IPSECME   D. Migault, Ed.
Internet-Draft   Ericsson
Intended status: Standards Track   T. Guggemos, Ed.
Expires: December 22, 2017   LMU Munich
                                                              Y. Nir
                                                         Check Point
                                                       June 20, 2017

               Implicit IV for Counter-based Ciphers in IPsec
                     draft-mglt-ipsecme-implicit-iv-03

Abstract

   IPsec ESP sends an initialization vector (IV) or nonce in each
   packet, adding 8 or 16 octets. Some algorithms such as AES-GCM, AES-
   CCM, AES-CTR and ChaCha20-Poly1305 require a unique nonce but do not
   require an unpredictable nonce. When using such algorithms the
   packet counter value can be used to generate a nonce, saving 8 octets
   per packet. This document describes how to do this.

Status of This Memo

Copyright Notice

Table of Contents

1. Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Introduction

Counter-based AES modes of operation such as AES-CTR ([RFC3686]), AES-CCM ([RFC4309]), and AES-GCM ([RFC4106]) require the specification of an nonce for each ESP packet. The same applies for ChaCha20-Poly1305 ([RFC7634]. Currently this nonce is sent in each ESP packet ([RFC4303]). This practice is designated in this document as "explicit nonce".

In some context, such as IoT, it may be preferable to avoid carrying the extra bytes associated to the IV and instead generate it locally on each peer. The local generation of the nonce is designated in this document as "implicit IV".

The size of this nonce depends on the specific algorithm, but all of the algorithms mentioned above take an 8-octet nonce.

This document defines how to compute the nonce locally when it is implicit. It also specifies how peers agree with the Internet Key Exchange version 2 (IKEv2 - [RFC7296]) on using an implicit IV versus an explicit IV.

This document limits its scope to the algorithms mentioned above.
Other algorithms with similar properties may later be defined to use
this extension.

This document does not consider AES-CBC ([RFC3602]) as AES-CBC
requires the IV to be unpredictable. Deriving it directly from the
packet counter as described below is insecure as mentioned in
Security Consideration of [RFC3602] and has led to real world chosen
plain-text attack such as BEAST [BEAST].

3. Terminology

o IoT: Internet of Things.

o IV: Initialization Vector.

o Nonce: a fixed-size octet string used only once. This is similar
   to IV, except that in common usage there is no implication of non-
   predictability.

4. Implicit IV

With the algorithms listed in Section 2, the 8 byte nonce MUST NOT
repeat. The binding between a ESP packet and its nonce is provided
using the Sequence Number or the Extended Sequence Number. Figure 1
and Figure 2 represent the IV with a regular 4-byte Sequence Number
and with an 8-byte Extended Sequence Number respectively.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Zero                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

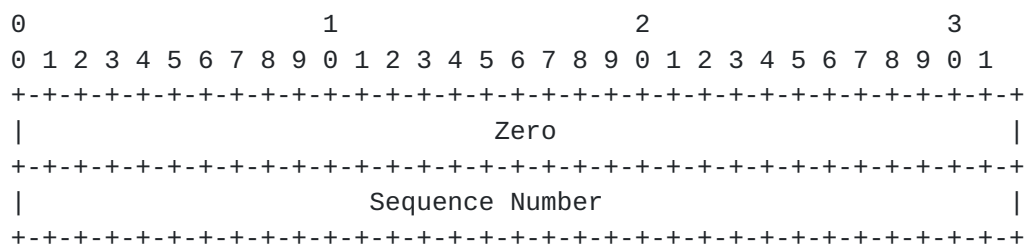   Figure 1: Implicit IV with a 4 byte Sequence Number

o Sequence Number: the 4 byte Sequence Number carried in the ESP
   packet.

o Zero: a 4 byte array with all bits set to zero.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Extended                             |
|                      Sequence Number                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
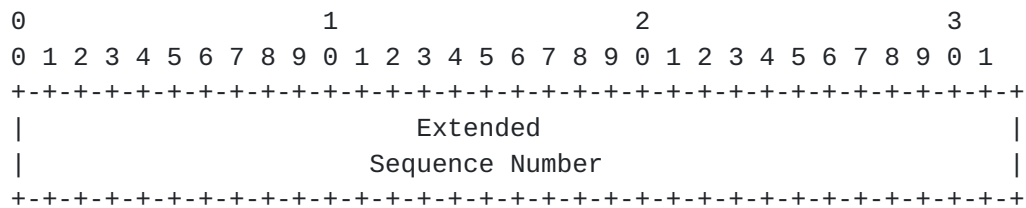
       Figure 2: Implicit IV with an 8 byte Extended Sequence Number

   o Extended Sequence Number: the 8 byte Extended Sequence Number of
      the Security Association. The 4 byte low order bytes are carried
      in the ESP packet.

5. Initiator Behavior

   An initiator supporting this feature SHOULD propose implicit IV for
   all relevant algorithms. To facilitate backward compatibility with
   non-supporting peers the initiator SHOULD also include those same
   algorithms without IIV. This may require extra transforms.

