RMT	V. Roca
Internet-Draft	INRIA
Intended status: Standards Track	October 26, 2009
Expires: April 29, 2010	

Simple Authentication Schemes for the ALC and NORM Protocols
draft-ietf-rmt-simple-auth-for-alc-norm-02

Status of this Memo

Copyright Notice

Abstract

This document introduces four schemes that provide a per-packet authentication and integrity service in the context of the ALC and NORM protocols. The first scheme is based on digital signatures. Because it relies on asymmetric cryptography, this scheme generates a high processing load at the sender and to a lesser extent at a receiver, as well as a significant transmission overhead. It is therefore well suited to low data rate sessions. The second scheme relies on the Elliptic Curve Digital Signature Algorithm (ECDSA). If this approach also relies an asymmetric cryptography, the processing load and the transmission overhead are significantly reduced compared to traditional digital signature schemes. It is therefore well suited to medium data rate sessions. The third scheme relies on a group Message Authentication Code (MAC). Because this scheme relies on symmetric cryptography, MAC calculation and verification are fast operations, which makes it suited to high data rate sessions. However it only provides a group authentication and integrity service, which means that it only protects against attackers that are not group members. Finally, the fourth scheme merges the digital signature and group group schemes, and is useful to mitigate DoS attacks coming from attackers that are not group members.

Table of Contents

---

1. Introduction

TOC

Many applications using multicast and broadcast communications require that each receiver be able to authenticate the source of any packet it receives to check check its integrity. For instance, ALC [RMT-PI-ALC] (Luby, M., Watson, M., and L. Vicisano, "Asynchronous Layered Coding (ALC) Protocol Instantiation," October 2009.) and NORM [RMT-PI-NORM] (Adamson, B., Bormann, C., Handley, M., and J. Macker, "Negative-acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Protocol," September 2009.) are two Content Delivery Protocols (CDP) designed to transfer reliably objects (e.g. files) between a session's sender and several receivers.

The NORM protocol is based on bidirectional transmissions. Each receiver acknowledges data received or, in case of packet erasures, asks for retransmissions. On the opposite, the ALC protocol defines unidirectional transmissions. Reliability can be achieved by means of cyclic transmissions of the content within a carousel, or by the use of proactive Forward Error Correction codes (FEC), 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 application level, on top of an erasure channel (e.g. the Internet) where packets can be lost (erased) during the transmission.

With these CDP, 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 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:

* digital signatures [RFC4359] (Weis, B., “The Use of RSA/SHA-1 Signatures within Encapsulating Security Payload (ESP) and Authentication Header (AH),” January 2006.): this scheme is well suited to low data rate flows, when a true packet sender authentication and packet integrity service is needed. However, digital signatures based on RSA asymmetric cryptography is limited by high computational costs and high transmission overheads. The use of ECC ("Elliptic Curve Cryptography") significantly relaxes these constraints, especially when seeking for higher security levels. For instance, the following key sizes provide equivalent security: 1024 bit RSA key versus 160 bit ECC key, or 2048 bit RSA key versus 224 bit ECC key.

* group Message Authentication Codes (MAC): 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.

* TESLA (Timed Efficient Stream Loss-tolerant Authentication) [RFC4082] (Perrig, A., Song, D., Canetti, R., Tygar, J., and B. Briscoe, “Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction,” June 2005.)[MSEC-TESLA] (Roca, V., Francillon, A., and S. Faurite, “Use of TESLA in the ALC and NORM Protocols,” October 2009.): this scheme is well suited to high data rate flows, when transmission overheads must be minimized, and when a true packet sender authentication and packet integrity service is needed. The price to pay 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.

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

	RSA Digital Signature	**ECC Digital Signature**	**Group MAC**	**TESLA**
True auth and integrity	Yes	Yes	No (group security)	Yes

Immediate auth	Yes	Yes	Yes	No
Processing load	-	0	+	+
Transmission overhead	-	0	+	+
Complexity	+	+	+	-

Several authentication schemes MAY be used in the same ALC or NORM session, even on the same communication path. Since all the above schemes make use of the same authentication header extension mechanism (Section 2.3 (Authentication Header Extension Format), Section 4.3 (Authentication Header Extension Format), Section 5.3 (Authentication Header Extension Format)) and [MSEC-TESLA] (Roca, V., Francillon, A., and S. Faurite, “Use of TESLA in the ALC and NORM Protocols,” October 2009.), section 5.1), the same 4-bit "ASID" (Authentication Scheme IDentifier) has been reserved in all the specifications. The association between the "ASID" value and the actual authentication scheme is defined at session startup and communicated to all the group members by an out-of-band mechanism.

