Network Working Group
Request for Comments: 4442
Category: Standards Track

S. Fries
H. Tschofenig
Siemens

March 2006

Bootstrapping
Timed Efficient Stream Loss-Tolerant Authentication (TESLA)

Status of This Memo

Copyright Notice

Abstract

TESLA, the Timed Efficient Stream Loss-tolerant Authentication protocol, provides source authentication in multicast scenarios. TESLA is an efficient protocol with low communication and computation overhead that scales to large numbers of receivers and also tolerates packet loss. TESLA is based on loose time synchronization between the sender and the receivers. Source authentication is realized in TESLA by using Message Authentication Code (MAC) chaining. The use of TESLA within the Secure Real-time Transport Protocol (SRTP) has been published, targeting multicast authentication in scenarios where SRTP is applied to protect the multimedia data. This solution assumes that TESLA parameters are made available by out-of-band mechanisms.

This document specifies payloads for the Multimedia Internet Keying (MIKEY) protocol for bootstrapping TESLA for source authentication of secure group communications using SRTP. TESLA may be bootstrapped using one of the MIKEY key management approaches, e.g., by using a digitally signed MIKEY message sent via unicast, multicast, or broadcast.

Table of Contents

1. Introduction

In many multicast, broadcast, and unicast communication scenarios, it is necessary to guarantee that a received message has been sent from a dedicated source and has not been altered in transit. In unicast communication, commonly a pairwise security association exists that enables the validation of message integrity and data origin. The approach in group-based communication is different, as here a key is normally shared between the members of a group and thus may not be used for data origin authentication. As in some applications a dedicated identification of a sender is required, there exists the requirement to support data origin authentication also in multicast scenarios. One of the methods supporting this is TESLA [RFC4082]. TESLA provides source authentication in multicast scenarios by using MAC chaining. It is based on loose time synchronization between the sender and the receivers.

[RFC4383] describes extensions for SRTP [RFC3711] in order to support TESLA [RFC4082] for source authentication in multicast scenarios. SRTP needs dedicated cryptographic context describing the security parameter and security policy per multimedia session to be protected. This cryptographic context needs to be enhanced with a set of TESLA parameters. It is necessary to provide these parameters before the actual multicast session starts. [RFC4383] does not address the bootstrapping for these parameters.

This document details bootstrapping of TESLA parameters in terms of parameter distribution for TESLA policy as well as the initial key, using the Multimedia Internet Keying (MIKEY) [RFC3830] protocol. MIKEY defines an authentication and key management framework that can be used for real-time applications (both for peer-to-peer communication and group communication). In particular, [RFC3830] is defined in a way that is intended to support SRTP in the first place but is open to enhancements to be used for other purposes too. Following the description in [RFC3830], MIKEY is targeted for point-to-point as well as group communication. In the context of group communication, an administrator entity can distribute session keys to the associated entities participating in the communication session. This scenario is also applicable for TESLA where one entity may provide information to many others in a way that the integrity of the communicated information can be assured. The combination of MIKEY and TESLA supports this group-based approach by utilizing the MIKEY framework to distribute TESLA parameter information to all involved entities. Note that this document focuses only on the distribution of the parameters, not on the generation of those parameters.

MIKEY [RFC3830] itself describes three authentication and key exchange protocols (symmetric key encryption, public key encryption, and signed Diffie-Hellman). Extensions to the MIKEY key exchange methods have been defined. A fourth key distribution method is provided by [DHHMAC] and describes a symmetrically protected Diffie-Hellman key agreement. A further option has been proposed in [RSA-R] that describes an enhanced asymmetric exchange variant, also supporting inband certificate exchange. All the different key management schemes mentioned above may be used to provide the TESLA parameters. The required TESLA parameters to be exchanged are already described in [RFC4383], while this document describes their transport within MIKEY.

The following security requirements have to be placed on the exchange of TESLA parameters:

- Authentication and Integrity MUST be provided when sending the TESLA parameters, especially for the initial key.
- Confidentiality MAY be provided for the TESLA parameters.

These security requirements apply to the TESLA bootstrapping procedure only. Security requirements for applications using TESLA are beyond the scope of this document. Security aspects that relate to TESLA itself are described in [RFC4082], and security issues for TESLA usage for SRTP are covered in [RFC4383].

