HAL
archives-ouvertes.fr

Simple Authentication Schemes for the Asynchronous Layered Coding (ALC) and NACK-Oriented Reliable Multicast (NORM) Protocols

Vincent Roca

▶ **To cite this version:**

Vincent Roca. Simple Authentication Schemes for the Asynchronous Layered Coding (ALC) and NACK-Oriented Reliable Multicast (NORM) Protocols. 2012. hal-00745908

HAL Id: hal-00745908
https://hal.inria.fr/hal-00745908

Submitted on 26 Oct 2012

Simple Authentication Schemes for the Asynchronous Layered Coding (ALC)
        and NACK-Oriented Reliable Multicast (NORM) Protocols

Abstract

   This document introduces four schemes that provide per-packet
   authentication, integrity, and anti-replay services in the context of
   the Asynchronous Layered Coding (ALC) and NACK-Oriented Reliable
   Multicast (NORM) protocols. The first scheme is based on RSA Digital
   Signatures. The second scheme relies on the Elliptic Curve Digital
   Signature Algorithm (ECDSA). The third scheme relies on a Group-
   keyed Message Authentication Code (MAC). Finally, the fourth scheme
   merges the Digital Signature and group schemes. These schemes have
   different target use cases, and they do not all provide the same
   service.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc6584.

Table of Contents

1. Introduction

   Many applications using multicast and broadcast communications
   require that each receiver be able to authenticate the source of any
   packet it receives, to check its integrity. For instance, ALC
   [RFC5775] and NORM [RFC5740] are two Content Delivery Protocols
   (CDPs) designed to reliably transfer objects (e.g., files) between a
   session's sender and several receivers.

   The NORM protocol is based on bidirectional transmissions. With
   NORM, each receiver acknowledges data received or, in the case of
   packet erasures, asks for retransmissions. On the contrary, the ALC
   protocol defines unidirectional transmissions. With ALC, reliability
   can be achieved by means of cyclic transmissions of the content
   within a carousel, or by the use of proactive Forward Error
   Correction (FEC) codes, or by the joint use of these mechanisms.
   Being purely unidirectional, ALC is massively scalable, while NORM is
   intrinsically limited in terms of the number of receivers that can be
   handled in a session. Both protocols have in common the fact that
   they operate at the application level, on top of an erasure channel
   (e.g., the Internet) where packets can be lost (erased) during the
   transmission.

   With these CDPs, an attacker might impersonate the ALC or NORM
   session sender and inject forged packets to the receivers, thereby
   corrupting the objects reconstructed by the receivers. An attacker
   might also impersonate a NORM session receiver and inject forged
   feedback packets to the NORM sender.

   In the case of group communications, several solutions exist to
   provide the receiver some guaranties on the integrity of the packets
   it receives and on the identity of the sender of these packets.
   These solutions have different features that make them more or less
   suited to a given use case:

   o Digital Signatures [RFC4359] (see Sections 3 and 4 of this
      document): This scheme is well suited to low data rate flows, when
      a packet sender authentication and packet integrity service is
      needed. However, Digital Signatures based on RSA asymmetric
      cryptography are limited by high computational costs and high
      transmission overheads. The use of ECC (Elliptic Curve
      Cryptography) [RFC6090] significantly relaxes these constraints.
      For instance, the following key lengths provide equivalent
      security: a 1024-bit RSA key versus a 160-bit ECC key, or a
      2048-bit RSA key versus a 224-bit ECC key. However, RSA puts more
      load on the signer but much less load on the verifier, whereas ECC
      puts more similar load on both; hence, with many verifiers, more
      CPU is consumed overall.

   o Group-keyed Message Authentication Codes (MACs) (see Section 5):
      This scheme is well suited to high data rate flows, when
      transmission overheads must be minimized. However, this scheme
      cannot protect against attacks coming from inside the group, where
      a group member impersonates the sender and sends forged messages
      to other receivers.

   o TESLA (Timed Efficient Stream Loss-tolerant Authentication)
      [RFC4082] [RFC5776]: This scheme is well suited to high data rate
      flows, when transmission overheads must be minimized, and when a
      packet sender authentication and packet integrity service is
      needed. The price is an increased complexity -- in particular,
      the need to loosely synchronize the receivers and the sender -- as
      well as the need to wait for the key to be disclosed before being
      able to authenticate a packet (i.e., the authentication check is
      delayed).

   The following table summarizes the pros and cons of each
   authentication/integrity scheme used at the application/transport
   level (where "-" means con, "0" means neutral, and "+" means pro):

	RSA Digital Signature	ECC Digital Signature	Group-Keyed MAC	TESLA
Sender auth and packet integrity	Yes	Yes	No (group security)	Yes
Non-delayed authentication	Yes	Yes	Yes	No
Anti-replay protection	Opt	Opt	Opt	No
Processing load	-	sender: -, recv: 0	+	+
Transmission overhead	-	0	+	+
Complexity	+	+	+	-

   Several authentication schemes MAY be used in the same ALC or NORM
   session, even on the same communication path. This is made possible
   through a dedicated identifier, the "ASID" (Authentication Scheme
   IDentifier), that is present in each HET=1 (EXT_AUTH) header
   extension and that tells a receiver how to interpret this HET=1
   header extension. This is discussed in Section 2.

All the applications built on top of ALC and NORM directly benefit
from the source authentication and packet integrity services defined
in this document. For instance, this is the case of the File
Delivery over Unidirectional Transport (FLUTE) application
[RMT-FLUTE], which is built on top of ALC.

The current specification assumes that several parameters (like
keying material) are communicated out-of-band, sometimes securely,
between the sender and the receivers. This is detailed in
Sections 3.2, 4.2, 5.2, and 6.2.

1.1. Scope of This Document

[RFC5776] explains how to use TESLA in the context of the ALC and
NORM protocols.

The current document specifies the use of the Digital Signature based
on RSA asymmetric cryptography, the Elliptic Curve Digital Signature
Algorithm (ECDSA), and Group-keyed MAC schemes. The current document
also specifies the joint use of Digital Signature and Group-keyed MAC
schemes.

Unlike the TESLA scheme, this specification considers the
authentication/integrity of the packets generated by the session's
sender as well as those generated by the receivers (NORM).

1.2. Terminology, Notations, and Definitions

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].

The following notations and definitions are used throughout this
document:

o MAC is the Message Authentication Code;

o HMAC is the Keyed-Hash Message Authentication Code;

o "sender" denotes the sender of a packet that needs the
   authentication/integrity check service. It can be an ALC or NORM
   session sender, or a NORM session receiver in the case of feedback
   traffic;

o "receiver" denotes the receiver of a packet that needs the
   authentication/integrity check service. It can be an ALC or NORM
   session receiver, or a NORM session sender in the case of feedback
   traffic;

o "ASID" is the Authentication Scheme IDentifier.

