

HIPRG  
Internet-Draft  
Intended status: Informational  
Expires: April 26, 2007

H. Tschofenig  
M. Shanmugam  
Siemens Networks GmbH & Co KG  
F. Muenz  
October 23, 2006

Using SRTP transport format with HIP  
draft-tschofenig-hiprg-hip-srtp-02.txt

## Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April 26, 2007.

## Copyright Notice

Copyright (C) The Internet Society (2006).

---

Internet-Draft Using SRTP transport format with HIP

October 2006

## Abstract

The Host Identity Protocol (HIP) is a signaling protocol which adds a new layer between the traditional Transport and the Network layer. HIP is an end-to-end authentication and key exchange protocol, which supports security and mobility in a commendable manner. The HIP base specification is generalized and purported to support different key exchange mechanisms in order to provide confidentiality protection for the subsequent user data traffic. This draft explains a mechanism to establish Secure Real Time Protocol associations, to protect the user data packets, by using HIP.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction . . . . .</a>	<a href="#">4</a>
<a href="#">2.</a>	<a href="#">Terminology . . . . .</a>	<a href="#">5</a>
<a href="#">3.</a>	<a href="#">Goals . . . . .</a>	<a href="#">6</a>
<a href="#">4.</a>	<a href="#">SRTP Parameter Exchange in HIP . . . . .</a>	<a href="#">7</a>
<a href="#">4.1.</a>	<a href="#">Setting up an SRTP Association . . . . .</a>	<a href="#">7</a>
<a href="#">4.2.</a>	<a href="#">Rekeying . . . . .</a>	<a href="#">8</a>
<a href="#">5.</a>	<a href="#">Parameter and Packet Formats . . . . .</a>	<a href="#">9</a>
<a href="#">5.1.</a>	<a href="#">General Parameters (SRTP_PARAM) . . . . .</a>	<a href="#">9</a>
<a href="#">5.2.</a>	<a href="#">Timestamp (SRTP_T) . . . . .</a>	<a href="#">11</a>
<a href="#">5.3.</a>	<a href="#">Pseudo-random byte-string (SRTP_RAND) . . . . .</a>	<a href="#">11</a>
<a href="#">5.4.</a>	<a href="#">Security Policies (SRTP_SP) . . . . .</a>	<a href="#">12</a>
<a href="#">5.5.</a>	<a href="#">Master Key Identifier (SRTP_MKI) . . . . .</a>	<a href="#">14</a>
<a href="#">5.6.</a>	<a href="#">NOTIFY Parameter . . . . .</a>	<a href="#">14</a>
<a href="#">6.</a>	<a href="#">Packet Processing . . . . .</a>	<a href="#">15</a>
<a href="#">6.1.</a>	<a href="#">Processing Outgoing Application Data . . . . .</a>	<a href="#">15</a>
<a href="#">6.2.</a>	<a href="#">Processing Incoming Application Data . . . . .</a>	<a href="#">15</a>
<a href="#">6.3.</a>	<a href="#">HMAC and SIGNATURE Calculation and Verification . . . . .</a>	<a href="#">15</a>
<a href="#">6.4.</a>	<a href="#">Processing Incoming SRTP Initialization (R1) . . . . .</a>	<a href="#">15</a>
<a href="#">6.5.</a>	<a href="#">Processing Incoming SRTP Reply (I2) . . . . .</a>	<a href="#">16</a>
<a href="#">6.6.</a>	<a href="#">Dropping HIP Associations . . . . .</a>	<a href="#">16</a>
<a href="#">6.7.</a>	<a href="#">Initiating SRTP master key rekeying . . . . .</a>	<a href="#">16</a>
<a href="#">6.8.</a>	<a href="#">Finalizing Rekeying . . . . .</a>	<a href="#">17</a>
<a href="#">6.9.</a>	<a href="#">Processing NOTIFY Packets . . . . .</a>	<a href="#">18</a>
<a href="#">7.</a>	<a href="#">Implementation Considerations . . . . .</a>	<a href="#">19</a>
<a href="#">8.</a>	<a href="#">Security Considerations . . . . .</a>	<a href="#">20</a>
<a href="#">8.1.</a>	<a href="#">Denial of Service . . . . .</a>	<a href="#">20</a>
<a href="#">8.2.</a>	<a href="#">Man in the Middle Attack . . . . .</a>	<a href="#">20</a>
<a href="#">8.3.</a>	<a href="#">Eavesdropping . . . . .</a>	<a href="#">20</a>
<a href="#">8.4.</a>	<a href="#">Authentication . . . . .</a>	<a href="#">21</a>
<a href="#">9.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">22</a>
<a href="#">10.</a>	<a href="#">Acknowledgements . . . . .</a>	<a href="#">23</a>
<a href="#">11.</a>	<a href="#">References . . . . .</a>	<a href="#">24</a>
<a href="#">11.1.</a>	<a href="#">Normative References . . . . .</a>	<a href="#">24</a>
<a href="#">11.2.</a>	<a href="#">Informative References . . . . .</a>	<a href="#">24</a>