6. Responder Behavior

   The rules of SA payload processing ensure that the responder will
   never send an SA payload containing the IIV indicator to an initiator
   that does not support IIV.

7. Security Consideration

   Nonce generation for these algorithms has not been explicitly
   defined. It has been left to the implementation as long as certain
   security requirements are met. This document provides an explicit
   and normative way to generate IVs. The mechanism described in this
   document meets the IV security requirements of all relevant
   algorithms.

   As the IV MUST NOT repeat for one SPI when Counter-Mode ciphers are
   used, Implicit IV as described in this document MUST NOT be used in
   setups with the chance that the Sequence Number overlaps for one SPI.
   Multicast as described in [RFC5374], [RFC6407] and
   [I-D.yeung-g-ikev2] is a prominent example, where many senders share
   one secret and thus one SPI. Section 3.5 of [RFC6407] explains how
   repetition MAY BE prevented by using a prefix for each group member,
   which could be prefixed to the Sequence Number. Otherwise, Implicit
   IV MUST NOT be used in multicast scenarios.

8. IANA Considerations

   AES-CTR, AES-CCM, AES-GCM and ChaCha20-Poly1305 are likely to
   implement the implicit IV described in this document. This section
   limits assignment of new code points to the recommended suites
   provided in [I-D.ietf-ipsecme-rfc4307bis] and
   [I-D.ietf-ipsecme-rfc7321bis], thus the new Transform Type 1 -
   Encryption Algorithm Transform IDs are as defined below:

   - ENCR_AES-CCM_8_IIV

   - ENCR_AES-GCM_16_IIV

   - ENCR_CHACHA20-POLY1305_IIV

9. References

9.1. Normative References

   [I-D.ietf-ipsecme-rfc4307bis]
              Nir, Y., Kivinen, T., Wouters, P., and D. Migault,
              "Algorithm Implementation Requirements and Usage Guidance
              for IKEv2", draft-ietf-ipsecme-rfc4307bis-18 (work in
              progress), March 2017.

   [I-D.ietf-ipsecme-rfc7321bis]
              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)", draft-ietf-
              ipsecme-rfc7321bis-06 (work in progress), June 2017.

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

   [RFC3602] Frankel, S., Glenn, R., and S. Kelly, "The AES-CBC Cipher
              Algorithm and Its Use with IPsec", RFC 3602,
              DOI 10.17487/RFC3602, September 2003,
              <http://www.rfc-editor.org/info/rfc3602>.

   [RFC3686] Housley, R., "Using Advanced Encryption Standard (AES)
              Counter Mode With IPsec Encapsulating Security Payload
              (ESP)", RFC 3686, DOI 10.17487/RFC3686, January 2004,
              <http://www.rfc-editor.org/info/rfc3686>.

   [RFC4106] Viega, J. and D. McGrew, "The Use of Galois/Counter Mode
              (GCM) in IPsec Encapsulating Security Payload (ESP)",
              RFC 4106, DOI 10.17487/RFC4106, June 2005,
              <http://www.rfc-editor.org/info/rfc4106>.

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

   [RFC4309] Housley, R., "Using Advanced Encryption Standard (AES) CCM
              Mode with IPsec Encapsulating Security Payload (ESP)",
              RFC 4309, DOI 10.17487/RFC4309, December 2005,
              <http://www.rfc-editor.org/info/rfc4309>.

   [RFC5374] Weis, B., Gross, G., and D. Ignjatic, "Multicast
              Extensions to the Security Architecture for the Internet
              Protocol", RFC 5374, DOI 10.17487/RFC5374, November 2008,
              <http://www.rfc-editor.org/info/rfc5374>.

   [RFC6407] Weis, B., Rowles, S., and T. Hardjono, "The Group Domain
              of Interpretation", RFC 6407, DOI 10.17487/RFC6407,
              October 2011, <http://www.rfc-editor.org/info/rfc6407>.

   [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, <http://www.rfc-editor.org/info/rfc7296>.

   [RFC7634] Nir, Y., "ChaCha20, Poly1305, and Their Use in the
              Internet Key Exchange Protocol (IKE) and IPsec", RFC 7634,
              DOI 10.17487/RFC7634, August 2015,
              <http://www.rfc-editor.org/info/rfc7634>.

9.2. Informational References

   [BEAST]   Thai, T. and J. Juliano, "Here Come The xor Ninjas", ,
              May 2011, <https://www.researchgate.net/
              publication/266529975_Here_Come_The_Ninjas>.

   [I-D.yeung-g-ikev2]
              Weis, B., Nir, Y., and V. Smyslov, "Group Key Management
              using IKEv2", draft-yeung-g-ikev2-11 (work in progress),
              March 2017.

Authors' Addresses

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

    Email: daniel.migault@ericsson.com

    Tobias Guggemos (editor)
    LMU Munich
    Oettingenstr. 67
    80538 Munich, Bavaria
    Germany

    Email: guggemos@mnm-team.org
    URI: http://mnm-team.org/~guggemos

    Yoav Nir
    Check Point Software Technologies Ltd.
    5 Hasolelim st.
    Tel Aviv 6789735
    Israel

    Email: ynir.ietf@gmail.com