Key definitions for Digital Signatures are as follows:

o The public key is used by a receiver to check a packet's
   signature. This key MUST be communicated to all receivers before
   starting the session;

o The private key is used by a sender to generate a packet's
   signature;

o The private key and public key length are expressed in bits. For
   security considerations [RFC5751], when using RSA, RSASSA-PSS, and
   Digital Signature Algorithm (DSA) signatures, key sizes of length
   strictly inferior to 1024 bits SHOULD NOT be used. Key sizes of
   length between 1024 and 2048 bits inclusive SHOULD be used. Key
   sizes of length strictly superior to 2048 bits MAY be used.

Key definitions for Group-keyed MAC are as follows:

o The shared group key is used by the senders and the receivers.
   This key MUST be communicated to all group members,
   confidentially, before starting the session;

o The group key length is expressed in bits;

o n_m is the length of the truncated output of the MAC [RFC2104].
   Only the n_m leftmost bits (most significant bits) of the MAC
   output are kept.

2. Authentication Scheme Identification with the ASID Field

   As mentioned in Section 1, several authentication schemes MAY be used
   in the same ALC or NORM session, even on the same communication path
   (i.e., from a sender to a receiver, or vice versa). All the schemes
   mentioned in Section 1 (some of which are specified in this document)
   use the same HET=1 (EXT_AUTH) Authentication Header extension
   mechanism defined in [RFC5651]. Therefore, the same 4-bit ASID field
   has been reserved in all the specifications (see Sections 3.1, 4.1,
   5.1, and 6.1, as well as Section 5.1 of [RFC5776]). For a given ALC
   or NORM session, the ASID value contained in an incoming packet
   enables a receiver to differentiate the actual use and format of the
   contents of the HET=1 (EXT_AUTH) header extension.

   The association between the ASID value and the actual authentication
   scheme of a given ALC or NORM session is defined at session startup
   and communicated to all the session members by an out-of-band
   mechanism. This association is per ALC or NORM session, and
   different sessions MAY reuse the same ASID values for different
   authentication schemes.

   With ALC, the ASID value is scoped by the {sender IP address;
   Transport Session Identifier (TSI)} tuple [RFC5651] that fully
   identifies an ALC session. Since [RFC5651] requires that "the TSI
   MUST be unique among all sessions served by the sender during the
   period when the session is active, and for a large period of time
   preceding and following when the session is active", there is no risk
   of confusion between different sessions. This is in line with
   Section 7.2.3.

   With NORM, there is no session identifier within NORM packets.
   Therefore, depending on whether an Any Source Multicast (ASM) or
   Source Specific Multicast (SSM) group communication is used, the ASID
   value is scoped either by the {destination multicast address;
   destination port number} or {source IP address; destination multicast
   address; destination port number} tuple that fully identifies a NORM
   session [RFC5740]. Care should be taken that the above tuples remain
   unique, within a given scope and for a sufficient period of time
   preceding, during, and following when the session is active, to avoid
   confusion between different sessions. However, this is a
   recommendation for NORM sessions, rather than something specific to
   an authentication scheme. Note also that the ASID value is not
   scoped by the {source_id; instance_id} tuple, which uniquely
   identifies a host's participation in a NORM session, rather than the
   session itself (Section 7.2.2).

   In any case, because this ASID field is 4 bits long, there is a
   maximum of 16 authentication schemes per ALC or NORM session.

3. RSA Digital Signature Scheme

3.1. Authentication Header Extension Format

   The integration of Digital Signatures is similar in ALC and NORM and
   relies on the header extension mechanism defined in both protocols.
   More precisely, this document details the HET=1 (EXT_AUTH) header
   extension defined in [RFC5651].

Several fields are added, in addition to the HET (Header Extension
Type) and HEL (Header Extension Length) fields (Figure 1).

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    HET (=1)    |      HEL      | ASID | rsvd|A|               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+R+               +
~                  anti-replay Sequence Number (SN)            ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                                                               ~
|                          Signature                            |
+                                        +-+-+-+-+-+-+-+-+
|                                        |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
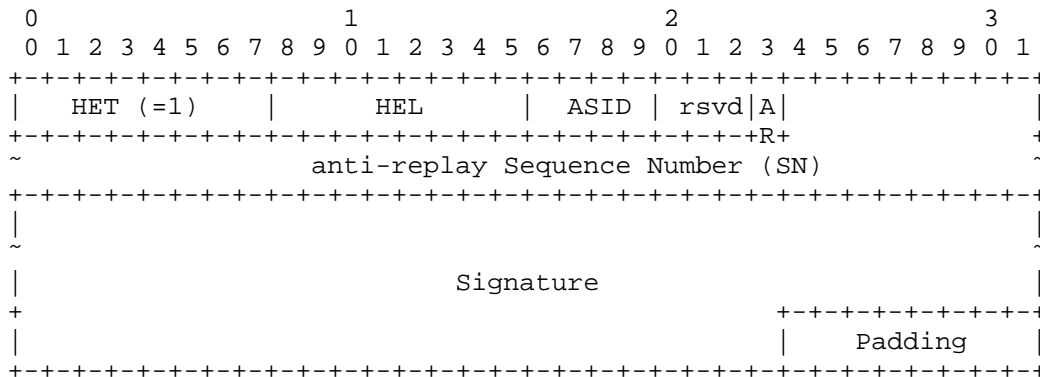
 Figure 1: Format of the Digital Signature EXT_AUTH Header Extension

The fields of the Digital Signature EXT_AUTH header extension are as
follows:

ASID (4 bits):

   The ASID identifies the source authentication scheme or protocol
   in use. The association between the ASID value and the actual
   authentication scheme is defined out-of-band, at session startup.

rsvd (Reserved) (3 bits):

   This is a reserved field that MUST be set to zero and ignored by
   receivers.

AR (anti-replay) (1 bit):

   The AR field, when set to 0, indicates that the anti-replay
   service is not used. When set to 1, it indicates that the
   anti-replay service is used.

SN (Sequence Number) (8 or 40 bits):

   The SN field contains an optional Sequence Number. When AR = 0,
   this is an 8-bit field that MUST be set to zero. No anti-replay
   mechanism is used in that case. When AR = 1, this is a 40-bit
   field (32 bits + 8 bits), and all of the 40 bits MUST be
   considered by the anti-replay mechanism.

   Signature (variable size, multiple of 32 bits):

      The Signature field contains a Digital Signature of the message.
      If need be, this field is padded (with 0) up to a multiple of
      32 bits.

3.2. Parameters

   Several parameters MUST be initialized by an out-of-band mechanism.
   The sender or group controller

   o MUST communicate its public key, for each receiver to be able to
      verify the signature of the packets received. For security
      reasons [RFC5751], the use of key sizes between 1024 and 2048 bits
      inclusive is RECOMMENDED. Key sizes inferior to 1024 bits SHOULD
      NOT be used. Key sizes above 2048 bits MAY be used. As a side
      effect, the receivers also know the key length and the signature
      length, the two parameters being equal;

   o MAY communicate a certificate (which also means that a PKI has
      been set up), for each receiver to be able to check the sender's
      public key;

   o MUST communicate the signature-encoding algorithm. For instance,
      [RFC3447] defines the RSASSA-PKCS1-v1_5 and RSASSA-PSS algorithms
      that are usually used for that purpose;

   o MUST communicate the One-way Hash Function -- for instance, SHA-1,
      SHA-224, SHA-256, SHA-384, or SHA-512. Because of security
      threats on SHA-1, the use of SHA-256 is RECOMMENDED [RFC6194];

   o MUST associate a value to the ASID field of the EXT_AUTH header
      extension (Section 3.1);

   o MUST communicate whether or not the anti-replay service is used
      for this session.