It is important to note that this document is one piece of a complete solution. Assuming that media traffic is to be secured using TESLA as described in [RFC4383], then (a) keying material and (b) parameters for TESLA are required. This document contributes the parameters and the authentication methods used in MIKEY to provide the keying material. The parameter exchange for TESLA also needs to be secured against tampering. This protection is also provided by MIKEY.

2. Terminology

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 RFC 2119 [RFC2119].

3. TESLA Parameter Overview

According to [RFC4383], a number of transform-dependent parameters need to be provided for proper TESLA operation. The complete list of parameters can be found in Section 4.3 of [RFC4383]. Note that parameter 10 of [RFC4383], describing the lag of the receiver clock relative to the sender clock, is omitted in this document since it can be computed.

MIKEY already requires synchronized clocks, which also provides for synchronization for TESLA. Moreover, Section 4.3 states an option to use MIKEY for clock drift determination between the sender and receiver. Thus, this parameter does not need to be transmitted in MIKEY directly.

The information in brackets provides the default values as specified in Section 6.2 of [RFC4383].

1. An identifier for the PRF (TESLA PRF), implementing the one-way function F(x) in TESLA (to derive the keys in the chain), and the one-way function F'(x) in TESLA (to derive the keys for the TESLA MAC, from the keys in the chain), e.g., to indicate the keyed hash function (default HMAC-SHA1).

2. A non-negative integer, determining the length of the F output, i.e., the length of the keys in the chain, which is also the key disclosed in an SRTP packet if TESLA is used in the SRTP context (default 160 bit).

3. A non-negative integer, determining the length of the output of F', i.e., the length of the key for the TESLA MAC (default 160 bit).

4. An identifier for the TESLA MAC that accepts the output of F'(x) as its key, e.g., to indicate a keyed hashing function (default HMAC-SHA1).

5. A non-negative integer, determining the length of the output of the TESLA MAC (default 80 bit).

6. The beginning of the session for which a key will be applied.

7. The interval duration (in milliseconds) for which a dedicated key will be used.

8. The key disclosure delay (in number of intervals) characterizes the period after which the key will be sent to the involved entities (e.g., as part of SRTP packets).

9. Non-negative integer, determining the length of the key chain, which is determined based on the expected duration of the stream.

10. The initial key of the chain to which the sender has committed himself.

4. Parameter Encoding within MIKEY

As mentioned in Section 3, TESLA parameters need to be transported before actually starting a session. MIKEY currently only defines a payload for transporting the SRTP policy (see Section 6.10 of [RFC3830]). This section describes the enhancement of MIKEY to allow the transport of a TESLA policy and additionally the initial TESLA key.

4.1. Security Policy (SP) Payload

The Security Policy payload defines a set of policies that apply to a specific security protocol. The definition here relies on the security policy payload definition in [RFC3830].

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next payload  ! Policy no     ! Prot type     ! Policy param  ˜
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
˜ length (cont) ! Policy param                                  ˜
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Next payload (8 bits):
  Identifies the payload that is added after
  this payload. See Section 6.1 of [RFC3830] for
  more details.

- Policy no (8 bits):
  Each security policy payload must be given a
  distinct number for the current MIKEY session by the
  local peer. This number is used to map a cryptographic session
  to a specific policy (see also Section 6.1.1 of [RFC3830]).

- Prot type (8 bits):
  This value defines the security protocol.
  A second value needs to be defined as shown below:
  (MIKEY already defines the value 0.)

Prot type	Value
SRTP	0
TESLA	1

- Policy param length (16 bits):
  This field defines the total length of the
  policy parameters for the selected security protocol.

- Policy param (variable length):
  This field defines the policy for the specific security protocol.

The Policy param part is built up by a set of Type/Length/Value (TLV) payloads. For each security protocol, a set of possible type/value pairs can be negotiated as defined.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Type          ! Length        ! Value                         ˜
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Type (8 bits):
  Specifies the type of the parameter.

- Length (8 bits):
  Specifies the length of the Value field (in bytes).

- Value (variable length):
  Specifies the value of the parameter.