1.1. Scope of this Document

TOC

[MSEC-TESLA] (Roca, V., Francillon, A., and S. Faurite, “Use of TESLA in the ALC and NORM Protocols,” October 2009.) explains how to use TESLA in the context of 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 MAC schemes. The current document also specifies the joint use of Digital Signature and Group MAC schemes which is useful to mitigate DoS attacks coming from attackers that are not group members.

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).

All the applications build 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 FLUTE application [RMT-FLUTE] (Paila, T., Walsh, R., Luby, M., Lehtonen, R., and V. Roca, “FLUTE - File Delivery over Unidirectional Transport,” August 2009.) 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 Section 2.2 (Parameters) and Section 4.2 (Parameters).

1.2. Conventions Used in this Document

TOC

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] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).

1.3. Terminology and Notations

TOC

The following notations and definitions are used throughout this document:

- *MAC is the Message Authentication Code;
- *HMAC is the Keyed-Hash Message Authentication Code;
- *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 case of feedback traffic;
- *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 case of feedback traffic;

Digital signature related notations and definitions:

- *K_pub is the public key used by a receiver to check a packet's signature. This key MUST be communicated to all receivers, before starting the session.
- *K_priv is the private key used by a sender to generate a packet's signature;
- *n_k is the private key and public key length, in bits. n_k is also the signature length, since both values are equal with digital signatures;

Group MAC related notations and definitions:

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

*n_k is the group key length, in bits;

*n_m is the length of the truncated output of the MAC [RFC2104] (Krawczyk, H., Bellare, M., and R. Canetti, “HMAC: Keyed-Hashing for Message Authentication,” February 1997.). Only the n_m left-most bits (most significant bits) of the MAC output are kept;

---

2. RSA Digital Signature Scheme

TOC

2.1. Principles

TOC

The computation of the digital signature, using K_priv, 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.
Upon receiving this packet, the receiver recomputes the Group MAC, using K_pub, and compares it to the value carried in the packet. During this computation, the Weak Group MAC field MUST also be set to 0. If the check fails, the packet MUST be immediately dropped.
Several "Signature Encoding Algorithms" can be used, including RSASSA-PKCS1-v1_5 and RSASSA-PSS. With these encodings, several "Signature Cryptographic Function" can be used, like SHA-256.
First, let us consider a packet sender. More specifically, from [RFC4359] (Weis, B., “The Use of RSA/SHA-1 Signatures within Encapsulating Security Payload (ESP) and Authentication Header (AH),” January 2006.): digital signature generation is performed as described in [RFC3447] (Jonsson, J. and B. Kaliski, “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1,” February 2003.), Section 8.2.1 for RSASSA-PKCS1-v1_5 and Section 8.1.1 for 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 signature MUST be prepended with between 1 and 7 bits set to zero such that the signature is a multiple of 8 bits [RFC4359] (Weis, B., “The Use of RSA/SHA-1

Signatures within Encapsulating Security Payload (ESP) and Authentication Header (AH)," January 2006.). The key size, which in practice is also equal to the "RSA modulus", has major security implications. [RFC4359] (Weis, B., "The Use of RSA/SHA-1 Signatures within Encapsulating Security Payload (ESP) and Authentication Header (AH)," January 2006.) explains how to choose this value depending on the maximum expected lifetime of the session. This choice is out of the scope of this document.

Now let us consider a receiver. From [RFC4359] (Weis, B., "The Use of RSA/SHA-1 Signatures within Encapsulating Security Payload (ESP) and Authentication Header (AH)," January 2006.): Digital signature verification is performed as described in [RFC3447] (Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1," February 2003.), Section 8.2.2 (RSASSA-PKCS1-v1_5) and [RFC3447] (Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1," February 2003.), Section 8.1.2 (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.

2.2. Parameters

TOC

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

* MUST communicate his 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, n_k, and the signature length, the two parameters being equal;
* MAY communicate a certificate (which also means that a PKI has been setup), for each receiver to be able to check the sender's public key;
* MUST communicate the Signature Encoding Algorithm. For instance, [RFC3447] (Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1," February 2003.) defines the RSASSA-PKCS1-v1_5 and RSASSA-PSS algorithms that are usually used to that purpose;

*MUST communicate the Signature Cryptographic 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;

*MUST associate a value to the "ASID" field (Authentication Scheme Identifier) of the EXT_AUTH header extension (Section 2.3 (Authentication Header Extension Format));

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 out of the scope of this document.

