Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 15, 2011

R. Blom
Y. Cheng
F. Lindholm
J. Mattsson
M. Naslund
K. Norrman
Ericsson

March 14, 2011

The Use of the Secure Real-time Transport Protocol (SRTP) in Store-and-Forward Applications
draft-naslund-srtp-saf-04

Abstract

This memo describes the use of so called store-and-forward cryptographic transforms within the Secure Real-time Transport Protocol (SRTP). The motivation is to support use cases when two end-points communicate via one (or more) store-and-forward middleboxes that are not fully trusted to access the media content. One of the main aspects of the transform is to make the confidentiality and message authentication independent of the RTP header. Another central aspect is to enable identification of the cryptographic context (keys etc.). Besides the security of the end-points, also trust assumptions regarding the store-and-forward middleboxes are addressed.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 15, 2011.

Copyright Notice

Table of Contents

1. Introduction

The Secure Real-time Transport Protocol (SRTP) [RFC3711] is a profile of RTP, which can provide confidentiality, message authentication, and replay protection to the RTP traffic and to the RTP Control Protocol (RTCP). The basic SRTP profile in [RFC3711] solves real-time end-to-end use cases, and does not consider use cases requiring Store-and-Forward (SaF) middleboxes. Such use cases are characterized by the need for a sender to deliver media to a receiver via a SaF middlebox. A SaF middlebox temporarily stores media and retransmits it to the intended receiver. Retransmission can be almost immediate (e.g. a push-to-talk group server), or be done at a much later time (e.g. a VoIP answering machine). The SaF middlebox is typically considered as semi-trusted, meaning that a SaF middlebox will store and deliver media as requested, but it cannot be excluded that a SaF middlebox will also try to extract the information (e.g. infringement of copyrighted content, legal or illegal intercept). The reason to not use a fully trusted middlebox is mainly cost and convenience, the same forces that drives out-sourcing and cloud computing. The trust model will be made more formal later in this document. What causes problems for standard end-to-end SRTP in these settings is its dependence on the actual RTP transport parameters which will differ when RTP is used on different hops, i.e., sender-middlebox and middlebox-receiver.

SRTP is a framework that allows new security functions and new transforms to be added and this document defines a so called store-and-forward extension to SRTP to meet the additional use cases considered. One of the main aspects of the transform is to make the confidentiality and message authentication independent of the RTP header. This allows for end-to-end protection to be achieved also when SaF middleboxes assign values to the RTP headers, independently on each hop.

Another aspect is that identification of the cryptographic context (keys etc.) between the end-points must be extended, as the parameters used in [RFC3711] are available only during transport of RTP packets over a "hop". For instance, [RFC3711] specifies that the receiver’s IP address shall be part of the context identifier, but this value may of course not be known to the sender when communicating messages via a SaF middlebox. Indeed, the receiver may not even be on-line at the time when the source initiates the communication. Another part of the cryptographic context identifier is the SSRC, which may be modified by SaF middleboxes.

While there certainly are differences between this document and [RFC3711] on mechanism level, it is worth noticing that the kind of extensions defined herein are conceptually almost identical to the

SRTP extensions previously defined in [RFC4383], which adds source origin authentication support to SRTP. Moreover, as far as the cryptographic processing is concerned, the SaF middleboxes may use [RFC3711] compliant processing and changes in cryptographic processing are thus only needed in the end-points.

1.1. Scope of this Document

The scope of this document is to specify extensions to SRTP (parameters, processing, and cryptographic transforms) to support the store-and-forward use case and its associated trust model. The SaF use case and trust models is defined in Section 3. No claims are made about supporting also other use cases, though of course, all the original uses cases from [RFC3711] can also be supported.

The SaF use case implies a different trust model than that originally considered when designing SRTP. This manifests itself in terms of the need to ensure authorized access to the different cryptographic keys involved, i.e. the extensions defined herein MUST have support from some key management scheme. Similar to the original SRTP specification, the actual definition of the key management solution is out of scope of this document. Requirements on key management can be found in Sections 3.3 and 5.5.

1.2. Conventions used in this Document

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].

Throughout the specification all protocol data fields are assumed to be byte aligned, i.e. all defined bit-sizes SHALL be multiples of 8.

1.2.1. Notation and Definitions

DoS: Denial of service

e2e: end-to-end

hbh: hop-by-hop

SaF: Store-and-Forward

For the purpose of this document we use the following definitions:

A is said to trust B with information I, if A is willing to share I with B. In the sequel we will simply say that A trusts B.

A is said to have sender-semi-trust in B if A considers B to be "honest-but-curious" in the following sense. A trust B to maintain information I provided by A, and (later) redistribute it to the intended recipients as specified by A (parties that A trust with I). However, A does not trust that B will not also try to extract the information I for him/herself and/or to attempt to distribute I also to other parties, e.g. parties that A does not trust with I.

A is similarly said to have receiver-semi-trust in B, if A trusts B to maintain information intended for A and to (later) distribute this information to A if and only if A so requests. However, A does not trust that B will not also attempt to distribute the information to other parties and/or try to extract it him/herself.

When it is obvious from the context (or irrelevant) we shall omit the directivity (sender/receiver) and simply say that A semi-trusts B.

2. SRTP

The Secure Real-time Transport Protocol (SRTP) [RFC3711] is a profile of RTP, which can provide confidentiality, message authentication, and replay protection to the RTP traffic and to the RTP Control Protocol (RTCP). Note that the term "SRTP" may often be used to indicate SRTCP as well. SRTP is a framework that allows new security functions and new transforms to be added. In the sequel, we assume that the reader is familiar with the SRTP specification [RFC3711], its packet structure, and its processing rules.

This specification defines a so called Store-and-Forward extension to SRTP to permit communication via semi-trusted SaF middleboxes. As mentioned, the SRTP extensions defined herein are very similar in nature to the SRTP extensions previously defined in [RFC4383] to add source origin authentication support to SRTP. In both cases, the extensions needed are: definition of new cryptographic transforms, a new packet format including additional in-band context signaling, and extensions to the SRTP cryptographic context concept.

3. The Store-and-Forward Use Cases

3.1. Problem Statement

We consider RTP communication solutions that include semi-trusted SaF middleboxes, i.e. middleboxes that should not have access to cleartext media, but still should be able to have access to other data in order to retransmit media according to RTP standard procedures. Below, we provide some use cases where S, M, and R refer

to Sender, SaF Middlebox, and Receiver. For each use case, we comment on aspects of the trust-model defined above.

Streaming Pre-encrypted Media: A content creator (S) distributes high value, encrypted content to clients (R). Distribution is made via a streaming server (M). From the content creator's point of view it is important that decrypted (plaintext) content is only made available to authorized clients. This means that S should be able to use a streaming server M, to which it assigns a bare minimum of sender-semi-trust. The clients may typically have some basic privacy requirement related to what type of content they access, but may otherwise be less concerned with whom else that also gets access to the content.