4.2. TESLA Policy

This policy specifies the parameters for TESLA. The types/values that can be negotiated are defined by the following table. The concrete default values are taken from [RFC4383], but other values may also be used:

Type	Meaning	Possible values
1	PRF identifier for f and f’, realising F(x) and F’(x)	see below
3	Length of PRF f’ output	160
4	Identifier for the TESLA MAC	see below
5	Length of TESLA MAC output	80 (truncated)
6	Start of session	in bytes
7	Interval duration (in msec)	in bytes
8	Key disclosure delay	in bytes
9	Key chain length (number of intervals)	in bytes
10	local timestamp media receiver	see below

The time values stated in items 6 and 10 SHALL be transported in NTP-UTC format, which is one of the three options described in Section 6.6 of [RFC3830]. A four-byte integer value for policy item 7 and a two-byte integer value for policy item 8 are RECOMMENDED, carrying interval duration and key disclosure delay. Note that policy type 10 does NOT correspond to TESLA parameter 10 stated in Section 3 and discussed in Section 4.4. Moreover, policy type 10 stated above is optional and SHOULD be used if the time synchronization described in Section 4.3, point two, is used. Otherwise, it SHOULD be omitted.

For the PRF realizing F(x) and F'(x), a one-byte length is sufficient. The currently defined possible values are:

TESLA PRF F(x), F'(x)	Value
HMAC-SHA1	0

For the TESLA MAC, a one-byte length is enough. The currently defined possible values are:

TESLA MAC	Value
HMAC-SHA1	0

4.3. Time Synchronization

MIKEY as well as TESLA require the time synchronization of the communicating peers. MIKEY requires time synchronization to provide timestamp-based replay protection for the one-roundtrip authentication and key exchange protocols. TESLA, on the other hand, needs this information to determine the clock drift between the senders and the receivers in order to release the disclosed key appropriately. Two alternatives are available for time synchronization:

1. Usage of out-of-band synchronization using NTP [RFC1305]. This approach is already recommended within [RFC3830]. The advantage of this approach is the option to use the MIKEY key management variants that perform within a half-roundtrip. The disadvantage is the required time synchronization via an additional protocol.

2. [RFC4082] also sketches a possible inband synchronization in Section 3.3.1. This approach is summarized here in the context of MIKEY. Note that here the actual TESLA policy payload is transmitted as part of the MIKEY responder message.

   * The data receiver, which would be the MIKEY initiator, sets the local time parameter t_r and sends it as part of the timestamp payload as described in [RFC3830]. This value t_r needs to be stored locally.

* Upon receipt of the MIKEY initiator message, the data sender replies with the MIKEY responder message, setting the local time stamp at data receiver (parameter 11) to the value t_r received in the MIKEY initiator message, and sets his local time as a 64-bit UTC value t_s in the timestamp payload as described in [RFC3830].

```
MIKEY initiator message
[MIKEY parameter incl. local timestamp (t_r)]
------------------>

MIKEY responder message
[MIKEY parameter incl. local timestamp (t_s), TESLA policy
 payload, received local time stamp t_r]
<------------------
```

* Upon receiving the MIKEY responder message the data receiver sets D_t = t_s - t_r + S, where S is an estimated bound on the clock drift throughout the duration of the session.

This approach has the advantage that it does not require an additional time synchronization protocol. The disadvantage is the necessity to perform a full MIKEY handshake, to enable correct parameter transport. Moreover this approach is direction dependent, as it may only be applied if the media receiver is also the MIKEY initiator.

Out-of-band synchronization using NTP (i.e., alternative 1) is the RECOMMENDED approach for clock synchronization. In scenarios where the media receiver is also the MIKEY initiator piggybacking timestamp information in MIKEY (i.e., alternative 2) MAY be used to allow for inband determination of the clock drift between sender and receiver.

4.4. Key Data Transport within MIKEY’s General Extension Payload

The General Extensions Payload was defined to allow possible extensions to MIKEY without the need for defining a completely new payload each time. This payload can be used in any MIKEY message and is part of the authenticated/signed data part.

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 ! Next payload  ! Type          ! Length                        !
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 ! Data                                                          ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- * Next payload (8 bits):
  Identifies the payload following this payload.

- * Type (8 bits):
  Identifies the type of general payload.
  MIKEY already defines the values 0 and 1.
  This document introduces a new value (2).