Authors' Addresses . . . . .	<a href="#">25</a>
Intellectual Property and Copyright Statements . . . . .	<a href="#">26</a>

Internet-Draft Using SRTP transport format with HIP October 2006

## [1.](#) Introduction

The Host Identity Protocol (HIP) [[I-D.ietf-hip-base](#)] provides a way to separate the dual role of IP (end point identifier and locator) by adding a new layer between the traditional Network and Transport layer. This separation helps the end host to achieve mobility, furthermore, HIP provides better security features (like end-to-end authentication, confidentiality for the data traffic etc) than other multi6 proposals.

HIP is based on public key cryptography. All HIP hosts have a public/private key pair. HIP introduces a new name space called Host Identity. It is nothing but the public key of an asymmetric key pair. It provides a rapid exchange of host identities (public keys) between communicating hosts, after mutual authentication, a HIP association is established. For operational purposes, HIP uses Host Identity Tags (HITs) [[I-D.ietf-hip-base](#)], which is hash of the public key. During the base exchange, the hosts generate a shared keying material using an authenticated Diffie-Hellman exchange. Note that the HIP base exchange provides mutual authentication of the hosts, but does not specify any mechanism for protecting data packets. [[I-D.ietf-hip-esp](#)] draft proposes a way to use IPsec ESP format with HIP.

Secure Real Time Protocol (SRTP) is a profile for Real Time Protocol (RTP), which provides a framework for providing encryption, integrity, message authentication, confidentiality and protection against replay attacks for the real-time data traffic. SRTP mandates the use of an external key management protocol to exchange keys and

cryptographic parameters, which are used to derive keys (like cipher suites, random number etc.,). This draft proposes a way to exchange SRTP relevant parameters during the HIP base exchange. [Appendix A](#) describes one possible use case to support this document.

This document is organized as follows. [Section 4](#) describes the revised base exchange, [Section 5](#) presents the packet format, [Section 6](#) explains the packet processing and [Section 7](#) talks about the key management.

This document was developed in the context of investigating the benefits of using HIP for SIP.

## [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 [[RFC2119](#)] .

This draft uses the terminology defined in [[I-D.ietf-hip-base](#)] and [[RFC3261](#)] .

The term MKI, an optional parameter, refers to Master Key Identifier used in SRTP packets.

### [3.](#) Goals

To facilitate the use of SRTP, the HIP base exchange messages require some minor additions to the parameters transported. In the R1 packet, the responder adds the necessary parameters, which contains transforms and other related information for the SRTP association, before sending it to the Initiator. The Initiator gets the proposed transforms, selects one of those proposed transforms, and adds it to the I2 packet, as well as other information in the corresponding parameters.

In this context, the goal of our proposal is to,

- o define new parameter exchange for exporting the relevant SRTP parameters in the HIP base exchange.

- o describe the relevant packet formats.