   These parameters MUST be communicated to all receivers before they
   can authenticate the incoming packets. For instance, it can be
   communicated in the session description, or initialized in a static
   way on the receivers, or communicated by means of an appropriate
   protocol. The details of this out-of-band mechanism are beyond the
   scope of this document.

3.3. Processing

3.3.1. Signature Processing

   The computation of the Digital Signature, using the private key, MUST
   include the ALC or NORM header (with the various header extensions)
   and the payload when applicable. The UDP/IP/MAC headers MUST NOT be
   included. During this computation, the Signature field MUST be set
   to 0.

   Several signature-encoding algorithms can be used, including
   RSASSA-PKCS1-v1_5 and RSASSA-PSS. With these encodings, several
   one-way hash functions can be used, like SHA-256.

   First, let us consider a packet sender. More specifically, as noted
   in [RFC4359], Digital Signature generation is performed as described
   in Section 8.2.1 of [RFC3447] (RSASSA-PKCS1-v1_5) and in
   Section 8.1.1 of [RFC3447] (RSASSA-PSS). The authenticated portion
   of the packet is used as the message M, which is passed to the
   signature generation function. The signer's RSA private key is
   passed as K. In summary (when SHA-256 is used), the signature
   generation process computes a SHA-256 hash of the authenticated
   packet bytes, signs the SHA-256 hash using the private key, and
   encodes the result with the specified RSA encoding type. This
   process results in a value S, which is the Digital Signature to be
   included in the packet.

   With RSASSA-PKCS1-v1_5 and RSASSA-PSS signatures, the size of the
   signature is equal to the "RSA modulus", unless the RSA modulus is
   not a multiple of 8 bits. In that case, the Digital Signature (also
   called the Integrity Check Value (ICV) in [RFC4359]) MUST be
   prepended with between 1 and 7 bits set to zero such that the Digital
   Signature is a multiple of 8 bits [RFC4359]. The key length, which
   in practice is also equal to the RSA modulus, has major security
   implications. [RFC4359] explains how to choose this value, depending
   on the maximum expected lifetime of the session. This choice is
   beyond the scope of this document.

   Now, let us consider a receiver. As noted in [RFC4359], Digital
   Signature verification is performed as described in Section 8.2.2 of
   [RFC3447] (RSASSA-PKCS1-v1_5) and Section 8.1.2 of [RFC3447]
   (RSASSA-PSS). Upon receipt, the Digital Signature is passed to the
   verification function as S. The authenticated portion of the packet
   is used as the message M, and the RSA public key is passed as (n, e).
   In summary (when SHA-256 is used), the verification function computes
   a SHA-256 hash of the authenticated packet bytes, decrypts the
   SHA-256 hash in the packet using the sender's public key, and

validates that the appropriate encoding was applied. The two SHA-256
hashes are compared, and if they are identical, the validation is
successful.

3.3.2. Anti-Replay Processing

Let us assume the anti-replay service is used. The principles are
similar to the Sequence Number mechanism described in [RFC4303], with
the exception that the present document uses a 40-bit field that
contains all the bits of the Sequence Number counter.

At the sender, the mechanism works as follows (Section 2.2 of
[RFC4303]). The sender's Sequence Number counter is initialized to 0
at session startup. The sender increments the Sequence Number
counter for this session and inserts the value into the SN field.
Thus, the first packet sent will contain an SN of 1. The SN value of
the Authentication Header extension MUST be initialized before the
signature generation process, in order to enable a receiver to check
the SN value during the integrity verification process.

The sender SHOULD ensure that the counter does not cycle before
inserting the new value in the SN field. Failing to follow this rule
would enable an attacker to replay a packet sent during the previous
cycle; i.e., it would limit the anti-replay service to a single SN
cycle. Since the Sequence Number is contained in a 40-bit field, it
is expected that cycling will never happen in most situations. For
instance, on a 10-Gbps network, with small packets (i.e., 64 bytes
long), cycling will happen after slightly more than 15 hours.

At the receiver, the mechanism works as follows (Section 3.4.3 and
Appendix A2 of [RFC4303]). For each received packet, the receiver
MUST verify that the packet contains a Sequence Number that does not
duplicate the Sequence Number of any other packets received during
the session. If this preliminary check fails, the packet is
discarded, thus avoiding the need for any cryptographic operations by
the receiver. If the preliminary check is successful, the receiver
cannot yet modify its local counter, because the integrity of the
Sequence Number has not been verified at this point.

Duplicates are rejected through the use of a sliding receive window.
The "right" edge of the window represents the highest, validated
Sequence Number value received on this session. Packets that contain
Sequence Numbers lower than the "left" edge of the window are
rejected. Packets falling within the window are checked against a
list of received packets within the window (how this list is managed
is a local, implementation-based decision). This window limits how
far out of order a packet can be, relative to the packet with the
highest Sequence Number that has been authenticated so far.

If the received packet falls within the window and is not a
duplicate, or if the packet is to the right of the window, then the
receiver proceeds to integrity verification. If the integrity check
fails, the receiver MUST discard the received packet as invalid;
otherwise, the receive window is updated and packet processing
continues.

3.4. In Practice

Each packet sent MUST contain exactly one Digital Signature EXT_AUTH
header extension. A receiver MUST drop all the packets that do not
contain a Digital Signature EXT_AUTH header extension.

All receivers MUST recognize EXT_AUTH but might not be able to parse
its content, for instance, because they do not support Digital
Signatures. In that case, the Digital Signature EXT_AUTH header
extension is ignored.

If the anti-replay mechanism is used, each packet sent MUST contain a
valid Sequence Number. All the packets that fail to contain a valid
Sequence Number MUST be immediately dropped.