Type	Value	Comments
Vendor ID	0	Vendor specific byte string
SDP IDs	1	List of SDP key mgmt IDs
TESLA I-Key	2	TESLA initial key

- * Length (16 bits):
  The length in bytes of the Data field.

- * Data (variable length):
  The general payload data.

5. Security Considerations

The security properties of multi-media data in a multicast environment depends on a number of building blocks.

SRTP-TESLA [RFC4383] describes extensions for SRTP [RFC3711] in order to support TESLA [RFC4082] for source authentication in multicast scenarios. As such, security considerations described with TESLA (see [PCST] and [RFC4082]), the TESLA SRTP mapping [RFC4383], and SRTP [RFC3711] itself are relevant in this context.

Furthermore, since this document details bootstrapping of TESLA using the Multimedia Internet Keying (MIKEY) [RFC3830] protocol, the security considerations of MIKEY are applicable to this document.

As a summary, in order for a multi-media application to support TESLA, the following protocol interactions (in relationship to this document) are necessary:

- MIKEY [RFC3830] is executed between the desired entities to perform authentication and a secure distribution of keying material. In order to subsequently use TESLA, the parameters described in this document are distributed using MIKEY. MIKEY itself uses another protocol for parameter transport, namely, the Session Description Protocol (SDP) [RFC2327]. SDP might again be used within Session Initiation Protocol (SIP, [RFC3261]) to set up a session between the desired entities.

- After the algorithms, parameters, and session keys are available at the respective communication entities, data traffic protection via SRTP-TESLA [RFC4383] can be used. SRTP-TESLA itself applies TESLA to the SRTP protocol, and as such the processing guidelines of TESLA need to be followed.

5.1. Man-in-the-Middle Attack

Threat:

The exchange of security-related parameters and algorithms without mutual authentication of the two peers can allow an adversary to perform a man-in-the-middle attack. The mechanisms described in this document do not themselves provide such an authentication and integrity protection.

Countermeasures:

Throughout the document, it is assumed that the parameter exchange is secured using another protocol, i.e., the exchange parameters and algorithms are part of a authentication and key exchange protocol (namely, MIKEY). Source authentication of group and multicast communication cannot be provided for the data traffic if the prior signaling exchange did not provide facilities to authenticate the source. Using an authentication protocol that does not provide session keys as part of a successful protocol exchange will make it impossible to derive the necessary parameters required by TESLA. MIKEY provides session key establishment. Additionally, the exchange of parameters and algorithms MUST be authenticated and integrity protected. The security protection of the parameter exchange needs to provide the same level or a higher level of security.

5.2. Downgrading Attack

Threat:

The exchange of security-related parameters and algorithms is always subject to downgrading whereby an adversary modifies some (or all) of the provided parameters. For example, a few parameters require that a supported hash algorithm be listed. To mount an attack, the adversary has to modify the list of provided algorithms and to select the weakest one.

Countermeasures:

TESLA parameter bootstrapping MUST be integrity protected to prevent modification of the parameters and their values. Moreover, since unmodified parameters from an unknown source are not useful, authentication MUST be provided. This functionality is not provided by mechanisms described in this document. Instead, the capabilities of the underlying authentication and key exchange protocol (MIKEY) are reused for this purpose.

5.3. Denial of Service Attack

Threat:

An adversary might want to modify parameters exchanged between the communicating entities in order to establish different state information at the respective communication entities. For example, an adversary might want to modify the key disclosure delay or the interval duration in order to disrupt the communication at a later state since the TESLA algorithm assumes that the participating communication entities know the same parameter set.

Countermeasures:

The exchanged parameters and the parameters and algorithms MUST be integrity protected to allow the recipient to detect whether an adversary attempted to modify the exchanged information. Authentication and key exchange algorithms provided by MIKEY offer this protection.

5.4. Replay Attack

Threat:

An adversary who is able to eavesdrop on one or multiple protocol exchanges (MIKEY exchanges with the parameters described in this document) might be able to replay the payloads in a later protocol exchange. If the recipients accept the parameters and algorithms (or even the messages that carry these payloads), then a denial of service, downgrading, or a man-in-the-middle attack might be the consequence (depending on the entire set of replayed attributes and messages).