Recording Encrypted Media: Encrypted IPTV is broadcasted in a network. Only clients trusted by the content creator (S) should have access. Before having acquired a license to view the content, a user (R) records media on a Hard Disk Drive (M), where the media is stored in encrypted format, awaiting a license for rendering. Here, the trust in the HDD (M) by S and R is probably very asymmetric since the end-user most likely has a very strong trust in his personal home equipment. An additional requirement is the possibility for M to authenticate the data source S in order not to exhaust storage capacity with garbage.

Answering Machine: Operators commonly provide an answering machine service to their customers. Communicating parties (S and R) may not wish to disclose the media to any other party. Thus, the answering machine (M) acts as a SaF middlebox, which has to store encrypted data and retransmit it to the callee. In this use case, sender and receiver-semi-trust in M is likely to be fairly symmetric. In this case, it may be convenient to be able to splice several e2e protected streams (i.e. messages left by several different senders) into the same hbh communication session between M and R.

Further examples and more details can be found in Appendix A.

The typical use case is thus to require that media is (at least) confidentiality protected end-to-end (e2e) between the sender and the receiver. At the same time the communication should be protected hop-by-hop (hbh) to prevent malicious users from performing denial of service attacks by sending bogus data to SaF middleboxes, which the SaF middleboxes then would store, eventually exhausting their storage space and/or corrupting the data stored.

3.2. Trust Model and Security Requirements

The following figure shows the assumed trust model in terms of previous definitions.

In practice, the model means that

- S trusts R,
- S semi-trusts M to deliver information to R, and,
- R semi-trusts M to forward any information intended for R.

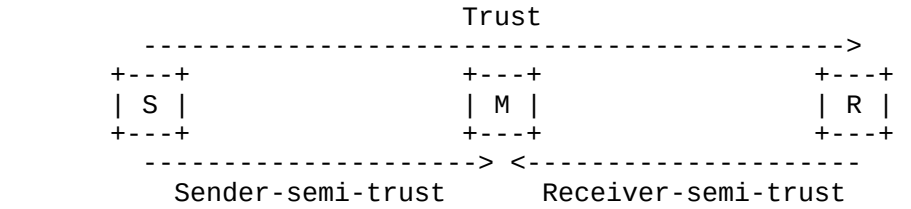

Figure 1: Trust Model (Sender, SaF Middlebox, Receiver)

As noted in the use cases above, S may be more concerned with who gets access to the information than R is. Still, this trust model, assuming a bare minimum of sender- and receiver-semi-trust as defined above, has been chosen since it is a simple trust model and seems to apply (qualitatively) as a common denominator for all the SaF use cases. Note also that the trust between S and R may often be mutual, but we do not require this.

M does not need to trust either of S or R. However, depending on the application, the security requirements, and the processing/storage capacity, for some SaF middleboxes it will be REQUIRED to be able to authenticate the source before accepting to store/forward data. If this was not the case, some anonymous malicious party might exhaust the storage resources of M. Similarly, a more robust implementation of M should have means to authenticate also R in order to avoid wasting resources, responding to spoofed requests. That is, the trust model also assumes the existence of other parties (not shown) that are not trusted by any of S, M, or R, and which may attempt to intervene with the communication between them and the SaF services provided by M.

When there are several SaF middleboxes in the path between S and R, it is necessary to assume that the SaF middleboxes semi-trust each other, at least in a transitive sense. Also, we may then have a

situation where S and R does not (directly) semi-trust a common M.

Some practical use cases when this trust model is likely to apply are given in Appendix A.

The security requirements for SRTP SaF hence are:

1. It SHALL be possible to provide e2e confidentiality and message authentication between S and R.

2. It SHALL be possible to provide hbh message authentication between S and M, between M and R, and between two middleboxes M, M'.

To provide a basis for enhanced privacy protection against other parties (e.g. traffic analysis), hbh confidentiality SHOULD also be provided.

As mentioned, hbh cryptographic processing is compatible with SRTP [RFC3711] and therefore also RTCP SHOULD be protected using SRTCP on hbh basis according to [RFC3711]. However, the SRTP SaF extension defined herein makes no provision to provide protection of RTCP also on e2e basis. Relevant considerations (rationale and caveats) relating to RTCP are provided in Section 4.6.

3.3. Requirements on e2e Key Management

The following applies to the generic Store-and-Forward problem. For specifics about SRTP SaF key management, see Sections 4.4.2 and 5.5. In the following we assume that the hbh and e2e key management are handled independently of each other.. As the receiver needs to map the received packets to a cryptographic context, the middlebox needs to store information that can later identify the context. This information can either be the entire context or an identifier for the context.

If the entire context is transferred via the middlebox, the context needs to be protected with a shared (sender-receiver) secret key or the receiver's public key. As no negotiation or message exchange influencing the key generation can be assumed to take place a half-roundtrip key management protocol needs to be used. The middle box may at most contribute some freshness nonce for the key derivation performed by the sender. The cryptographic context used by the sender must then be transferred to the receiver as is.

Credentials used to protect the cryptographic context against intercept by the middlebox may have been (pre-)shared by direct communication (sender-receiver) or obtained via a trusted third party

(PKI or some corresponding secret key infrastructure).

If an identifier for the context is used, the context itself can be exchanged before, concurrently with, or after the message has been stored. Context exchange before storage places no specific requirements on the key management; any existing key management protocol could be used as long as it is possible to bind the generated cryptographic context to the identifier. Context exchange concurrent with storage requires that sender and receiver are both online. This is reasonable for media distribution use cases, but not for answering machine use cases. Context exchange after message storage requires that the receiver has no influence on the session key, so key management protocols requiring negotiation (e.g. Diffie-Hellman) cannot be used. If sender and receiver are online at the same time, the context exchange can be done directly between them and otherwise it can be done via a trusted third party.

3.4. Problems with SRTP in SaF Scenarios

It would be desirable to be able to offer use of SRTP as a general, lightweight mechanism to achieve the above type of protection, but trying to do so reveals two main problems.

The first problem is due to the fact that RTP streams recorded and later resent by an middlebox in general are independent; received SRTP-encrypted payloads cannot just be stored and later retransmitted as a new SSRC is most likely used when retransmitting. And if several recorded streams are spliced together, an offset must be added to the SEQ and timestamp so that they form a continuous sequence. This in particular implies that SRTP with currently defined transforms cannot be applied end-to-end as they depend on the RTP header.

The second problem is that in order to provide both e2e and hbh protection, two independent security contexts with associated protection mechanisms have to coexist; a feature unavailable in SRTP as currently specified. While it is not too difficult to imagine how two contexts in place of one might be used, a problem arises when specifying how the e2e part of the context should be identified and signaled, as current SRTP context definition rests on parameters which are not constant end-to-end in the SaF scenario, namely SSRC and receiver’s IP address and port.