2.3. Authentication Header Extension Format

TOC

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 EXT_AUTH==1 header extension defined in [RFC5651] (Luby, M., Watson, M., and L. Vicisano, “Layered Coding Transport (LCT) Building Block,” October 2009.).
Several fields are added in addition to the HET (Header Extension Type) and HEL (Header Extension Length) fields (Figure 1 (Format of the Digital Signature EXT_AUTH header extension.)).

```
 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 |        Reserved        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                                                               ~
|                           Signature                           |
+                                               +-+-+-+-+-+-+-+-+
|                                               |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Format of the Digital Signature EXT_AUTH header extension.

The fields of the Digital Signature EXT_AUTH header extension are:
"ASID" (Authentication Scheme Identifier) field (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.

"Reserved" field (12 bits):

This is a reserved field that MUST be set to zero in this specification.

"Signature" field (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.

2.4. In Practice

TOC

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 MAY 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.

```
  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            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---
 |                                                               | ^ 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.

For instance Figure 2 (Example: Format of the Digital Signature EXT_AUTH header extension using 1024 bit signatures.) 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 byte long). The Digital Signature EXT_AUTH header extension is then 132 byte long.

---

3. Elliptic Curve Digital Signature Scheme

TOC

3.1. Principles

TOC

The computation of the ECC digital signature, using K_priv, 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.
Upon receiving this packet, the receiver recomputes the Group MAC, using K_pub, and compares it to the value carried in the packet. During this computation, the Weak Group MAC field MUST also be set to 0. If the check fails, the packet MUST be immediately dropped.
Several "Elliptic Curves" 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 size. Using a larger hash algorithm and then truncated the output is also feasible, however it consumes more processing power than is necessary. The following table lists the RECOMMENDED choices [RFC4754] (Fu, D. and J. Solinas, “IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA),” January 2007.) [RFC5480] (Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, “Elliptic Curve Cryptography Subject Public Key Information,” March 2009.).

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

The ECDSA-256, ECDSA-384 and ECDSA-521 are designed to offer security comparable with AES-128, AES-192 and AES-256 respectively [RFC4754] (Fu, D. and J. Solinas, “IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA),” January 2007.).

3.2. Parameters

TOC

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

- MUST communicate his 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, n_k, and the signature length, the two parameters being equal;
- MAY communicate a certificate (which also means that a PKI has been setup), for each receiver to be able to check the sender's public key;
- MUST communicate the Message Digest Algorithm;
- MUST communicate the Elliptic Curve;
- MUST associate a value to the "ASID" field (Authentication Scheme Identifier) of the EXT_AUTH header extension (Section 2.3 (Authentication Header Extension Format));

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 out of the scope of this document.

3.3. Authentication Header Extension Format

TOC

The integration of ECC 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 EXT_AUTH==1 header extension defined in [RFC5651] (Luby, M., Watson, M., and L. Vicisano, “Layered Coding Transport (LCT) Building Block,” October 2009.).
Several fields are added in addition to the HET (Header Extension Type) and HEL (Header Extension Length) fields (Figure 1 (Format of the Digital Signature EXT_AUTH header extension.)).

```
  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 |       Reserved        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 ~                                                               ~
 |                           Signature                           |
 +                                               +-+-+-+-+-+-+-+-+
 |                                               |    Padding    |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Format of the Digital Signature EXT_AUTH header extension.

The fields of the Digital Signature EXT_AUTH header extension are:
"ASID" (Authentication Scheme Identifier) field (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.

"Reserved" field (12 bits):

> This is a reserved field that MUST be set to zero in this specification.

"Signature" field (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.

3.4. In Practice

TOC

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 MAY 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.