Countermeasures:

In order to prevent replay attacks, a freshness guarantee MUST be provided. As such, the TESLA bootstrapping message exchange MUST be unique and fresh, and the corresponding authentication and key exchange protocol MUST provide the same properties. In fact, it is essential to derive a unique and fresh session key as part of the authentication and key exchange protocol run that MUST be bound to the protocol session. This includes the exchanged parameters.

5.5. Traffic Analysis

Threat:

An adversary might be able to learn parameters and algorithms if he is located along the signaling path. This information can then later be used to mount attacks against the end-to-end multimedia communication. In some high-security and military environments, it might even be desirable not to reveal information about the used parameters to make it more difficult to launch an attack.

Countermeasures:

Confidentiality protection can be provided by a subset of the available MIKEY authentication and key exchange protocols, namely, those providing public key encryption and symmetric key encryption. The initial hash key, which is also one of the TESLA bootstrapping parameters, does not require confidentiality protection due to the properties of a hash chain.

6. IANA Considerations

This document requires an IANA registration for the following attributes. The registries are provided by MIKEY [RFC3830].

Prot Type:

This attribute specifies the protocol type for the security protocol as described in Section 4.1.

Type:

Identifies the type of the general payload. The General Extensions Payload was defined to allow possible extensions to MIKEY without the need for defining a completely new payload each time. Section 4.4 describes this attribute in more detail.

Following the policies outlined in [RFC3830], the values in the range up to 240 (including 240) for the above attributes are assigned after expert review by the MSEC working group or its designated successor. The values in the range from 241 to 255 are reserved for private use.

The IANA has added the following attributes and their respective values to an existing registry created in [RFC3830]:

Prot Type:

Prot Type	Value	Description
TESLA	1	TESLA as a security protocol

The value of 1 for the ’Prot Type’ must be added to the ’Prot type’ registry created by [RFC3830].

Type:

Type	Value	Description
TESLA I-Key	2	TESLA initial key

The value of 2 for the ’Type’ must be added to the ’Type’ registry created by [RFC3830]. The values of 0 and 1 are already registered in [RFC3830].

Also, the IANA has created two new registries:

TESLA-PRF: Pseudo-random Function (PRF) used in the TESLA policy:

This attribute specifies values for pseudo-random functions used in the TESLA policy (see Section 4.2).

TESLA-MAC: MAC Function used in TESLA:

This attribute specifies values for pseudo-random functions used in the TESLA policy (see Section 4.2).

Following the policies outlined in [RFC2434], the values for the TESLA-PRF and the TESLA-MAC registry in the range up to 240 (including 240) for the above attributes are assigned after expert review by the MSEC working group or its designated successor. The values in the range from 241 to 255 are reserved for private use.

IANA has added the following values to the TESLA-PRF and the TESLA-MAC registry:

TESLA-PRF:

PRF Function	Value
HMAC-SHA1	0

TESLA-MAC:

MAC Function	Value
HMAC-SHA1	0

7. Acknowledgements

The authors would like to thank Mark Baugher and Ran Canetti for the discussions in context of time synchronization. Additionally, we would like to thank Lakshminath Dondeti, Russ Housley, and Allison Mankin for their document reviews and for their guidance.

8. References

8.1. Normative References

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

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830, August 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.

[RFC4383] Baugher, M. and E. Carrara, "The Use of Timed Efficient Stream Loss-Tolerant Authentication (TESLA) in the Secure Real-time Transport Protocol (SRTP)", RFC 4383, February 2006.

8.2. Informative References

[DHHMAC] Euchner, M., "HMAC-authenticated Diffie-Hellman for MIKEY", Work in Progress, April 2005.

[PCST] Perrig, A., Canetti, R., Song, D., and D. Tygar, "Efficient and Secure Source Authentication for Multicast", in Proc. of Network and Distributed System Security Symposium NDSS 2001, pp. 35-46, 2001.

[RFC1305] Mills, D., "Network Time Protocol (Version 3) Specification, Implementation", RFC 1305, March 1992.

[RFC2327] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

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

[RSA-R] Ignjatic, D., "An additional mode of key distribution in MIKEY: MIKEY-RSA-R", Work in Progress, February 2006.

Authors’ Addresses

Steffen Fries
Siemens
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany

EMail: steffen.fries@siemens.com

Hannes Tschofenig
Siemens
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany

EMail: Hannes.Tschofenig@siemens.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).