The SRTP SaF extension defined in this document addresses these problems.

3.5. Design Rationale

As noted above, different SaF scenarios may have slightly different security requirements and trust levels and there may be many different possibilities to extend SRTP in different directions to handle a specific SaF use case (or some subset of use cases). For example, the problems related to the most basic trust model extension (need to provide confidentiality e2e and integrity hbh) are due to the fact that in SRTP, parties always know both the encryption key and the authentication key. This could be addressed (mainly) by just separating encryption and authentication keys (i.e. modifying SRTP key derivation and cryptographic context). However, the solution would then become severely limited, e.g. it would not support pre-encryption of data or re-transmission of stored data. Similarly, as will be seen below, SRTP SaF adds some additional in-band data fields, though some use cases above could probably be handled without them. Again, the solution would be limited to these use-cases and would then not allow e.g. the secure fast-forward/rewind use cases, which requires in-band synchronization data. By making the added fields optional, it is possible to support these features as needed, yet keeping bandwidth low when such features are not needed.

Another approach would be to use some already defined standard like S/MIME or OpenPGP, which are mostly used for secure email. This works well when the entire message is e2e protected and transferred as a file from sender to middlebox and from middlebox to receiver, but it does not support streaming. Another drawback is that both S/MIME and OpenPGP require the use of public keys. Protecting each RTP packet with both SRTP, and S/MIME or OpenPGP would support streaming but would add significant overhead. Use of (D)TLS or IPsec is clearly ruled out since it would only provide hbh protection.

This specification is rather based on

- identifying the common denominator(s) to the SaF use case problems, captured in the trust model and requirements of Section 3.2
- proposing a single extension of the SRTP framework (see Section 4.1) powerful enough to handle all foreseen SaF use cases, which, by
- simple configuration of the extended framework (Section 4.3) can be adopted to support the requirements of the specific SaF use case at hand.

Considering that the impacts of the present specification on SRTP are very similar to those of [RFC4383], there does not appear to be any disadvantage in having a single SaF extension compared to having per-SaF-use-case extensions.

4. Usage of SaF Security within SRTP

4.1. The SaF Extension

The SaF extension consists of a new packet format (Section 4.3), an extended cryptographic context concept (Section 4.4), and new SRTP processing at sender/receiver (Section 4.5). Considering only the cryptographic processing, SaF middleboxes are compatible with [RFC3711], and the necessary additional processing is defined in Section 4.5.2. Senders/receivers need to support new cryptographic transforms (see Section 4.7).

4.2. Terminology

A SaF e2e session is defined as the set of SaF e2e protected data produced under a single e2e context (a security association between sender and the ultimate receiver, see Section 4.4.1 for the exact definition of e2e context). A SaF e2e session may comprise several so-called SaF sources, i.e. several distinct logical e2e media streams to be protected by the same e2e context.

A SaF hbh session is defined as the set of SaF hbh protected data produced under a single hbh context (a security association between two entities, see Section 4.4 for the exact definition of hbh context).

The cryptographic transforms, keys, etc., used for the e2e and hbh protection, respectively, are denoted e2e transform, hbh transform, e2e key, hbh key, etc.

4.3. SRTP SaF Packet Format

Figure 2 illustrates the format of the SRTP packet when SaF is applied.

The packet format is composed of an "inner" e2e (sender-receiver) part embedded in an "outer" hbh (sender-middlebox or middlebox-receiver) part. Between these parts, a new CCI field (explained below) is introduced.

The e2e protected portion provides e2e encryption of the payload, RTP padding, and RTP pad count. The e2e protected portion also defines two new fields (PUV and SSS) for cryptographic synchronization, and an e2e MAC tag field.

The e2e MAC tag covers the e2e protected portion, except the e2e MAC tag itself. Whether authentication implies source origin authentication or only message integrity depends on the transform

used. Thus, e2e encryption is provided over the Payload, RTP padding, and RTP pad count fields, while authentication is provided for the PUV and SSS as well.

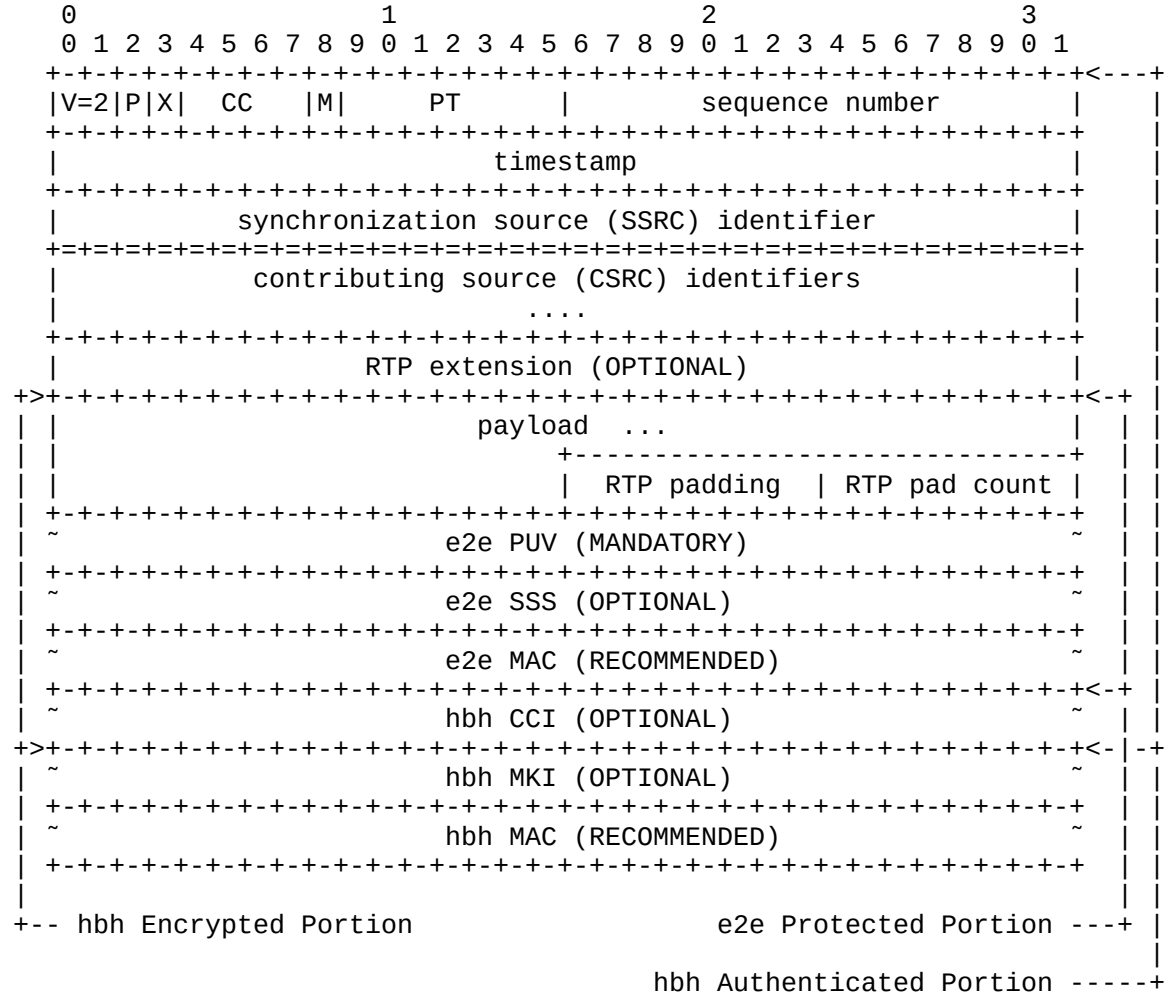

Figure 2: The format of the SRTP packet when SaF is applied.

