RMT	V. Roca
Internet-Draft	INRIA
Intended status: Experimental	November 19, 2007
Expires: May 22, 2008	

TOC

Simple Authentication Schemes for the ALC and NORM Protocols
draft-roca-rmt-simple-auth-for-alc-norm-01.txt

Status of this Memo

Abstract

This document introduces two 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 a group Message Authentication Code (MAC). Because this scheme relies 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.

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 as well as its integrity. For instance, ALC [draft-ietf-rmt-pi-alc-revised] (Luby, M., Watson, M., and L. Vicisano, “Asynchronous Layered Coding (ALC) Protocol Instantiation,” November 2007.) and NORM [draft-ietf-rmt-pi-norm-revised] (Adamson, B., Bormann, C., Handley, M., and J. Macker, “Negative-acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Protocol,” March 2007.) 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. 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 some use case, an attacker might impersonate the ALC or NORM session's sender and inject forged packets to the receivers, thereby corrupting the objects reconstructed by the receivers.
In case of group communications, several solutions exist to provide the receiver some guarantees 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 this solution is limited by high computational costs and high transmission overheads.

- *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.): 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 scheme:

	Digital Signature	**Group MAC**	**TESLA**
True authentication and integrity service	Yes	No (group security)	Yes
Immediate authentication	Yes	Yes	No

Processing load	--	++	+
Transmission overhead	--	++	+
Protocol complexity	++	++	--

[draft-ietf-msec-tesla-for-alc-norm] (Adamson, B., Bormann, C., Handley, M., and J. Macker, “Use of TESLA in the ALC and NORM Protocols,” November 2007.) explains how to use TESLA in the context of ALC and NORM protocols. The current document specifies the use of the first two schemes, namely the Digital Signature and Group MAC schemes, in the ALC and NORM content delivery protocols. Since the FLUTE application [RFC3926] (Paila, T., Luby, M., Lehtonen, R., Roca, V., and R. Walsh, “FLUTE - File Delivery over Unidirectional Transport,” October 2004.) is built on top of ALC, it will directly benefit from the services offered by TESLA at the transport layer. 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).

1.1. 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.).

1.2. 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;

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 and public) key length, in bits. n_k is also the signature length, since both values must be equal with digital signatures;

Group MAC related notations and definitions:

*K_g is a shared group key, communicated to all group members, confidentially, before starting the session. The mechanism by which this group key is shared by the group members is out of the scope of this document;

*n_k is the 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. Digital Signature Scheme

TOC

2.1. Principles

TOC

The computation of the digital signature, using K_priv, 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 "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.
With RSASSA-PKCS1-v1_5 (default) and RSASSA-PSS signatures (Section 5 (IANA Considerations)), 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.

2.2. Parameters that Need to Be Initialized Out-of-Band

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 bootstrap (and direct time synchronization response messages when applicable). 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 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 initialization 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 in ALC or NORM is similar and relies on the header extension mechanism defined in both protocols. More precisely this document details the EXT_AUTH==1 header extension defined in [draft-ietf-rmt-bb-lct-revised] (Luby, M., Watson, M., and L. Vicisano, “Layered Coding Transport (LCT) Building Block,” November 2007.).

> ----- Editor's note: All authentication schemes using the EXT_AUTH header extension MUST reserve the same 4 bit "ASID" field after the HET/HEL fields. This way, several authentication schemes can be used in the same ALC or NORM session, even on the same communication path. -----

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. Use of Authentication Header Extensions