#### [4.](#) SRTP Parameter Exchange in HIP

In this section, the protocol for setting up an SRTP association to be used with HIP association is described.

##### [4.1.](#) Setting up an SRTP Association

Setting up an SRTP Association between hosts using HIP consists of

two messages passed between the hosts. The parameters are included in R1 and I2 messages during base exchange.

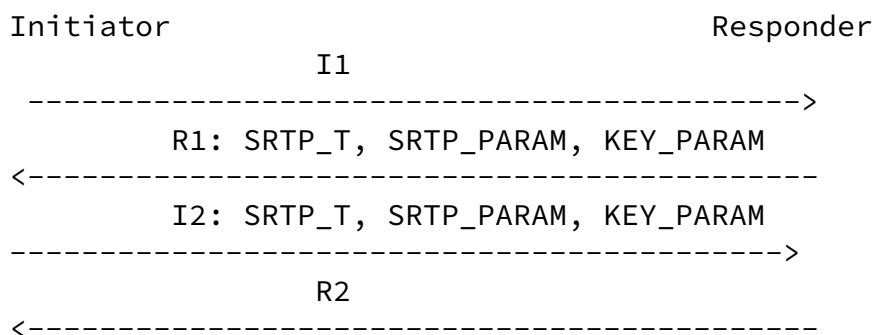


Figure 1: Setting up an SRTP Association

The integration of HIP and SRTP requires some changes, as mentioned earlier, in the HIP parameters. The changes are (will be) adding,

SRTP\_T: The timestamp, used mainly to prevent replay attacks.

The SRTP\_PARAM contains the information about the general description of the message exchange. It is used for mapping a Crypto Sessions (CS) to security protocol sessions.

KEYING parameter (KEY\_PARAM) contains

SRTP\_RAND: Random/pseudo-random byte-string, RAND(nonce) is used as a fresh value for the key generation.

SRTP\_SP: The security policies for the data security protocol. (eg. algorithms and transforms and PRFs supported by the peers). The cipher suites can be negotiated from R1/I2 packet.

SRTP\_MKI : to identify the Master key and Master salt.

The R1 message contains the KEYING PARAMS, in which the sending host defines the possible Algorithms and transforms, random number and, optionally, a MKI it is willing to use for the SRTP association.

The I2 message contains the response to KEYING PARAMS received in the

R1 message. The sender must select one of the proposed transforms



from the SP parameter in the R1 message and include the selected one in the SP parameter in the I2 packet. In addition to the transform, the host includes the RAND parameter, containing the random value (and optionally MKI) to be used as a salt by the peer host. In the R2 message, HIP exchange is finalized.

#### [4.2.](#) Rekeying

Rekeying can be supported using the UPDATE packet of HIP. The peer which wants to rekey should use the UPDATE packet with the appropriate parameters. The mechanism is explained below:

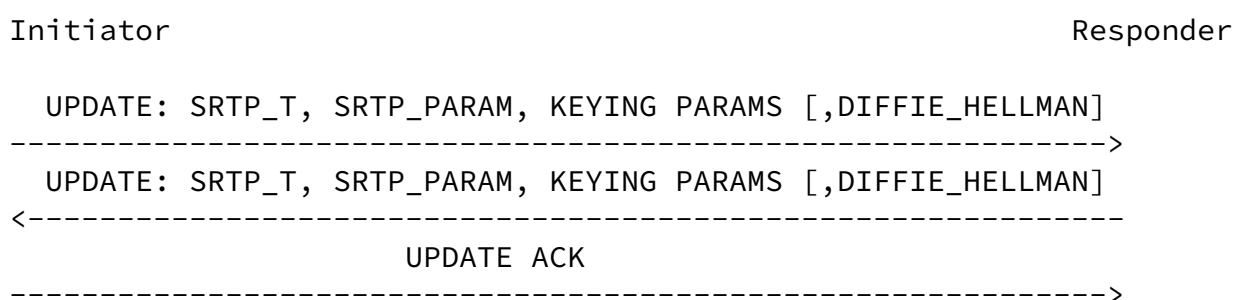


Figure 2: Rekeying mechanism

Figure 2 depicts the rekeying scenario. Here, assume that the Initiator wants to rekey after the Initial exchange. It can send the rekeying parameters in the Update packet, includes its new Diffie-Hellmann value, and sends it to the Responder. It may send a new MKI value to identify the incoming packet.