Default e2e transforms, which provide both encryption and authentication, and which SHALL be supported are defined in Section 4.7.3.

The e2e protected portion is opaque from SaF middlebox point-of-view.

Thus, by treating the inner e2e protected portion and the Crypto

Context Identifier (CCI, see below) as the (hbh) "encrypted portion" of [RFC3711], the overall SRTP SaF packet format conforms to standard [RFC3711] compliant SRTP. (Note that the additional fields added in the inner e2e part could just as well have been added by a new transform defined for SRTP, e.g. padding and/or crypto synch fields.) Hence, the hbh MAC and hbh MKI are in one-to-one correspondence with the MAC and MKI of [RFC3711] and will not be discussed further.

The additional fields added by the inner e2e security processing are:

- SSS: SRTP SaF Source is a value used by the SRTP SaF transform as an identifier for the SaF source within a SaF e2e session. Thus, SSS MUST be unique for all SaF sources within the SaF e2e session. Since there may be only one such SaF source, the SSS field is OPTIONAL and of configurable length. SSS resembles the SSRC usage in RTP/SRTP in the sense that it ensures that two-time pads do not occur under the same e2e master key, see Sections 4.7 and 5.2. The implementation of the necessary anti-collision mechanism is outside the scope of this specification. The format is implementation specific, but the values 0, 1, 2, ..., n-1 can be used if there are n SaF sources.

- PUV: Packet Unique Value for the e2e transform. PUV is transform dependent, of configurable length, and MANDATORY. The format is transform dependent and security aspects need to be considered when defining the format, see Sections 5.2 and 5.4. For a given SaF e2e session and SaF source, the PUV SHALL be unique for each generated e2e protected portion. The PUV is used as input to the IV formation for the e2e encryption transform.

- MAC (e2e): This field is used to carry payload authentication data e2e. It is transform dependent, of configurable length and is RECOMMENDED to be used. Observe that the e2e MAC SHALL cover the RTP payload, the PUV and SSS but SHALL NOT cover the RTP header, nor the CCI.

- CCI: Crypto Context Identifier: used to signal hbh, which e2e cryptographic context (keys and other parameters, see Section 4.4.1) to use. The field is OPTIONAL and of configurable length. The format is implementation specific, but the values 0, 1, 2, ..., n-1 can be used if there are n e2e contexts.

Parameters which are configurable have default values (see Section 4.7.4), and are otherwise negotiated during SaF e2e/hbh session establishment, agreed upon out of band, or hard coded for a specific application. The new fields are of configurable length for maximum data compactness (see Table 4.1).

Field	SRTP Counterpart	Typical Size (bytes)
PUV	SRTP Index	3
SSS	SSRC (IV formation)	0-1
MAC (e2e)	MAC (hbh)	4-10
CCI	SSRC (context identification)	0-1
Total		7-15

Table 4.1: Additional parameters

4.4. Extension of the SRTP Cryptographic Context

A SRTP SaF cryptographic context SHALL consist of two main parts.

1. A hbh context. The hbh context SHALL be an SRTP cryptographic context conforming to [RFC3711] and SHALL be used for the hbh protection between sender and SaF middlebox, between SaF middlebox and receiver, or, between two SaF middleboxes. The hbh context SHALL thus be identified by the <SSRC, destination network address, destination port number> triplet exactly as defined in [RFC3711].

2. One (or more) e2e contexts: this part of the context is defined below and SHALL be used for the e2e protection between sender and receiver.

The motivation for allowing more than one e2e context is to support scenarios where the SaF middlebox and receiver use a single (S)RTP session into which several e2e protected sessions are spliced, see Appendix A for use cases.

If the SRTP SaF context contains more than one e2e context, then each e2e context SHALL be associated with a unique CCI value. Since the length of the CCI field is variable, the length of the CCIs SHALL be determined by a length parameter, n_CCI.

4.4.1. Definition of e2e Context

The e2e context SHALL contain the following e2e transform independent parameters.

- an identifier for the e2e encryption algorithm, i.e., the cipher and its mode of operation, see Section 4.7.3 for the default e2e encryption transform specification,

- an identifier for the e2e message authentication algorithm, see Section 4.7.3 for the default e2e message authentication transform

specification,

- an identifier for the e2e pseudo-random function,
- an e2e master key, which MUST be random and secret to all except sender and receiver. The e2e master key MUST be cryptographically independent of any hbh key,
- an e2e master salt. Use of e2e master salt is strongly RECOMMENDED. This value, when used, MUST be random, but MAY be public.
- non-negative integers n_e, n_a, n_s, and n_tag determining the length of the e2e session keys for encryption and message authentication, the e2e session salt, and the e2e authentication tag,
- non-negative integers n_PUV, and n_SSS determining the length of the PUV, and SSS fields.

There may also be need to include e2e transform dependent parameters, see Section 4.7.3 for the parameters associated with the default e2e transforms.

Observe that there is no replay protection data in the e2e context, see Section 4.5.3.1. Also note that unlike [RFC3711] cryptographic contexts, the e2e context SHALL only contain parameters for RTP protection, and SHALL NOT contain parameters for RTCP protection, see Section 4.6.

Only end-points need to support e2e contexts, i.e. senders and receivers. SaF middleboxes need, however, to understand the usage of the e2e context identifiers (CCI) as discussed next.

If a SaF Middlebox needs to send information to the receiver that is not e2e protected and not associated with an e2e context beforehand, the SaF middlebox SHALL create an associated e2e context and follow the processing steps for SRTP SaF. A typical use-case is e.g. an answering machine sending supporting information in-band to the receiver, e.g. "You have three new messages", or, to the sender, e.g. "Please leave a message after the tone".

4.4.2. Identification of e2e Context

The e2e context SHALL be identified by out-of-band (outside the SRTP SaF Packet) and in-band (in the SRTP SaF Packet) signaling.

4.4.2.1. Out-of-band Signaling

The e2e context MAY be identified by simply transferring the entire context out-of-band (e.g. in the SIP signaling). Only half-roundtrip key management protocols can be used and the e2e context MUST be e2e protected so that middleboxes or other unauthorized entities cannot access or modify it.