For instance, Figure 2 shows the Digital Signature EXT_AUTH header
extension when using 128-byte (1024-bit) key Digital Signatures
(which also means that the Signature field is 128 bytes long). The
Digital Signature EXT_AUTH header extension is then 132 bytes long.

```
   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
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |   HET (=1)    |   HEL (=33)   | ASID | 0   |0|       0        |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---
  |                                                               | ^ 1
  +                                                               + | 2
  |                                                               | | 8
  .                                                               . |
  .                      Signature (128 bytes)                    . | b
  .                                                               . | y
  |                                                               | | t
  +                                                               + | e
  |                                                               | v s
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---
```
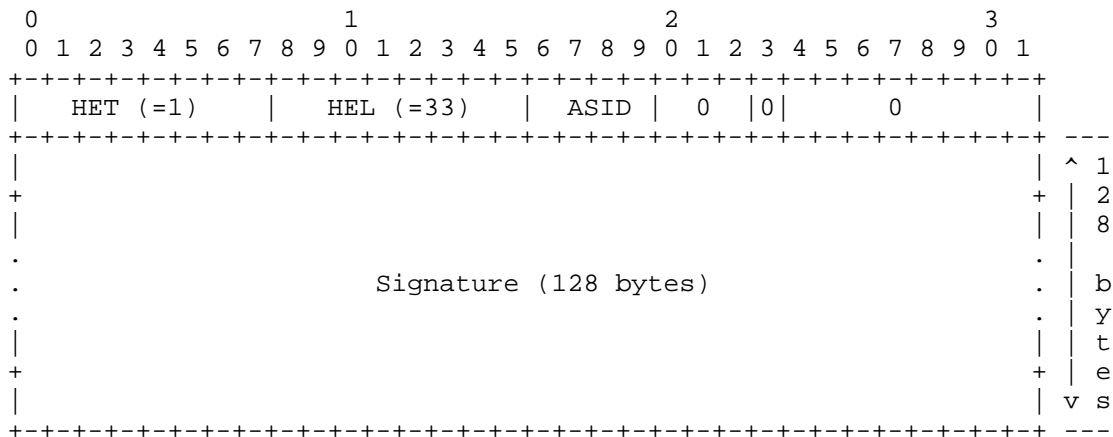
            Figure 2: Example: Format of the Digital Signature EXT_AUTH
                   Header Extension Using 1024-Bit Signatures,
                      without Any Anti-Replay Protection

4. Elliptic Curve Digital Signature Scheme

   This document focuses on the Elliptic Curve Digital Signature
   Algorithm (ECDSA). However, [RFC6090] describes alternative elliptic
   curve techniques, like KT-I signatures. The use of such alternatives
   is not considered in this document, but may be added in the future.

4.1. Authentication Header Extension Format

   The integration of ECC Digital Signatures is similar to that of RSA
   Digital Signatures. Several fields are added, in addition to the HET
   and HEL fields, as illustrated in Figure 1.

   The fields of the Digital Signature EXT_AUTH header extension are as
   follows:

   ASID (4 bits):

      The ASID identifies the source authentication scheme or protocol
      in use. The association between the ASID value and the actual
      authentication scheme is defined out-of-band, at session startup.

   rsvd (3 bits):

      This is a reserved field that MUST be set to zero and ignored by
      receivers.

   AR (1 bit):

      The AR field, when set to 0, indicates that the anti-replay
      service is not used. When set to 1, it indicates that the
      anti-replay service is used.

   SN (8 or 40 bits):

      The SN field contains an optional Sequence Number. When AR = 0,
      this is an 8-bit field that MUST be set to zero. No anti-replay
      mechanism is used in that case. When AR = 1, this is a 40-bit
      field (32 bits + 8 bits), and all of the 40 bits MUST be
      considered by the anti-replay mechanism.

   Signature (variable size, multiple of 32 bits):

      The Signature field contains a Digital Signature of the message.
      If need be, this field is padded (with 0) up to a multiple of
      32 bits.

4.2. Parameters

   Several parameters MUST be initialized by an out-of-band mechanism.
   The sender or group controller

   o MUST communicate its public key, for each receiver to be able to
      verify the signature of the packets received. As a side effect,
      the receivers also know the key length and the signature length,
      the two parameters being equal;

   o MAY communicate a certificate (which also means that a PKI has
      been set up), for each receiver to be able to check the sender's
      public key;

   o MUST communicate the message digest algorithm;

   o MUST communicate the elliptic curve;

   o MUST associate a value to the ASID field of the EXT_AUTH header
      extension (Section 3.1);

   o MUST communicate whether or not the anti-replay service is used
      for this session.

   These parameters MUST be communicated to all receivers before they
   can authenticate the incoming packets. For instance, it can be
   communicated in the session description, or initialized in a static
   way on the receivers, or communicated by means of an appropriate
   protocol. The details of this out-of-band mechanism are beyond the
   scope of this document.

4.3. Processing

4.3.1. Signature Processing

   The computation of the ECC Digital Signature, using the private key,
   MUST include the ALC or NORM header (with the various header
   extensions) and the payload when applicable. The UDP/IP/MAC headers
   MUST NOT be included. During this computation, the Signature field
   MUST be set to 0.

   Several elliptic curve groups can be used, as well as several hash
   algorithms. In practice, both choices are related, and there is a
   minimum hash algorithm size for any key length. Using a larger hash
   algorithm and then truncating the output is also feasible; however,

it consumes more processing power than is necessary. In order to
promote interoperability, [RFC4754] and [RFC5480] list several
possible choices (see table below).

+----------------------------+--------+------------------+-----------+
| Digital Signature | Key | Message Digest | Elliptic |
| Algorithm Name [RFC4754] | Size | Algorithm | Curve |
+----------------------------+--------+------------------+-----------+
ECDSA-256 (default)	256	SHA-256	secp256r1
ECDSA-384	384	SHA-384	secp384r1
ECDSA-521	512	SHA-512	secp521r1
+----------------------------+--------+------------------+-----------+

ECDSA-256, ECDSA-384, and ECDSA-521 are designed to offer security
comparable with AES-128, AES-192, and AES-256, respectively
[RFC4754]. Among them, the use of ECDSA-256/secp256r1 is
RECOMMENDED.

4.3.2. Anti-Replay Processing

The anti-replay processing follows the principles described in
Section 3.3.2.

4.4. In Practice

Each packet sent MUST contain exactly one ECC Digital Signature
EXT_AUTH header extension. A receiver MUST drop all the packets that
do not contain an ECC Digital Signature EXT_AUTH header extension.

All receivers MUST recognize EXT_AUTH but might not be able to parse
its content, for instance, because they do not support ECC Digital
Signatures. In that case, the Digital Signature EXT_AUTH header
extension is ignored.

If the anti-replay mechanism is used, each packet sent MUST contain a
valid Sequence Number. All the packets that fail to contain a valid
Sequence Number MUST be immediately dropped.