The other parameters are explained in [[I-D.ietf-hip-base](#)]. The Responder checks the return routability by sending the Update seq message containing its Diffie-Hellmann value and relevant parameters for the rekeying. After receiving the packet, the Initiator sends the ACK thereby both the peers conclude the rekeying procedure and now on, the peers expect to receive the traffic in the new keying material.

## 5. Parameter and Packet Formats

This section discusses the SRTP related new parameters and presents the proposed packet format.

The following list gives an overview of the parameters to be exchanged in addition to the HIP base exchange:

Master Key and its length - obtained from Diffie-Hellmann key exchange

Master Salt - RAND in the KEYING parameter

MKI - Master Key Identifier

Security Policies (SP) - Each policy will define pseudo-random functions, algorithms and transforms for the establishment of a Cryptographic Context for SRTP.

Timestamp - Used for preventing replay attacks.

Session keys are derived using Master key, Master salt and SP and the details are up to the key management protocol.

The new parameters contain five elements summarized in the following table:

Parameter	Type	Length	Data
SRTP_PARAM	40000	variable	Mapping Crypto Sessions to security protocol sessions
SRTP_T	40001	4	Used mainly to prevent replay attacks
SRTP_RAND	40002	14	used as a fresh value for the key generation
SRTP_SP	40003	variable	algorithms, transforms, PRFs supported by the peers
SRTP_MKI	40004	variable	Master Key Identifier

The parameters SRTP\_RAND, SRTP\_SP and SRTP\_MKI together form the KEYING (KEY\_PARAM) parameters.

### 5.1. General Parameters (SRTP\_PARAM)

The general parameter is used for mapping a Crypto Sessions (CS) to

security protocol sessions.