```
 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---
|                                                               | ^ 3
+                                                               + | 2
.                                                               . |
.                      Signature (32 bytes)                     . | b
.                                                               . | y
|                                                               | | t
+                                                               + | e
|                                                               | v s
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---
```

Figure 4: Example: Format of the ECC Digital Signature EXT_AUTH header extension using ECDSA-256 signatures.

For instance Figure 4 (Example: Format of the ECC Digital Signature EXT_AUTH header extension using ECDSA-256 signatures.) 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 byte long.

4. Group Message Authentication Code (MAC) Scheme

TOC

4.1. Principles

TOC

The computation of the Group MAC, using K_g, 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 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 MAC Authentication header.
Upon receiving this packet, the receiver recomputes the Group MAC, using K_g, and compares it to the value carried in the packet. During this computation, the Group MAC field MUST also be set to 0. If the check fails, the packet MUST be immediately dropped.

[RFC2104] (Krawczyk, H., Bellare, M., and R. Canetti, “HMAC: Keyed-Hashing for Message Authentication,” February 1997.) 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 out of the scope of this document.

4.2. Parameters

TOC

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

* MUST communicate the Cryptographic MAC Function, for instance, HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, or HMAC-SHA-512. Because of security threats on SHA-1, the use of HMAC-SHA-256 is RECOMMENDED. As a side effect, the receivers also know the key length, n_k, and the non truncated MAC output length;
* 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 left-most bits (most significant bits) of the MAC output are kept. Of course, n_m MUST be lower or equal to n_k;
* MUST communicate the K_g group key to the receivers, confidentially, before starting the session. This key might have to be periodically refreshed for improved robustness;
* MUST associate a value to the "ASID" field (Authentication Scheme Identifier) of the EXT_AUTH header extension (Section 4.3 (Authentication Header Extension Format));

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 be often the case when periodic re-keying is required). The details of this out-of-band mechanism are out of the scope of this document.

4.3. Authentication Header Extension Format

TOC

The integration of Group MAC is similar in ALC and NORM and relies on the header extension mechanism defined in both protocols. More precisely this document details the EXT_AUTH==1 header extension

defined in [RFC5651] (Luby, M., Watson, M., and L. Vicisano, “Layered Coding Transport (LCT) Building Block,” October 2009.).
Several fields are added in addition to the HET (Header Extension Type) and HEL (Header Extension Length) fields (Figure 5 (Format of the Group MAC EXT_AUTH header extension.)).

```
 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 |        Reserved       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                                                               ~
|                           Group MAC                           |
+                                               +-+-+-+-+-+-+-+-+
|                                               |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Format of the Group MAC EXT_AUTH header extension.

The fields of the Group MAC EXT_AUTH header extension are:
"ASID" (Authentication Scheme Identifier) field (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.

"Reserved" field (12 bits):

> This is a reserved field that MUST be set to zero in this specification.

"Group MAC" field (variable size, multiple of 32 bits):

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

TOC

4.4. In Practice

Each packet sent MUST contain exactly one Group MAC EXT_AUTH header extension. A receiver MUST drop packets that do not contain a Group MAC EXT_AUTH header extension.
All receivers MUST recognize EXT_AUTH but MAY not be able to parse its content, for instance because they do not support Group MAC. In that case the Group MAC EXT_AUTH extension is ignored.

```
 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           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                      Group MAC (10 bytes)                     |
+                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |            Padding            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Example: Format of the Group MAC EXT_AUTH header extension using HMAC-SHA-1.

For instance Figure 6 (Example: Format of the Group MAC EXT_AUTH header extension using HMAC-SHA-1.) shows the Group MAC EXT_AUTH header extension when using HMAC-SHA-1. The Group MAC EXT_AUTH header extension is then 16 byte long.

---

5. Combined Use of the RSA/ECC Digital Signatures and Group MAC Schemes

TOC

5.1. Principles

TOC

In some situations, it can be interesting to use both authentication schemes. The goal of the Group MAC is to mitigate DoS attacks coming from attackers that are not group members [RFC4082] (Perrig, A., Song, D., Canetti, R., Tygar, J., and B. Briscoe, “Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication

Transform Introduction," June 2005.) by adding a light authentication scheme as a front-end.

More specifically, before sending a message, the sender sets the Signature field and Group MAC field to zero. Then the sender computes the Signature as detailed in Section 2.1 (Principles) or in Section 3.1 (Principles) and stores the value in the Signature field. Then the sender computes the Group MAC as detailed in Section 4.1 (Principles) and stores the value in the Group MAC field. The (RSA or ECC) digital signature value is therefore protected by the Group MAC, which avoids DoS attacks where the attacker corrupts the digital signature itself.

Upon receiving the packet, the receiver first checks the Group MAC, as detailed in Section 4.1 (Principles). If the check fails, the packet MUST be immediately dropped. Otherwise the receiver checks the Digital Signature, as detailed in Section 2.1 (Principles). If the check fails, the packet MUST be immediately dropped.

This scheme features a few limits:

* the Group MAC is of no help if a group member (who knows K_g) impersonates the sender and sends forged messages to other receivers. DoS attacks are still feasible;
* it requires an additional MAC computing for each packet, both at the sender and receiver sides;
* 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 [RFC3711] (Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)," March 2004.) section 9.5). In the current specification, n_m MUST be a multiple of 32 bits, and 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 be successfully forged is one in 2^32. Since the Group 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 MAC must be balanced against these limitations.

5.2. Parameters

TOC

Several parameters MUST be initialized by an out-of-band mechanism, as defined in Section 2.2 (Parameters), Section 3.2 (Parameters) and Section 4.2 (Parameters).

5.3. Authentication Header Extension Format

TOC

The integration of combined RSA/ECC Digital Signature and Group MAC is similar in ALC and NORM and relies on the header extension mechanism defined in both protocols. More precisely this document details the EXT_AUTH==1 header extension defined in [RFC5651] (Luby, M., Watson, M., and L. Vicisano, “Layered Coding Transport (LCT) Building Block,” October 2009.).
Several fields are added in addition to the HET (Header Extension Type) and HEL (Header Extension Length) fields (Figure 7 (Format of the Group MAC EXT_AUTH header extension.)).