For instance, Figure 3 shows the Digital Signature EXT_AUTH header
extension when using ECDSA-256 (256-bit) ECC Digital Signatures.
The ECC Digital Signature EXT_AUTH header extension is then 36 bytes
long.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   HET (=1)    |   HEL (=9)    | ASID | 0  |0|      0          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---
|                                                               | ^  3
+                                                               + |  2
.                                                               . |
.                      Signature (32 bytes)                     . |  b
.                                                               . |  y
|                                                               | |  t
+                                                               + |  e
|                                                               | v  s
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---
```
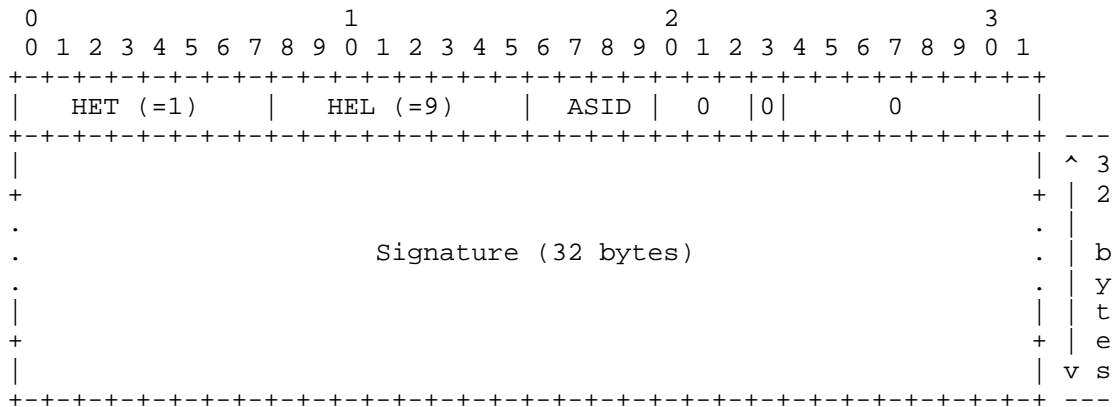
                Figure 3: Example: Format of the ECC Digital Signature EXT_AUTH
                      Header Extension Using ECDSA-256 Signatures,
                         without Any Anti-Replay Protection

5. Group-Keyed Message Authentication Code (MAC) Scheme

5.1. Authentication Header Extension Format

    The integration of Group-keyed MAC is similar in ALC and NORM and
    relies on the header extension mechanism defined in both protocols.
    More precisely, this document details the HET=1 (EXT_AUTH) header
    extension defined in [RFC5651].

Several fields are added, in addition to the HET and HEL fields
(Figure 4).

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   HET (=1)    |      HEL      | ASID | rsvd|A|               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+R+               +
~                 anti-replay Sequence Number (SN)            ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                             |
~                                                             ~
|                      Group-keyed MAC                        |
+                                             +-+-+-+-+-+-+-+-+
|                                             |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
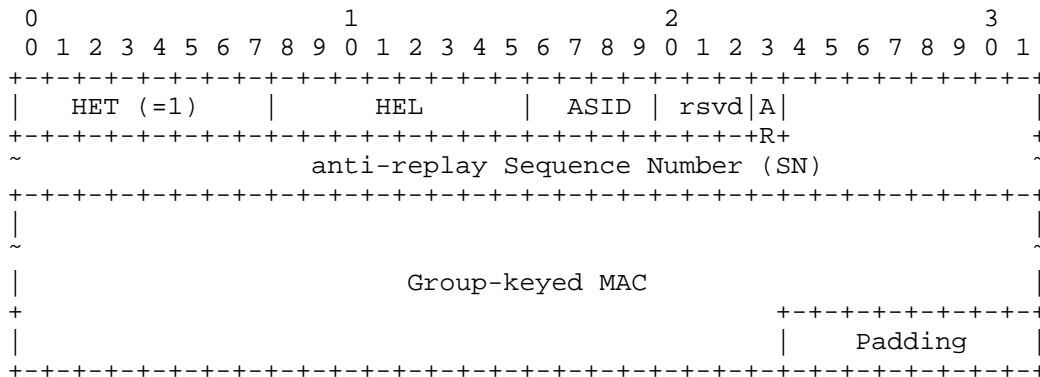
   Figure 4: Format of the Group-Keyed MAC EXT_AUTH Header Extension

The fields of the Group-keyed MAC EXT_AUTH header extension are as
follows:

ASID (4 bits):

   The ASID identifies the source authentication scheme or protocol
   in use. The association between the ASID value and the actual
   authentication scheme is defined out-of-band, at session startup.

rsvd (3 bits):

   This is a reserved field that MUST be set to zero and ignored by
   receivers.

AR (1 bit):

   The AR field, when set to 0, indicates that the anti-replay
   service is not used. When set to 1, it indicates that the
   anti-replay service is used.

SN (8 or 40 bits):

   The SN field contains an optional Sequence Number. When AR = 0,
   this is an 8-bit field that MUST be set to zero. No anti-replay
   mechanism is used in that case. When AR = 1, this is a 40-bit
   field (32 bits + 8 bits), and all of the 40 bits MUST be
   considered by the anti-replay mechanism.

Group-keyed MAC (variable size, multiple of 32 bits):

> The Group-keyed MAC field contains a truncated Group-keyed MAC of
> the message. If need be, this field is padded (with 0) up to a
> multiple of 32 bits.

5.2. Parameters

Several parameters MUST be initialized by an out-of-band mechanism.
The sender or group controller

o MUST communicate the Cryptographic MAC Function -- for instance,
   HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, or
   HMAC-SHA-512. As a side effect, with these functions, the
   receivers also know the key length and the non-truncated MAC
   output length. Because of security threats on SHA-1, the use of
   HMAC-SHA-256 is RECOMMENDED [RFC6194];

o MUST communicate the length of the truncated output of the MAC,
   n_m, which depends on the Cryptographic MAC Function chosen. Only
   the n_m leftmost bits (most significant bits) of the MAC output
   are kept. Of course, n_m MUST be less than or equal to the key
   length;

o MUST communicate the group key to the receivers, confidentially,
   before starting the session. This key might have to be
   periodically refreshed for improved robustness;

o MUST associate a value to the ASID field of the EXT_AUTH header
   extension (Section 5.1);

o MUST communicate whether or not the anti-replay service is used
   for this session.

These parameters MUST be communicated to all receivers before they
can authenticate the incoming packets. For instance, it can be
communicated in the session description, or initialized in a static
way on the receivers, or communicated by means of an appropriate
protocol (this will often be the case when periodic re-keying is
required). The details of this out-of-band mechanism are beyond the
scope of this document.

5.3. Processing

5.3.1. Signature Processing

   The computation of the Group-keyed MAC, using the group key, includes
   the ALC or NORM header (with the various header extensions) and the
   payload when applicable. The UDP/IP/MAC headers are not included.
   During this computation, the weak Group-keyed MAC field MUST be set
   to 0. Then, the sender truncates the MAC output to keep the n_m most
   significant bits and stores the result in the Group-keyed MAC
   Authentication Header.

   Upon receiving this packet, the receiver computes the Group-keyed
   MAC, using the group key, and compares it to the value carried in the
   packet. During this computation, the Group-keyed MAC field MUST also
   be set to 0. If the check fails, the packet MUST be immediately
   dropped.

   [RFC2104] explains that it is current practice to truncate the MAC
   output, on condition that the truncated output length, n_m, be not
   less than half the length of the hash and not less than 80 bits.
   However, this choice is beyond the scope of this document.

5.3.2. Anti-Replay Processing

   The anti-replay processing follows the principles described in
   Section 3.3.2.

5.4. In Practice

   Each packet sent MUST contain exactly one Group-keyed MAC EXT_AUTH
   header extension. A receiver MUST drop packets that do not contain a
   Group-keyed MAC EXT_AUTH header extension.

   All receivers MUST recognize EXT_AUTH but might not be able to parse
   its content, for instance, because they do not support Group-keyed
   MAC. In that case, the Group-keyed MAC EXT_AUTH extension is
   ignored.

   If the anti-replay mechanism is used, each packet sent MUST contain a
   valid Sequence Number. All the packets that fail to contain a valid
   Sequence Number MUST be immediately dropped.

For instance, Figure 5 shows the Group-keyed MAC EXT_AUTH header
extension when using HMAC-SHA-256. The Group-keyed MAC EXT_AUTH
header extension is then 16 bytes long.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    HET (=1)    |    HEL (=4)    | ASID |  0  |0|      0       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                   Group-keyed MAC (16 bytes)                  |
+                                                               +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
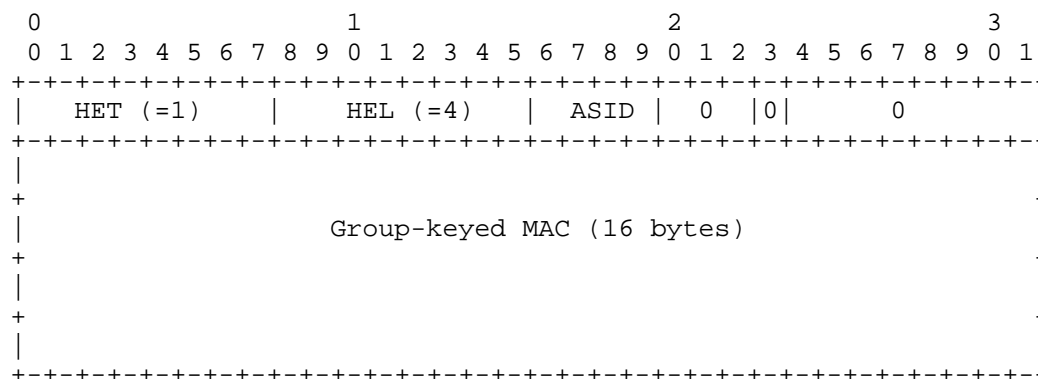
   Figure 5: Example: Format of the Group-Keyed MAC EXT_AUTH Header
   Extension Using HMAC-SHA-256, without Any Anti-Replay Protection