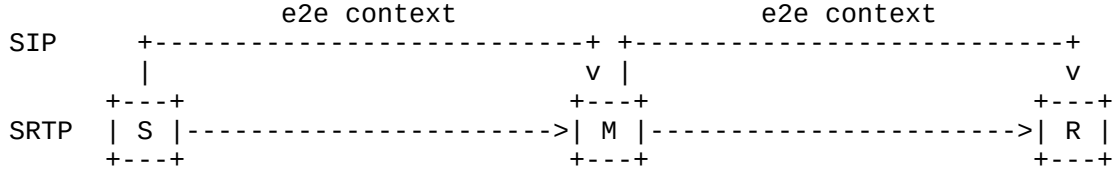

Figure 3: Transferring the entire e2e context via the middlebox

Alternatively, out-of-band context identification may use indirection and SHALL then be defined as follows. For each e2e context, a Content ID (CID) is defined. When used, the CID MUST uniquely determine the context between a sender and a receiver but the exact format of the CID is outside the scope of this specification. For example, a statistically unique (e.g. 256-bit) value may be used. The CID is communicated by out-of-band means:

- e2e (together with the e2e context), directly between sender and receiver or via a trusted third party.
- hbh, between sender and SaF middlebox, between two SaF middleboxes (as applicable), and between SaF middlebox and receiver.

How the CID communication is done (which protocol to use etc.) is outside the scope of this specification. Any SRTP key management protocol (e.g. DTLS-SRTP) can be used. If the e2e communication is done directly between sender and receiver, they must be simultaneously online, which is reasonable in many use cases (e.g. media distribution). The e2e communication MAY be done before or after the communication with M.

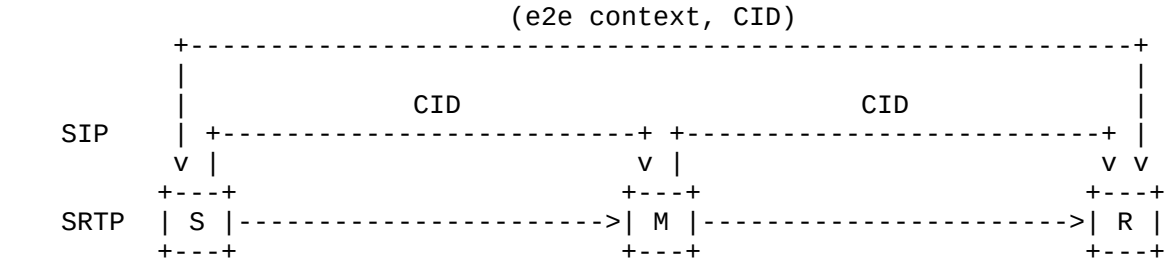

Figure 4: Indirection via CID

In the sequel, the term CID will often be used instead of "CID/e2e context".

4.4.2.2. In-band Signaling

The in-band context identification is similar to that of [RFC3711] and SHALL be defined as follows. The e2e context is uniquely identified by the quadruplet context identifier:

<CCI, SSRC, destination network address, destination port number>

The e2e context (either the context itself or its CID) is here implicit from the quadruplet. If the SaF context contains more than one e2e context, the triplet context identifier [RFC3711] cannot uniquely identify the e2e context and the context identifier has therefore been extended with the CCI field in the SRTP SaF packet (see Figure 2). Just like in [RFC3711] the entire quadruplet MAY not be needed to uniquely identify the e2e context. Note that different quadruplets identify the e2e context on different "hops".

The CCI may thus be thought of as a short, in-band alias for the e2e context (possibly via additional indirection through CID) and is only used on hbh basis. If multiple pieces of content corresponding to multiple CIDs are transferred within the same SaF hbh session, the source (initial sender or SaF middlebox) SHALL ensure the use of distinct CCIs for all CIDs.

4.4.2.3. Mapping CCI to CID

The SaF middlebox SHALL, in conjunction to informing the next hop destination about the CID values, also inform if and how it has associated the CCIs to CIDs, e.g. as part of session setup signaling. For instance, the SaF middlebox MAY provide pairs of form:

(CID1, CCI1), (CID2, CCI2), ...

During transfer of e2e protected content associated with a certain CID, the source (initial sender or SaF middlebox) SHALL add the associated CCI to each packet being part of that content.

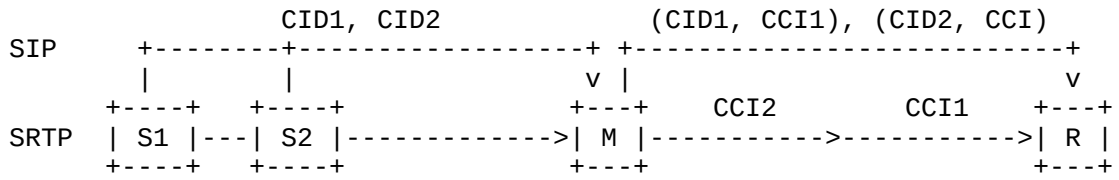

Figure 5: Switching between e2e contexts

Figure 5 illustrates CCI to CID mapping and e2e context switching. The senders S1 and S2 stores messages for the same receiver R on a SaF middlebox M. They specify different CIDs (CID1, CID2) in the signaling to M. When R later retrieves the content from M it is multiplexed inside the same hbh session. Different CCIs are used to identify the e2e contexts. Before the messages are sent, M informs R about the CIDs and their corresponding CCIs.

4.5. SRTP SaF Processing

In what follows, it is assumed that the sender and receiver agree out-of-band on the e2e cryptographic context.

It is similarly assumed that sender-middlebox and middlebox-receiver, respectively, agree on the hbh cryptographic context.

4.5.1. Sender

The sender SHALL first, out-of-band, establish the necessary CIDs, CCIs, and hbh context parameters with the SaF middlebox as discussed above. The rest of sender’s processing is identical to [RFC3711] with the following exceptions and extensions.

S1 In analogy with step 1 of [RFC3711], the sender SHALL determine both the hbh context and the e2e context as discussed in Section 4.4. Next, and prior to performing step 2 of [RFC3711], the sender SHALL perform step S2-S6 as defined below.

S2 The sender SHALL from the e2e master key and master salt derive the e2e session key/salt as described in Section 4.7.2.

S3 The sender SHALL next apply the e2e encryption transform as described in Section 4.7.3.

S4 The sender SHALL next apply the e2e authentication transform as described in Section 4.7.3 to the result of S3 concatenated by the PUV and the SSS.

S5 The sender SHALL then form the e2e protected portion of the SRTP SaF packet by concatenating the result of S3, the PUV, the SSS, and the e2e MAC from S4.

S6 The sender SHALL then concatenate the result of S5 and the CCI.