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               |                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
!                               CSB ID                               !
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
! #CS                          ! CS ID map info                    | RESERVED |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
! Policy_no_1    ! SSRC_1                                           !
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
! SSRC_1 (cont) ! ROC_1                                           !
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
! ROC_1 (cont)  ! Policy_no_2    ! SSRC_2                           !
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
! SSRC_2 (cont)                               ! ROC_2                           !
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
! ROC_2 (cont)                               !                               :
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
:                               :                               :
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
! Policy_no_#CS !                               SSRC_#CS           !
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
!SSRC_#CS (cont)!                               ROC_#CS           !
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
! ROC_#CS (cont)!
+---+---+---+---+---+---+

```

Fig 4: Mapping parameters

Type: 40000 (experimental identifier range)

Length: variable

Value: Mapping information

CSB ID (32 bits): identifies the CSB. It is RECOMMENDED that the CSB ID be chosen at random by the Responder and sent with the R1 packet. This ID MUST be unique between each Initiator-Responder pair, i.e., not globally unique. A Responder MUST check for

collisions when choosing the ID (if the Responder already has one or more established CSBs with the Initiator). The Initiator uses the same CSB ID in the response.

#CS (8 bits): indicates the number of Crypto Sessions that can be handled concurrently by a CSB. A CSB may handle up to 255 CS at a time. However, it is unlikely that this limited will be reached. The integer 0 is interpreted as no CS included. This may be the case in an initial setup message.

CS ID map info (16 bits): identifies the crypto session(s) for which the SA should be created.

Policy\_no\_i (8 bits): The security policy applied for the stream with SSRC\_i. The same security policy may apply for all CSs.

SSRC\_i (32 bits): specifies the SSRC that MUST be used for the i-th SRTP stream.

ROC\_i (32 bits): Current rollover counter used in SRTP.

NOTE: The stream using SSRC\_i will also have Crypto Session ID equal to no i (NOT to the SSRC).

## 5.2. Timestamp (SRTP\_T)

The timestamp, used mainly to prevent replay attacks. As in the SRTP packet format, a 32-bit value is used to store the timestamp.