6. Combined Use of the RSA/ECC Digital Signatures and Group-Keyed MAC
    Schemes

6.1. Authentication Header Extension Format

    The integration of combined RSA/ECC Digital Signatures and
    Group-keyed MAC schemes is similar in ALC and NORM and relies on the
    header extension mechanism defined in both protocols. More
    precisely, this document details the HET=1 (EXT_AUTH) header
    extension defined in [RFC5651].

Several fields are added, in addition to the HET and HEL fields
(Figure 6).

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   HET (=1)    |      HEL      | ASID | rsvd|A|               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+R+               +
|                 anti-replay Sequence Number (SN)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                             |
~                                                             ~
|                         Signature                           |
+                                             +-+-+-+-+-+-+-+-+
|                                             |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Group-keyed MAC                        |
~                                                             ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Figure 6: Format of the Group-Keyed MAC EXT_AUTH Header Extension

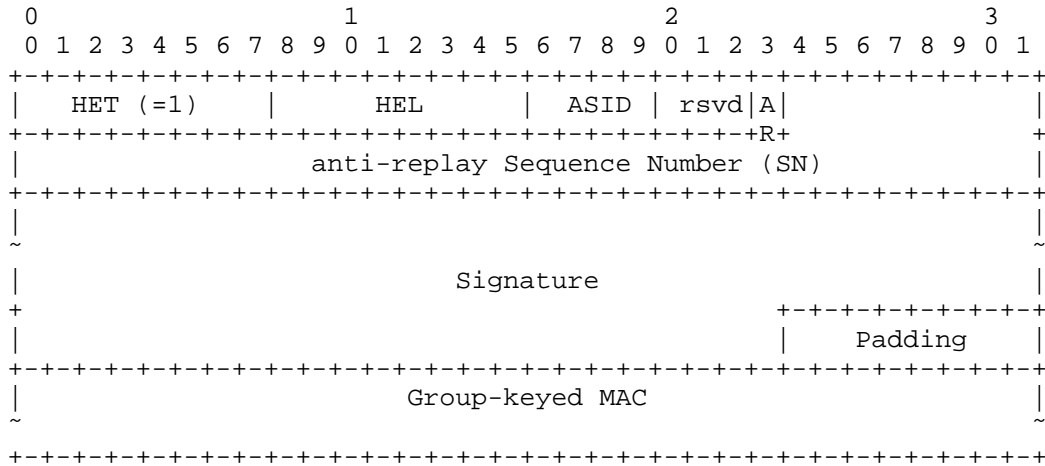The fields of the Group-keyed MAC EXT_AUTH header extension are as
follows:

ASID (4 bits):

   The ASID identifies the source authentication scheme or protocol
   in use. The association between the ASID value and the actual
   authentication scheme is defined out-of-band, at session startup.

rsvd (3 bits):

   This is a reserved field that MUST be set to zero and ignored by
   receivers.

AR (1 bit):

   The AR field MUST be set to 1, indicating that the anti-replay
   service is used (see Section 6.3).

SN (8 or 40 bits):

   The SN field contains a Sequence Number. Since AR = 1, this is a
   40-bit field (32 bits + 8 bits), and all of the 40 bits MUST be
   considered by the anti-replay mechanism.

Signature (variable size, multiple of 32 bits):

> The Signature field contains a Digital Signature of the message.
> If need be, this field is padded (with 0) up to a multiple of
> 32 bits.

Group-keyed MAC (variable size, multiple of 32 bits, by default
32 bits):

> The Group-keyed MAC field contains a truncated Group-keyed MAC of
> the message.

6.2. Parameters

Several parameters MUST be initialized by an out-of-band mechanism,
as defined in Sections 3.2, 4.2, and 5.2.

6.3. Processing

In some situations, it can be interesting to use both authentication
schemes. The goal of the Group-keyed MAC is to mitigate denial-of-
service (DoS) attacks coming from attackers that are not group
members [RFC4082], by adding a light authentication scheme as a
front-end.

6.3.1. Signature Processing

Before sending a message, the sender sets the Signature field and
Group-keyed MAC field to zero. Then, the sender computes the
signature as detailed in Section 3.3 or in Section 4.3 and stores the
value in the Signature field. Then, the sender computes the
Group-keyed MAC as detailed in Section 5.3 and stores the value in
the Group-keyed MAC field. The (RSA or ECC) Digital Signature value
is therefore protected by the Group-keyed MAC, which avoids DoS
attacks where the attacker corrupts the Digital Signature itself.

Upon receiving the packet, the receiver first checks the Group-keyed
MAC, as detailed in Section 5.3. If the check fails, the packet MUST
be immediately dropped. Otherwise, the receiver checks the Digital
Signature, as detailed in Section 3.3. If the check fails, the
packet MUST be immediately dropped.

   This scheme features a few limits:

   o The Group-keyed MAC is of no help if a group member (who knows the
      group key) impersonates the sender and sends forged messages to
      other receivers. DoS attacks are still feasible;

   o It requires an additional MAC computing for each packet, both at
      the sender and receiver sides;

   o It increases the size of the Authentication Headers. In order to
      limit this problem, the length of the truncated output of the MAC,
      n_m, SHOULD be kept small (see Section 9.5 of [RFC3711]). In the
      current specification, n_m MUST be a multiple of 32 bits, and the
      default value is 32 bits. As a side effect, with n_m = 32 bits,
      the authentication service is significantly weakened, since the
      probability that any packet would be successfully forged is one in
      2^32. Since the Group-keyed MAC check is only a pre-check that is
      followed by the standard signature authentication check, this is
      not considered to be an issue.

   For a given use case, the benefits brought by the Group-keyed MAC
   must be balanced against these limitations.

6.3.2. Anti-Replay Processing

   The anti-replay processing follows the principles described in
   Section 3.3.2. Here, an anti-replay service MUST be used. Indeed,
   failing to enable anti-replay protection would facilitate DoS
   attacks, since all replayed (but otherwise valid) packets would pass
   the light authentication scheme and oblige a receiver to perform a
   complex signature verification.

6.4. In Practice

   Each packet sent MUST contain exactly one combined Digital Signature/
   Group-keyed MAC EXT_AUTH header extension. A receiver MUST drop
   packets that do not contain a combined Digital Signature/Group-keyed
   MAC EXT_AUTH header extension.

   All receivers MUST recognize EXT_AUTH but might not be able to parse
   its content, for instance, because they do not support combined
   Digital Signature/Group-keyed MAC. In that case, the combined
   Digital Signature/Group-keyed MAC EXT_AUTH extension is ignored.

   Since the anti-replay mechanism MUST be used, each packet sent MUST
   contain a valid Sequence Number. All the packets that fail to
   contain a valid Sequence Number MUST be immediately dropped.

It is RECOMMENDED that the n_m parameter of the group authentication
scheme be small, and by default equal to 32 bits (Section 6.3).

For instance, Figure 7 shows the combined Digital Signature/
Group-keyed MAC EXT_AUTH header extension when using 128-byte
(1024-bit) key RSA Digital Signatures (which also means that the
Signature field is 128 bytes long). The EXT_AUTH header extension is
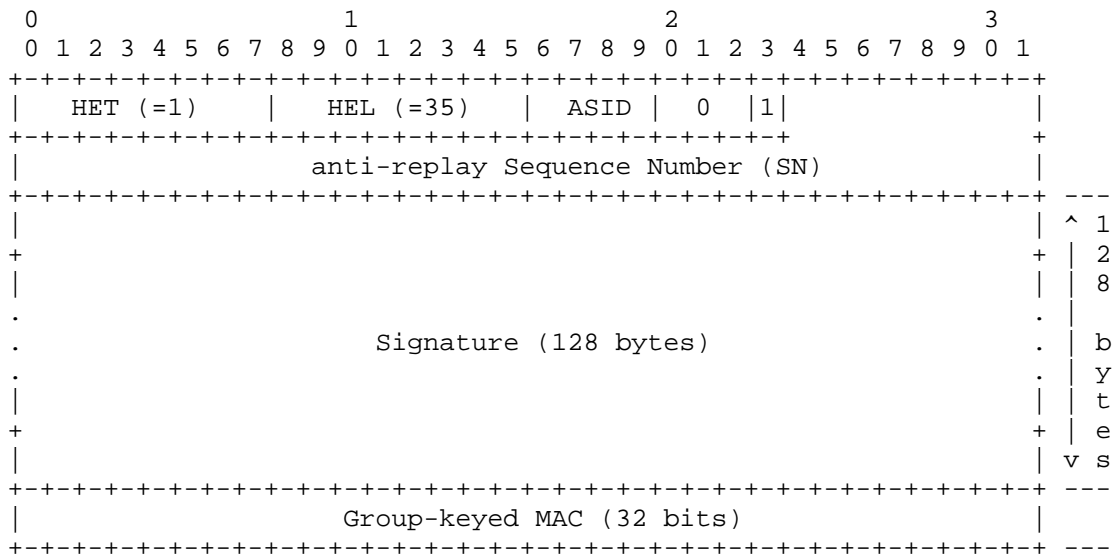then 140 bytes long.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   HET (=1)    |   HEL (=35)   | ASID |  0  |1|              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+              +
|               anti-replay Sequence Number (SN)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---
|                                                            | ^ 1
+                                                            + | 2
|                                                            | | 8
.                                                            . |
.               Signature (128 bytes)                        . | b
.                                                            . | y
|                                                            | | t
+                                                            + | e
|                                                            | v s
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---
|                   Group-keyed MAC (32 bits)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---
```

