example if the
    device benefits from AES hardware modules and uses AES-CTR, it
    may prefer AUTH_AES-XCBC for its authentication. In addition,
    some devices may also embed radio modules with hardware
    acceleration for AES-CCM, in which case, this mode may be
    preferred.

5. Power Consumption and Bandwidth Consumption: Similarly to the
    encryption algorithm transform complexity, reducing the payload
    sent, may significantly reduce the energy consumption of the
    device. As a result, encryption algorithm transforms with low
    overhead may be considered. To reduce the overall payload size
    one may for example:

    1. Use of counter-based ciphers without fixed block length (e.g.
        AES-CTR, or ChaCha20-Poly1305).

    2. Use of ciphers with capability of using implicit IVs
        [RFC8750].

    3. Use of ciphers recommended for IoT [RFC8221].

    4. Avoid Padding by sending payload data which are aligned to
        the cipher block length - 2 for the ESP trailer.

9. IANA Considerations

   There are no IANA consideration for this document.

10. Security Considerations

   Security considerations are those of [RFC4303]. In addition, this
   document provided security recommendations an guidances over the
   implementation choices for each fields.

   The security of a communication provided by ESP is closely related to
   the security associated to the management of that key. This usually
   include mechanisms to prevent a nonce to repeat for example. When a
   node is provisioned with a session key that is used across reboot,
   the implementer MUST ensure that the mechanisms put in place remain
   valid across reboot as well.

   It is RECOMMENDED to use ESP in conjunction of key management
   protocols such as for example IKEv2 [RFC7296] or minimal IKEv2
   [RFC7815]. Such mechanisms are responsible to negotiate fresh
   session keys as well as prevent a session key being use beyond its
   life time. When such mechanisms cannot be implemented and the
   session key is, for example, provisioned, the nodes SHOULD ensure
   that keys are not used beyond their life time and that the
   appropriated use of the key remains across reboots.

   When a node generates its key or when random value such as nonces are
   generated, the random generation MUST follow [RFC4086].

11. Acknowledgment

   The authors would like to thank Daniel Palomares, Scott Fluhrer, Tero
   Kivinen, Valery Smyslov, Yoav Nir, Michael Richardson for their
   valuable comments. In particular Scott Fluhrer suggested to include
   the rekey index in the SPI as well as the use of transform resilient
   to nonce misuse.

12. References

12.1. Normative References

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

   [RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker,
              "Randomness Requirements for Security", BCP 106, RFC 4086,
              DOI 10.17487/RFC4086, June 2005,
              <https://www.rfc-editor.org/info/rfc4086>.

   [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>.

   [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>.

   [RFC7815] Kivinen, T., "Minimal Internet Key Exchange Version 2
              (IKEv2) Initiator Implementation", RFC 7815,
              DOI 10.17487/RFC7815, March 2016,
              <https://www.rfc-editor.org/info/rfc7815>.

   [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>.

   [RFC8221] Wouters, P., Migault, D., Mattsson, J., Nir, Y., and T.
              Kivinen, "Cryptographic Algorithm Implementation
              Requirements and Usage Guidance for Encapsulating Security
              Payload (ESP) and Authentication Header (AH)", RFC 8221,
              DOI 10.17487/RFC8221, October 2017,
              <https://www.rfc-editor.org/info/rfc8221>.

   [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>.

12.2. Informative References

   [I-D.mglt-ipsecme-diet-esp]
              Migault, D., Guggemos, T., Bormann, C., and D. Schinazi,
              "ESP Header Compression and Diet-ESP", draft-mglt-ipsecme-
              diet-esp-07 (work in progress), March 2019.

   [I-D.mglt-ipsecme-ikev2-diet-esp-extension]
              Migault, D., Guggemos, T., and D. Schinazi, "Internet Key
              Exchange version 2 (IKEv2) extension for the ESP Header
              Compression (EHC) Strategy", draft-mglt-ipsecme-ikev2-
              diet-esp-extension-01 (work in progress), June 2018.

   [I-D.nikander-esp-beet-mode]
              Nikander, P. and J. Melen, "A Bound End-to-End Tunnel
              (BEET) mode for ESP", draft-nikander-esp-beet-mode-09
              (work in progress), August 2008.

   [RFC8452] Gueron, S., Langley, A., and Y. Lindell, "AES-GCM-SIV:
              Nonce Misuse-Resistant Authenticated Encryption",
              RFC 8452, DOI 10.17487/RFC8452, April 2019,
              <https://www.rfc-editor.org/info/rfc8452>.

Appendix A. Document Change Log

   [RFC Editor: This section is to be removed before publication]

   -00: First version published.

   -01: Clarified description

   -02: Clarified description

Authors' Addresses

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

   EMail: daniel.migault@ericsson.com

   Tobias Guggemos
   LMU Munich
   MNM-Team
   Oettingenstr. 67
   80538 Munich, Bavaria
   Germany

   EMail: guggemos@mnm-team.org