The rest of the sender's processing conforms to [RFC3711], steps 2-8, by treating the result of S6 as the part to be encrypted ("encrypted portion" of [RFC3711]) and using the hbh context.

4.5.2. SaF Middlebox

SaF middleboxes do not have access to the e2e contexts and may even be unaware of their definition. Hence, "context" in this section refers to standard [RFC3711] cryptographic contexts, which in turn agrees with the hbh contexts defined herein.

Generally, the SaF middlebox SHALL first, out-of-band, establish the necessary CIDs, CCIs, and hbh context parameters with the source or destination.

4.5.2.1. Acting as Receiver ("Store")

MR1 When receiving media from a sender, the SaF middlebox SHALL retrieve the correct context and process the packet exactly according to the receiver behavior of [RFC3711].

MR2 The SaF middlebox SHALL store sufficient information to later be able to map the correct content to the intended receiver, e.g. e2e context, the CID, or the intended receiver's identity (ID). ID format and usage is otherwise out of scope for this specification, but could, e.g., be retrieved during the session establishment.

MR3 The SaF middlebox SHALL store information sufficient to later reconstruct the e2e protected portion of the packets (corresponding to Figure 2) and to allow the receiver to uniquely identify the correct e2e context, e.g. by storing the CID or the e2e context. Note that header information (e.g. P, M, PT, and timestamp) is needed for rendering and information from RTCP SR is used for synchronization between streams e.g. in a multimedia video/audio session. Such information also has to be stored by the SaF middlebox.

4.5.2.2. Acting as Sender ("Forward")

MS1 When forwarding media to the receiver, the SaF middlebox SHALL retrieve the correct hbh context as specified in [RFC3711].

MS2 A payload SHALL be formed by concatenating the e2e protected portion and the CCI.

MS3 MS3 An RTP Header SHALL be formed with the P, M, and PT fields identical to when the payload was stored. The original timestamp MAY be used, but if several messages are spliced together, an offset needs to be added to that the hbh timestamps form a continuous sequence. The SEQ and SSRC SHOULD be defined strictly on hbh basis.

MS4 The SaF middlebox SHALL then concatenate the payload from MS2 with the RTP header from MS3 and process the packet exactly according to the sender behavior of [RFC3711] using the retrieved context. As noted above, certain information from RTCP messages, originating from the sender (e.g. RTCP SRs), may also need to be forwarded (and sometimes modified as discussed in Section 4.6). These (and other RTCP messages) SHALL be processed according to the SRTCP specification of [RFC3711].

4.5.2.3. Multiple SaF Middleboxes

When more than one SaF middlebox is present, we consider a pair of adjacent SaF middleboxes M1 and M2, where M1 forwards media to M2.

M1 SHALL act as a SaF middlebox sender (Section 4.5.2.2) treating M2 as a receiver. M2 SHALL act as a SaF middlebox receiver (Section 4.5.2.1) treating M1 as a sender.

4.5.3. Receiver

R1 The receiver SHALL first, out-of-band, establish the necessary CIDs, CCIs, and hbh context parameters with the SaF middlebox.

R2 Step 1 to 8 of [RFC3711] SHALL then be applied, using the hbh context to perform hbh processing.

The remainder of the processing concerns the e2e protection. The result after performing the hbh authentication check and decryption as described above MAY be stored at the receiver for later application of the e2e processing. If so, the receiver MUST store the e2e protected portion and the CCI in order to be able to perform the further steps as described below.

R3 The receiver SHALL next determine the e2e context as discussed in Section 4.4.2. (In case the CCI was NOT used or NOT encrypted by the hbh transform, the receiver MAY determine the e2e context already in step R1.)

R4 The receiver SHALL derive the e2e session encryption/ authentication key(s) as describe in Section 4.7.2 using the e2e master key and salt.

R5 The receiver SHALL verify authentication and decrypt the e2e protected portion as specified by the e2e transform(s), see Section 4.7.3. If the result of authentication is "FAILURE", the packet MUST be discarded from further processing and the event SHOULD be logged. Note that there is no replay protection for the e2e context (see Section 5.4).

R6 The receiver removes PUV, SSS, e2e MAC, and CCI as appropriate.

4.5.3.1. Replay Protection

For reasons discussed in Appendix A, it is in general not meaningful or desirable to provide application independent replay protection for the e2e part. Some of the identified use cases make this clear by having a requirement that the receiver should be able to jump back/ forward in the e2e media stream. See Section 5.4 for security considerations.

4.6. Use of SRTCP with SRTP SaF

SRTCP protection SHALL be provided hbh, conforming to [RFC3711], and SHALL NOT be provided e2e, as this covers most/all use cases currently identified. Protecting e.g. the synchronization information e2e would prevent use cases where several stored streams are spliced together. Further RFCs may specify additional e2e functionality for SRTCP SaF.

As noted, it may still be needed to forward information from some of the inbound RTCP messages (e.g. RTCP SR and APP). Note that if several stored streams are spliced together, the timestamps and therefore also the synchronization information has to be modified. Also note that it may in general not be possible for the SaF middlebox to reproduce RTCP reports accurately reflecting the ongoing SaF hbh session. For instance, since the e2e encryption hides any possible RTP padding, there may be a discrepancy between sender’s byte counts on the S-M and M-R links, respectively. After decryption at R, however, the correct values will be possible to reconstruct.

4.7. Cryptographic Transforms

We define a set of pre-defined SRTP SaF e2e transforms. Note that SaF middleboxes do not need to support any cryptographic transform outside what is already defined in [RFC3711]. The hbh protection may reuse any of the existing SRTP transforms such as those defined in the original specification [RFC3711], or, transforms that have been added later. The e2e protection may use the transforms defined in Section 4.7.1, or, transforms that have been added later (see Section 4.7.5). When separate e2e transforms are used for encryption and authentication, the sender SHALL first apply the e2e encryption transform and then the e2e authentication transform.

4.7.1. Pre-Defined e2e Transforms

The pre-defined e2e encryption transform is AES-CM (AES Counter Mode) as specified in [RFC3711], Section 4.1.1, with the following modification. Instead of forming the initialization vector as defined in [RFC3711], the IV SHALL be formed as:

$$IV = (k\_s * 2^{16}) \text{ XOR } (SSS * 2^{64}) \text{ XOR } (PUV * 2^{16})$$

where k_s is the session salting key (derived from the e2e master key and salt, see Section 4.7.2) and where SSS and PUV are the SSS/PUV fields from the packet. The PUV is a counter, initially set to zero and then increasing by one (1) for each packet. The maximum allowed size of the PUV for AES-CM SHALL be 48 bits. If the SSS field is not present, the value 0 (zero) SHALL be used. The maximum allowed size of the SSS for AES-CM SHALL be 64 bits.

The key used SHALL be the session encryption key k_e (derived from the e2e master key and salt, see Section 4.7.2).