Figure 7: Example: Format of the Combined RSA Digital Signature/
Group-Keyed MAC EXT_AUTH Header Extension Using 1024-Bit Signatures,
with Anti-Replay Protection

7. Security Considerations

7.1. Dealing with DoS Attacks

   Let us consider packets secured through the use of a Digital
   Signature scheme first. Because faked packets are easy to create but
   checking them requires computation of a costly Digital Signature,
   this scheme introduces new opportunities for an attacker to mount DoS
   attacks. More precisely, an attacker can easily saturate the
   processing capabilities of the receiver.

   In order to mitigate these attacks, it is RECOMMENDED that the
   combined Digital Signature/Group-keyed MAC scheme (Section 6.3) be
   used. However, no mitigation is possible if a group member acts as
   an attacker. Additionally, even if checking a Group-keyed MAC is

significantly faster than checking a Digital Signature, there are
practical limits on how many Group-keyed MACs can be checked per time
unit. Therefore, it is RECOMMENDED that limiting the number of
authentication checks per time unit be done when the number of
incoming packets that fail the authentication check exceeds a given
threshold (i.e., in the case of a DoS attack).

The RECOMMENDED action of limiting the number of checks per time unit
under (presumed) attack situations can be extended to the other
authentication schemes.

7.2. Dealing with Replay Attacks

Replay attacks involve an attacker storing a valid message and
replaying it later. It is RECOMMENDED that the anti-replay service
defined in this document be used with the signature and Group-keyed
MAC solutions, and this anti-replay service MUST be used in the case
of a combined use of signatures and Group-keyed MAC schemes (see
Section 6.3.2).

The following section details some of the potential consequences of
not using anti-replay protection.

7.2.1. Impacts of Replay Attacks on the Simple Authentication Schemes

Since all the above authentication schemes are stateless, replay
attacks have no impact on these schemes.

7.2.2. Impacts of Replay Attacks on NORM

In this subsection, we review the potential impacts of a replay
attack on the NORM component. Note that we do not consider here the
protocols that could be used along with NORM -- for instance,
congestion control protocols.

First, let us consider replay attacks within a given NORM session.
As NORM is a stateful protocol, replaying a packet may have
consequences.

NORM defines a "sequence" field that may be used to protect against
replay attacks [RFC5740] within a given NORM session. This sequence
field is a 16-bit value that is set by the message originator (sender
or receiver) as a monotonically increasing number incremented with
each NORM message transmitted. Using this field for anti-replay
protection would be possible if there is no wrapping to zero, i.e.,
would only be possible if at most 65535 packets are sent; this may be
true for some use cases but not for the general case. Using this

field for anti-replay protection would also be possible if the keying
material is updated before wrapping to zero happens; this may be true
for some use cases but not for the general case.