```
 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 |       Reserved        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                                                               ~
|                           Signature                           |
+                                               +-+-+-+-+-+-+-+-+
|                                               |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Group MAC                           |
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Format of the Group MAC EXT_AUTH header extension.

The fields of the Group MAC EXT_AUTH header extension are:
"ASID" (Authentication Scheme Identifier) field (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.

"Reserved" field (12 bits):

> This is a reserved field that MUST be set to zero in this specification.

"Signature" field (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 MAC" field (variable size, multiple of 32 bits, by default 32 bits):

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

5.4. In Practice

TOC

Each packet sent MUST contain exactly one combined Digital Signature/ Group MAC EXT_AUTH header extension. A receiver MUST drop packets that do not contain a combined Digital Signature/Group MAC EXT_AUTH header extension.
All receivers MUST recognize EXT_AUTH but MAY not be able to parse its content, for instance because they do not support combined Digital Signature/Group MAC. In that case the combined Digital Signature/Group MAC EXT_AUTH extension is ignored.
It is RECOMMENDED that the n_m parameter of the group authentication scheme be small, and by default equal to 32 bits (Section 5.1 (Principles)).

```
  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 (=34)  |  ASID |           0           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---
 |                                                               | ^ 1
 +                                                               + | 2
 |                                                               | | 8
 .                                                               . |
 .                      Signature (128 bytes)                    . | b
 .                                                               . | y
 |                                                               | | t
 +                                                               + | e
 |                                                               | v s
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---
 |                       Group MAC (32 bits)                     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---
```

Figure 8: Example: Format of the combined RSA Digital Signature/Group MAC EXT_AUTH header extension using 1024 bit signatures.

For instance Figure 8 (Example: Format of the combined RSA Digital Signature/Group MAC EXT_AUTH header extension using 1024 bit signatures.) shows the combined Digital Signature/Group MAC EXT_AUTH header extension when using 128 byte (1024 bit) key RSA digital signatures (which also means that the signature field is 128 byte long). The EXT_AUTH header extension is then 136 byte long.

6. IANA Considerations

TOC

This document does not require any IANA registration.

7. Security Considerations

TOC

7.1. Dealing With DoS Attacks

TOC

Digital signatures introduces new opportunities for an attacker to mount DoS attacks. For instance an attacker can try to saturate the processing capabilities of the receiver (faked packets are easy to create but checking them requires to compute a costly digital signature).
In order to mitigate these attacks, it is RECOMMENDED to use the combined Digital Signature/Group MAC scheme (Section 5.1 (Principles)). However, no mitigation is possible if a group member acts as an attacker.

7.2. Dealing With Replay Attacks

TOC

Replay attacks consist for an attacker to store a valid message and to replay it later on.

TOC

7.2.1. Impacts of Replay Attacks on the Simple Authentication Schemes

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

7.2.2. Impacts of Replay Attacks on NORM

TOC

We review here 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 the congestion control protocols. First, let us consider replay attacks within a given NORM session. NORM defines a "sequence" field that can be used to protect against replay attacks [RMT-PI-NORM] (Adamson, B., Bormann, C., Handley, M., and J. Macker, “Negative-acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Protocol,” September 2009.) 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. It is RECOMMENDED that a receiver check this sequence field and drop messages considered as replayed. Similarly, it is RECOMMENDED that a sender check this sequence, for each known receiver, and drop messages considered as replayed. In both cases, checking this sequence field SHOULD be done before authenticating the packet: if the sequence field has not been corrupted, the replay attack will immediately be identified, and otherwise the packet will fail the authentication test. This analysis shows that NORM itself is robust in front of replay attacks within the same session.
Now let us consider replay attacks across several NORM sessions. A host participation 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 the "instance_id" be changed for each NORM instance. It is also RECOMMENDED, when the Group MAC authentication/integrity check scheme is used, that the shared group key, K_g, be changed across sessions. Therefore, NORM can be made robust in front of replay attacks across different sessions.