The pre-defined e2e authentication transform is HMAC-SHA1 as defined in [RFC3711], Section 4.2.1, with the difference that it SHALL be applied to the e2e protected portion, excluding the e2e MAC field itself. Note also that the e2e MAC SHALL NOT be applied to the CCI field. The resulting MAC tag SHALL be inserted in the e2e MAC field.

The key used SHALL be the session authentication key k_a (derived from the e2e master key and salt, see Section 4.7.2).

If a key management protocol designed to key the original SRTP specification [RFC3711] is used to set up the e2e part of SRTP SaF, the values that specify AES-CM and HMAC-SHA1 in [RFC3711] SHALL specify the transforms defined in Section 4.7.1.

4.7.2. Session Key Derivation

For the hbh security processing, session key derivation SHALL be done exactly as in [RFC3711] using the hbh master key and salt.

For the e2e security processing the key derivation is also identical to [RFC3711] with the following exceptions

- The e2e master key and salt, SHALL be used together with the defined labels of [RFC3711] for derivation of the different keys.
- The key derivation rate SHALL be zero.

4.7.3. Default Transforms

The default hbh encryption transform SHALL be the NULL encryption algorithm, the default hbh authentication transform SHALL be HMAC-SHA1, and the default hbh pseudo-random function SHALL be AES-CM. The transforms are defined in Sections 4.1.1, 4.2.1, and 4.3.3 of [RFC3711].

The default e2e encryption transform SHALL be AES-CM as defined in Section 4.7.1. The default e2e authentication transform SHALL be HMAC-SHA1 as defined in Section 4.7.1. The default e2e pseudo-random function SHALL be AES-CM as defined in [RFC3711], Section 4.3.3.

4.7.4. SRTP SaF Default Parameters

The default hbh parameters SHALL be identical to [RFC3711].

The default e2e parameters for master and session key lengths are the same as in [RFC3711] with the differences in transform definition as defined above and the following additional exception.

- Replay window size: N/A (or 0).

We also add the following additional parameters:

parameter	min	default
n_PUV	16	24
n_SSS	0	0
n_CCI	0	0

Table 4.2: Additional parameters

4.7.5. Adding Future e2e Transforms

Adding transforms for the hbh protection SHALL follow the existing guidelines of [RFC3711]. Indeed, any current (or future, as far as we can see) transform specification for SRTP is applicable for usage with the hbh protection.

To add an e2e transform, the accompanying specification MUST, besides specifying the cryptographic operations, define the format and usage of the PUV field and, if used, for the SSS field and any possible additional field, e.g. padding. An authentication transform MUST define how the e2e MAC is computed and MUST NOT include the CCI field in the authentication coverage.

When separate transforms are used for encryption and authentication, the sender SHALL first apply the e2e encryption transform and then the e2e authentication transform. When a combined (data encapsulation) transform is used, the order of processing is typically built in to the transform.

5. Security Considerations

5.1. General

Though it may seem that there are quite a few differences between the cryptography and key management used in [RFC3711] and the corresponding functions defined here, the differences are actually smaller then one may think and the security considerations turn out to be essentially equivalent.

As noted, a problem of SRTP in SaF applications is the transforms' dependence of the SSRC. The SSRC is part of IV formation and crypto context identification in [RFC3711].

In this specification three new in-band parameters, PUV, CCI, and SSS, are specified. Note that CCI and SSS are used in exactly the same way the SSRC is used in [RFC3711]: context identification (CCI) and IV formation (SSS). Basically, one can think of the CCI as the e2e context identifier and the SSS as the e2e source identifier. The SSS is e2e protected. As the e2e authentication fails if the CCI is modified it does not need to be e2e protected (similar to MKI [RFC3711]).

5.2. Keystream Reuse

A main concern of [RFC3711] is to avoid keystream reuse. This concern is present also here. The currently defined encryption

transforms are additive stream ciphers, which are sensitive to keystream reuse. It is therefore RECOMMENDED that each session utilizes random and cryptographically independent e2e and hbh keys.

When sender and receiver share an e2e master key it may be convenient to reuse the key for several e2e sessions/messages via the SaF middlebox. Another situation when key reuse may be beneficial is if sender and receiver use the SaF middlebox in a "chat-like" fashion (with bi-directional communication using the same e2e master key in both directions). In this case there may be a risk that a message in one direction (e.g. "A-to-B") reuses keystream of some message in the other direction ("B-to-A"). For the predefined e2e encryption transform such reuse will only be secure if the sender and receiver keep state to prohibit reuse of IVs.

Unique IVs MAY be assured by putting requirement on the implementation of the sender to ensure that unique SSS values are used each time the same e2e master key is reused. For the bidirectional case (as well as for the more general case where a group key is used as e2e master key), some out-of-band signaling that assures that end-points use distinct SSSs is, as mentioned, REQUIRED.

The situation is essentially equivalent to that of SRTP. As noted in the security considerations of [RFC3711], keys may be reused (with the predefined transforms) if (and only if) unique SSRC values can be guaranteed. Due to the risks of misuse, reuse of master keys between sessions is, just as in [RFC3711], therefore NOT RECOMMENDED.

5.3. Authentication and Authorization

For reasons already discussed, it is RECOMMENDED that middleboxes authorize senders and receivers (typically involving authentication) before storing/forwarding messages. While the content is protected by keys supposedly only known to the receiver, this provides extra protection if the e2e keys have fallen into the wrong hands and it also avoids that the SaF middlebox wastes resources, responding to spoofed requests. It is also RECOMMENDED to have e2e authentication between sender and receiver, which is achieved by applying authentication/integrity to the e2e protected portion.

5.4. Replay Protection

Replay protection is provided on an hbh basis by use of a hbh transform including message authentication. It is RECOMMENDED to use hbh message authentication as it protects from outsiders attempting to change the order of packets.

Since some scenarios considered makes it reasonable to expect that

the receiver may wish to jump (fast-forward or rewind) in the e2e protected media flow, it is not meaningful to strictly enforce replay protection on an e2e basis. Note however that our trust model assumes that the SaF middleboxes are trusted enough not to attempt to replay or reorder media unless the receiver so requests.

It is however still possible (and RECOMMENDED) to provide e2e authentication of the packets in combination with inclusion of a sequence number in the PUV (as the default e2e transform does). It then becomes infeasible even for the SaF middlebox to fake the relative association between a particular packet and its sequence number. This means that the receiver will be able to detect a replay that occurs without the receiver actually having requested it.

5.5. Key Management Considerations

Key management is outside the scope of this specification which is an intentional design choice in order not to introduce any dependency on using a specific key management scheme. Nevertheless, some considerations need to be highlighted and taken into account when deploying this specification in practice.

To implement the targeted trust model, the main concern is that the e2e keys MUST be independent from the hbh keys. In other words knowledge of any hbh key MUST NOT reveal non-trivial information about any e2e key.

This can be achieved by ensuring that key management for hbh and e2e protection is carried out independently using fresh, random and independent keys each time. This is the RECOMMENDED approach.