Now, let us consider replay attacks across several NORM sessions. A
host participating in a NORM session is uniquely identified by the
{source_id; instance_id} tuple. Therefore, when a given host
participates in several NORM sessions, it is RECOMMENDED that
instance_id be changed for each NORM instance. It is also
RECOMMENDED, when the Group-keyed MAC authentication/integrity check
scheme is used, that the shared group key be changed across sessions.
Therefore, NORM can be made robust when confronted with replay
attacks across different sessions.

7.2.3. Impacts of Replay Attacks on ALC

   In this subsection, we review the potential impacts of a replay
   attack on the ALC component. Note that we do not consider here the
   protocols that could be used along with ALC -- for instance, layered
   or wave-based congestion control protocols.

   First, let us consider replay attacks within a given ALC session:

   o Replayed encoding symbol: A replayed encoding symbol (coming from
      a replayed data packet) is detected, thanks to the object/block/
      symbol identifiers, and is silently discarded.

   o Replayed control information:

      * At the end of the session, a "close session" (A flag) packet is
         sent. Replaying a packet containing this flag has no impact,
         since the receivers have already left the session.

      * Similarly, replaying a packet containing a "close object"
         (B flag) has no impact, since this object is probably already
         marked as closed by the receiver.

      * Timing information sent as part of a Layered Coding Transport
         (LCT) EXT_TIME header extension [RFC5651] may be more sensitive
         to replay attacks. For instance, replaying a packet containing
         an ERT (Expected Residual Time) may mislead a receiver to
         believe an object transmission will continue for some time
         whereas the transmission of symbols for this object is about to
         stop. Replaying a packet containing a Sender Current Time
         (SCT) is easily identified if the receiver verifies that time
         progresses upon receiving such EXT_TIME header extensions.

> Replaying a packet containing a Session Last Changed (SLC) is
> easily identified if the receiver verifies the chronology upon
> receiving such EXT_TIME header extensions.

This analysis shows that ALC might be, to a limited extent, sensitive
to replay attacks within the same session if timing information is
used. Otherwise, ALC is robust when confronted with replay attacks
within the same session.

Now, let us consider replay attacks across several ALC sessions. An
ALC session is uniquely identified by the {sender IP address; TSI}
tuple. Therefore, when a given sender creates several sessions, the
TSI MUST be changed for each ALC session, so that each TSI is unique
among all active sessions of this sender and for a long period of
time preceding and following when the session is active [RFC5651].
Therefore, ALC can be made robust when confronted with replay attacks
across different sessions. Of course, when the Group-keyed MAC
authentication/integrity check scheme is used, the shared group key
SHOULD be changed across sessions if the set of receivers changes.

7.3. Dealing with Attacks on the Parameters Sent Out-of-Band

This specification requires that several parameters be communicated
to the receiver(s) via an out-of-band mechanism that is beyond the
scope of this document. This is in particular the case for the
mapping between an ASID value and the associated authentication
scheme (Section 1). Since this mapping is critical, this information
SHOULD be carried in a secure way from the sender to the receiver(s).

8. Acknowledgments

The author is grateful to the authors of [RFC4303], [RFC4359],
[RFC4754], and [RFC5480]; their documents inspired several sections
of the present document. The author is also grateful to all the IESG
members, and in particular to David Harrington, Stephen Farrell, and
Sean Turner for their very detailed reviews.

9. References

9.1. Normative References

   [RFC2104]   Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-
                 Hashing for Message Authentication", RFC 2104,
                 February 1997.

   [RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
                 Requirement Levels", BCP 14, RFC 2119, March 1997.

   [RFC5651]   Luby, M., Watson, M., and L. Vicisano, "Layered Coding
                  Transport (LCT) Building Block", RFC 5651, October 2009.

   [RFC5740]   Adamson, B., Bormann, C., Handley, M., and J. Macker,
                  "NACK-Oriented Reliable Multicast (NORM) Transport
                  Protocol", RFC 5740, November 2009.

   [RFC5775]   Luby, M., Watson, M., and L. Vicisano, "Asynchronous
                  Layered Coding (ALC) Protocol Instantiation", RFC 5775,
                  April 2010.

9.2. Informative References

   [RFC3447]   Jonsson, J. and B. Kaliski, "Public-Key Cryptography
                  Standards (PKCS) #1: RSA Cryptography Specifications
                  Version 2.1", RFC 3447, February 2003.

   [RFC3711]   Baugher, M., McGrew, D., Naslund, M., Carrara, E., and
                  K. Norrman, "The Secure Real-time Transport Protocol
                  (SRTP)", RFC 3711, March 2004.

   [RFC4082]   Perrig, A., Song, D., Canetti, R., Tygar, J., and B.
                  Briscoe, "Timed Efficient Stream Loss-Tolerant
                  Authentication (TESLA): Multicast Source Authentication
                  Transform Introduction", RFC 4082, June 2005.

   [RFC4303]   Kent, S., "IP Encapsulating Security Payload (ESP)",
                  RFC 4303, December 2005.

   [RFC4359]   Weis, B., "The Use of RSA/SHA-1 Signatures within
                  Encapsulating Security Payload (ESP) and Authentication
                  Header (AH)", RFC 4359, January 2006.

   [RFC4754]   Fu, D. and J. Solinas, "IKE and IKEv2 Authentication
                  Using the Elliptic Curve Digital Signature Algorithm
                  (ECDSA)", RFC 4754, January 2007.

   [RFC5480]   Turner, S., Brown, D., Yiu, K., Housley, R., and T.
                  Polk, "Elliptic Curve Cryptography Subject Public Key
                  Information", RFC 5480, March 2009.

   [RFC5751]   Ramsdell, B. and S. Turner, "Secure/Multipurpose
                  Internet Mail Extensions (S/MIME) Version 3.2 Message
                  Specification", RFC 5751, January 2010.

   [RFC5776]   Roca, V., Francillon, A., and S. Faurite, "Use of Timed
                Efficient Stream Loss-Tolerant Authentication (TESLA) in
                the Asynchronous Layered Coding (ALC) and NACK-Oriented
                Reliable Multicast (NORM) Protocols", RFC 5776,
                April 2010.

   [RFC6090]   McGrew, D., Igoe, K., and M. Salter, "Fundamental
                Elliptic Curve Cryptography Algorithms", RFC 6090,
                February 2011.

   [RFC6194]   Polk, T., Chen, L., Turner, S., and P. Hoffman,
                "Security Considerations for the SHA-0 and SHA-1
                Message-Digest Algorithms", RFC 6194, March 2011.

   [RMT-FLUTE] Paila, T., Walsh, R., Luby, M., Roca, V., and R.
                Lehtonen, "FLUTE - File Delivery over Unidirectional
                Transport", Work in Progress, March 2012.

Author's Address

   Vincent Roca
   INRIA
   655, av. de l'Europe
   Inovallee; Montbonnot
   ST ISMIER cedex 38334
   France

   EMail: vincent.roca@inria.fr
   URI: http://planete.inrialpes.fr/people/roca/