7.2.3. Impacts of Replay Attacks on ALC

TOC

We review here 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 the layered or wave based congestion control protocols.

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

- *Regular packets containing an authentication tag: a replayed message containing an encoding symbol will be detected once authenticated, thanks to the object/block/symbol identifiers, and will be silently discarded. This kind of replay attack is only penalizing in terms of memory and processing load, but does not compromise the ALC behavior.

- *Control packets containing an authentication tag: ALC control packets, by definition, do not include any encoding symbol and therefore do not include any object/block/symbol identifier that would enable a receiver to identify duplicates. However, a sender has a very limited number of reasons to send control packets. More precisely:

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

  - -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.

This analysis shows that ALC itself is robust in front of 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's IP address; Transport Session Identifier (TSI)} [RFC5651] (Luby, M., Watson, M., and L. Vicisano, “Layered Coding Transport (LCT) Building Block,” October 2009.). Therefore, when a given sender creates several sessions, it is RECOMMENDED that the TSI be changed for each ALC instance. It is also RECOMMENDED, when the Group MAC authentication/integrity check scheme is used, that the shared group key, K_g, be changed across sessions. Therefore, ALC can be made robust in front of replay attacks across different sessions.

---

8. Acknowledgments

TOC

The author is grateful to the authors of [RFC4359] (Weis, B., “The Use of RSA/SHA-1 Signatures within Encapsulating Security Payload (ESP) and Authentication Header (AH),” January 2006.), [RFC4754] (Fu, D. and J. Solinas, “IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA),” January 2007.) and [RFC5480] (Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, “Elliptic Curve Cryptography Subject Public Key Information,” March 2009.) that inspired several sections of the present document.

9. References

TOC

9.1. Normative References

TOC

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, BCP 14, March 1997.
[RFC5651]	Luby, M., Watson, M., and L. Vicisano, "Layered Coding Transport (LCT) Building Block," RFC 5651, October 2009.
[RMT-PI-ALC]	Luby, M., Watson, M., and L. Vicisano, "Asynchronous Layered Coding (ALC) Protocol Instantiation," draft-ietf-rmt-pi-alc-revised-09.txt (work in progress), October 2009.
[RMT-PI-NORM]	Adamson, B., Bormann, C., Handley, M., and J. Macker, "Negative-acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Protocol," draft-ietf-rmt-pi-norm-revised-14.txt (work in progress), September 2009.

9.2. Informative References

TOC

[MSEC-TESLA]	Roca, V., Francillon, A., and S. Faurite, "Use of TESLA in the ALC and NORM Protocols," draft-ietf-msec-tesla-for-alc-norm-09.txt (work in progress), October 2009.
[RFC2104]	Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," RFC 2104, February 1997 (TXT).
[RFC3447]	Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1," RFC 3447, February 2003 (TXT).
[RFC3711]	Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)," RFC 3711, March 2004 (TXT).
[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 (TXT).
[RFC4359]	Weis, B., "The Use of RSA/SHA-1 Signatures within Encapsulating Security Payload (ESP) and Authentication Header (AH)," RFC 4359, January 2006 (TXT).
[RFC4754]	

	Fu, D. and J. Solinas, “IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA),” RFC 4754, January 2007 (TXT).
[RFC5480]	Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, “Elliptic Curve Cryptography Subject Public Key Information,” RFC 5480, March 2009.
[RMT-FLUTE]	Paila, T., Walsh, R., Luby, M., Lehtonen, R., and V. Roca, “FLUTE - File Delivery over Unidirectional Transport,” draft-ietf-rmt-flute-revised-07 (work in progress), August 2009.

Author's Address

TOC

	Vincent Roca
	INRIA
	655, av. de l'Europe
	Zirst; Montbonnot
	ST ISMIER cedex 38334
	France
Email:	vincent.roca@inria.fr
URI:	http://planete.inrialpes.fr/people/roca/