TOC

Each packet sent by the session's sender MUST contain exactly one Digital Signature EXT_AUTH header extension. A receiver MUST drop 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 these receivers MUST ignore the Digital Signature EXT_AUTH header extensions.

```
 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. Group MAC Scheme

TOC

3.1. 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_w 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.

3.2. Parameters that Need to Be Initialized Out-of-Band

TOC

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

* MUST communicate the cryptographic Message Authentication Code (MAC). For instance, HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, or HMAC-SHA-512. 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.

* MUST communicate the K_g group key to the receivers, confidentially, before starting the session. This key might have to be periodically refreshed.

* MUST associate a value to the "ASID" field (Authentication Scheme Identifier) of the EXT_AUTH header extension (Section 3.3 (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 initialization 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 Group MAC in ALC or NORM is similar and relies on the header extension mechanism defined in both protocols. More precisely this document details the EXT_AUTH==1 header extension defined in [draft-ietf-rmt-bb-lct-revised] (Luby, M., Watson, M., and L. Vicisano, “Layered Coding Transport (LCT) Building Block,” November 2007.).

> ----- Editor's note: All authentication schemes using the EXT_AUTH header extension MUST reserve the same 4 bit "ASID" field after the HET/HEL fields. This way, several authentication schemes can be used in the same ALC or NORM session, even on the same communication path. -----

Several fields are added in addition to the HET (Header Extension Type) and HEL (Header Extension Length) fields (Figure 3 (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 3: 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 Group MAC of the message. If need be, this field is padded (with 0) up to a multiple of 32 bits.

3.4. Use of Authentication Header Extensions

TOC

Each packet sent by the session's sender 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 these receivers MUST ignore the Group MAC EXT_AUTH extensions.

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

For instance Figure 4 (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.

TOC

4. Combined Use of the Digital Signatures and Group MAC Schemes

4.1. 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 member [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 computes the Group MAC MAC(K_g, M), which includes the ALC or NORM header (with the various header extensions), plus the payload when applicable. During this computation, the Weak Group MAC field MUST be set to 0. However the digital signature MUST have been calculated and is included in the Group MAC calculation itself. Then the sender truncates the MAC output to keep the n_w most significant bits and stores the result in the Group MAC authentication header. Upon receiving this packet, the receiver recomputes the Group MAC and compares it to the value carried in the packet. If the check fails, the packet MUST be immediately dropped.

This scheme features a few limits:

- it is of no help if a group member (who knows K_g) impersonates the sender and sends forged messages to other receivers;
- 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 (e.g. 32 bits) (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). As a side effect, the authentication service is significantly weakened (the probability that any packet be successfully forged is one in 2^32). Since the Group MAC check is only a pre-check that will be 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.

4.2. Combined Use of both Authentication Header Extensions

TOC

In order to use both authentication schemes, the packet sender calculates and includes two EXT_AUTH header extensions, in any order, one for each authentication scheme. It is RECOMMENDED that the n_m parameter of the group authentication scheme be small, for instance equal to 32 bits (Section 4.1 (Principles)).
When it is decided that both schemes should be combined, then all the packets MUST include both header extensions. A receiver receiving a packet with only one of the two schemes MUST reject it. This requirement is meant to prevent DoS attacks where the attacker would inject forged packets containing only the Digital Signature EXT_AUTH header extension, to force the receiver to check it.

---

5. IANA Considerations

TOC

This document does not require any IANA registration.

---

6. Security Considerations

TOC

TBD

---

7. Acknowledgments

TOC

TBD

---

8. References

TOC

8.1. Normative References

TOC

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, BCP 14, March 1997.
[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).
[draft-ietf-msec-tesla-for-alc-norm]	Adamson, B., Bormann, C., Handley, M., and J. Macker, "Use of TESLA in the ALC and NORM Protocols," draft-ietf-msec-tesla-for-alc-norm-03.txt (work in progress), November 2007.
[draft-ietf-rmt-bb-lct-revised]	Luby, M., Watson, M., and L. Vicisano, "Layered Coding Transport (LCT) Building Block," draft-ietf-rmt-bb-lct-revised-06.txt (work in progress), November 2007.
[draft-ietf-rmt-pi-alc-revised]	Luby, M., Watson, M., and L. Vicisano, "Asynchronous Layered Coding (ALC) Protocol Instantiation," draft-ietf-rmt-pi-alc-revised-05.txt (work in progress), November 2007.
[draft-ietf-rmt-pi-norm-revised]	Adamson, B., Bormann, C., Handley, M., and J. Macker, "Negative-acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Protocol," draft-ietf-rmt-pi-norm-revised-05.txt (work in progress), March 2007.

8.2. Informative References

TOC

[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).
[RFC3926]	Paila, T., Luby, M., Lehtonen, R., Roca, V., and R. Walsh, "FLUTE - File Delivery over Unidirectional Transport," RFC 3926, October 2004 (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).

Author's Address

TOC

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

Full Copyright Statement

TOC