Another alternative which may be attractive in some cases is to use the slightly weaker notion of cryptographic independence. Here, the hbh keys MAY be derived from the e2e keys by applying a sufficiently strong pseudo-random function.

Even if hbh keys are random and independent each time, it is still RECOMMENDED that e2e keys are not cached/reused (see Section 5.2 for discussion on keystream reuse).

5.6. Privacy

In order for a SaF middlebox to deliver the correct media (produced with the correct e2e context) to the receiver, some SaF applications may choose to store information regarding the identity of the sender and will be able to deduce the communication taking part between the two.

To enhance privacy, senders/receivers may use agreed pseudonyms or other similar Privacy Enhancing Techniques (PET)s. Such techniques are to use random CIDs, and to assign CCIs independent on the order the messages were stored. Complete anonymity may be in conflict with the requirement that the SaF middlebox needs protection from flooding by garbage or other forms of unwanted traffic.

5.7. RTCP Considerations

As specified, RTCP is only protected on hbh basis. This is motivated by the assumption that a SaF middlebox indeed is a true store-and-forward entity (as opposed to performing a more intelligent function). The inbound/outbound RTP sessions are then different and RTCP then reports only on the current RTP session. As noted though, it may still be useful to forward e.g. (modified) sender reports to the receiver using hbh RTCP protection.

5.8. Malicious middleboxes

SaF middleboxes are semi-trusted which implies that they are assumed to (at least) forward data as requested by the sender/receiver. Malicious middleboxes therefore falls outside the trust model. Nevertheless, even if a SaF middlebox is malicious beyond our assumptions, such attacks will only have DoS effects if e2e authentication is used (RECOMMENDED) and could as easily have been done by some other party (non-middlebox). If hbh authentication is used (RECOMMENDED), it can even be detected that it is the middlebox that modified the e2e protected part.

By modifying RTCP, a malicious middleboxes could perform attacks that are not detected but which would be detected if done by some other party (non-middlebox). By incorrectly altering RTCP SR packets, a malicious middlebox could forward only parts of messages or mess with the synchronization information without being detected. However, for the intended use cases, there seems to be no gain to the middlebox owner (typically a network operator) to perform such attacks to its paying customers.

6. Acknowledgements

The authors would like to thank Daniel Catrein, Steffen Fries, Frank Hartung, Joerg Ott, and Magnus Westerlund for their support and valuable comments. We are also grateful to Eric Rescorla for his feedback when reviewing the -00 version of this draft.

7. IANA Considerations

To signal that the new transforms are used, each relevant key management protocol needs to register the new transforms including numbering scheme and syntax with IANA.

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.

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

8.2. Informative References

[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.

Appendix A. Use Cases

In the use cases below, we map the entities to the trust model of Section 3.2 by indicating which entity that corresponds to S, M, R.

A.1. Streaming Pre-encrypted Media

A content creator (S) wants to distribute high value content to clients (R). The content provider distributes the media via a streaming server (M) which should not have access to cleartext media, typically because it is not trusted by the content creator.

A.2. Recording Encrypted Media at Home

High value encrypted media (e.g. IPTV, and radio) is broadcasted in a network. Only clients trusted by the content creator (S) have access to the encryption key. A user (R) is recording the media on a Hard Disk Drive (M), but does not yet have a license or have a license that does not allow cleartext copying. The media is therefore stored in protected format on the HDD.

A.3. Answering Machine

Operators commonly provide an answering machine service to their customers. In this case the communicating parties (S and R) may not wish to disclose the media to any other party, and hence want to apply encryption between each other. The answering machine (M) acts as a SaF middlebox, which has to store encrypted data and retransmit it to the callee.

In this use case it is also likely that several callers leave messages protected by different e2e keys. As discussed in the SRTP SaF specification, the receiver and SaF middlebox may agree to use a single hbh context into which the different e2e contexts are multiplexed using the CCI.

A.4. Media Rewind

Common to the use cases above is the possible desire to be able to rewind or jump forward in the media stream. For instance, a user may wish to listen once again to a message left in a voice mail without terminating and reinitiating the session with the SaF middlebox.

Appendix B. Test Vector

The parameters are chosen to be typical for a voice call. A frame size of 32 bytes is used by AMR 12.2 (Adaptive Multi-Rate) and with 20 ms speech frames, a PUV length of 24 bits equals 93.2 h. A 32-bit MAC offers good integrity protection for a voice call. The 16 bit SSS is not typical but is included to make the test vector more general.

```
Encryption algorithm:                    AES-CM
Authentication algorithm:                HMAC-SHA-1
Pseudo Random Function:                  AES-CM

n_e (encr session key length):           128
n_a (auth session key length):           160
n_s (session salt key length):           112
n_PUV (Packet Unique Value length):       24
n_SSS (SRTP SaF Source length):           16
n_tag (Authentication tag length):        32
```

The values below are in hexadecimal.

```
e2e Master key (128 bits)
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
```

```
   e2e Master salt (112 bits)
   40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d

   e2e Session encryption key (128 bits)
   12 ed 05 3a f7 8c 9a f2 96 5c 64 26 f4 d1 56 23

   e2e Session authentication key (160 bits)
   73 0c 3c ac 1d 75 27 36 91 97 d4 ab c2 b4 6b 46
   cd e0 19 83

   e2e Session salting key (112 bits)
   eb 31 d1 cb af 09 68 cd 14 f2 2b be 35 18

   Packet Unique Value (24 bits)
   80 81 82

   SRTP SaF Source (16 bits)
   c0 c1

   Payload (256 bits)
   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

   e2e Protected Portion (328 bits)
   82 37 69 bd f8 9c f3 61 57 e4 3d 74 b7 e6 07 4b
   05 80 52 ec 7d 68 72 63 b2 e1 10 ae b9 7b 7c a0
   80 81 82 c0 c1 bd ab 1e f6
```

Authors' Addresses

Rolf Blom
Ericsson AB
SE-164 80 Stockholm
Sweden

Phone: +46 10 71 31 707
Email: rolf.j.blom@ericsson.com

Yi Cheng
Ericsson AB
SE-164 80 Stockholm
Sweden

Phone: +46 10 71 17 589
Email: yi.cheng@ericsson.com

Fredrik Lindholm
Ericsson AB
SE-164 80 Stockholm
Sweden

Phone: +46 10 71 31 705
Email: fredrik.lindholm@ericsson.com

John Mattsson
Ericsson AB
SE-164 80 Stockholm
Sweden

Phone: +46 10 71 43 501
Email: john.mattsson@ericsson.com

Mats Naslund
Ericsson AB
SE-164 80 Stockholm
Sweden

Phone: +46 10 71 33 739
Email: mats.naslund@ericsson.com

Karl Norrman
Ericsson AB
SE-164 80 Stockholm
Sweden

Phone: +46 10 71 44 502
Email: karl.norrman@ericsson.com