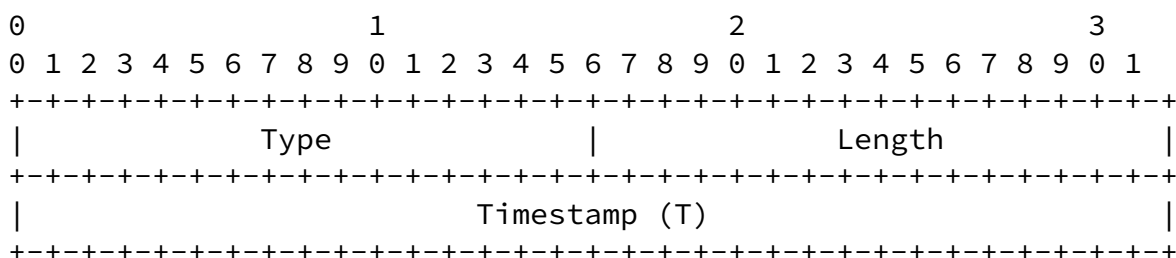


Fig 4: Timestamp parameter

Type: 40001 (experimental identifier range)  
Length: 4  
Value: Timestamp

### 5.3. Pseudo-random byte-string (SRTP RAND)

The RAND or master salt parameter is used as a fresh value for the key generation. The RAND parameter is a 112 bit quantity.

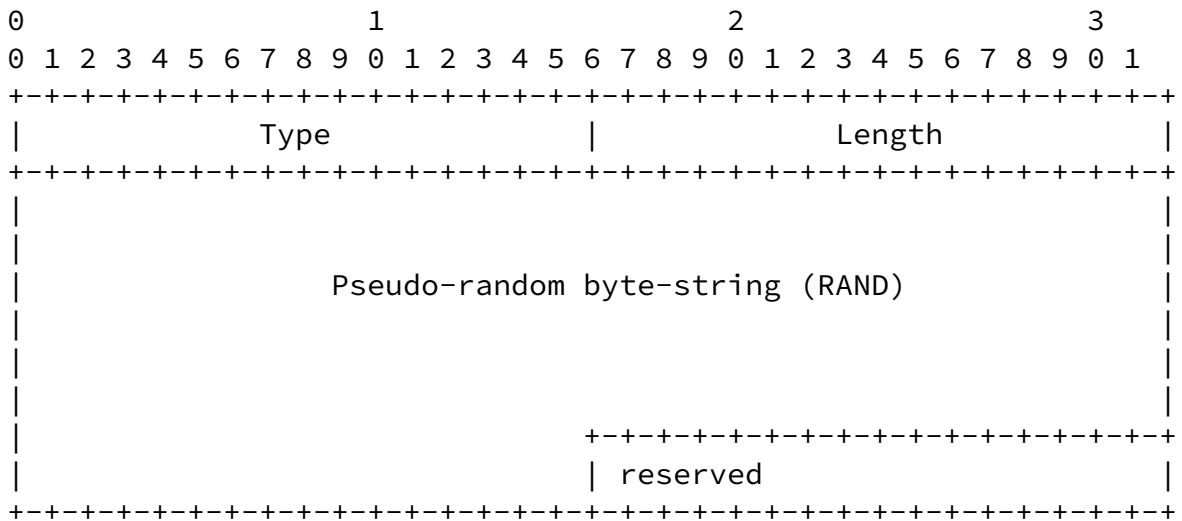


Fig 5: Pseudo-random byte-string parameter

Type: 40002 (experimental identifier range)

Length: 14

Value: Pseudo-random byte-string

#### 5.4. Security Policies (SRTP\_SP)

The security policies for the data security protocol (e.g., algorithms, transforms and PRFs supported by the peers). The cipher

suites can be negotiated from I2/R2 packet. The security policy parameters are grouped together for being used with the policy selector in the mapping parameter, that is a SRTP\_SP parameter may actually consist of multiple policies.

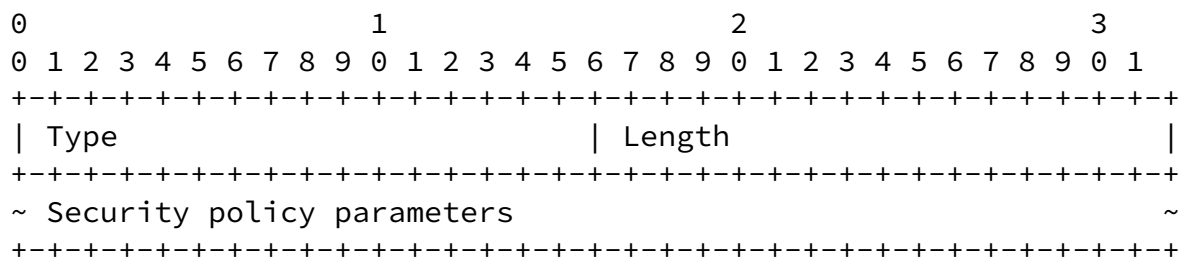
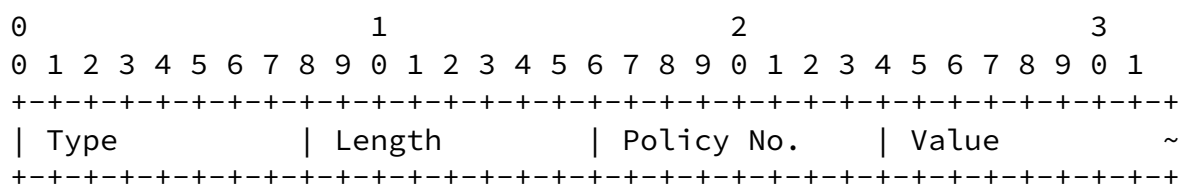


Fig 6: Security policy parameters

Type: 40003 (experimental identifier range)  
Length: variable  
Value: See below

The security policy parameters themselves are built up by a set of Type/Length/Value fields:



Type (8 bits): specifies the type of the parameter.

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

Policy No. (8 bits): specifies the policy to which this TLV belongs. It is used in conjunction with the Mapping parameter. All values (0-255) may be used for policy identification.

Value (variable length): specifies the value of the parameter.

Type	Length	Meaning	Value
-----	-----	-----	-----

1	1	SRTP and SRTCP encr transf	see below
2	2	Encr session key length	128
3	1	SRTP and SRTCP auth transf	see below
4	2	Auth session key length	160
5	2	Tag length	80
6	4	SRTP prefix_length	var(default 0)
7	1	Key derivation PRF	see below
8	8	Key derivation rate	var(default 0)
9	8	SRTP-packets-max-lifetime	var
10	8	SRTCP-packets-max-lifetime	var
11	1	Forward Error Control	2-bits

For the Encryption transforms, a one byte length is enough. The currently defined possible Values are:

SRTP and SRTCP encr transf	Value
-----+-----	
NULL	0
AES-CM	1
AES-F8	2

where AES-CM is AES in CM, and AES-F8 is AES in f8 mode [[RFC3711](#)].

For the Authentication transforms, a one byte length is enough. The currently defined possible Values are:

SRTP and SRTCP auth transf	Value
-----+-----	
NULL	0
HMAC-SHA-1	1

For the Key derivation PRF, a one byte length is enough. The currently defined possible values are:

Key derivation PRF	Value
-----+-----	
NULL	0
AES_CM	1

#### [5.5](#). Master Key Identifier (SRTP\_MKI)

The MKI identifies the master key and master salt from which the

session key(s) were derived that authenticate and/or encrypt the particular packet. This parameter is OPTIONAL.

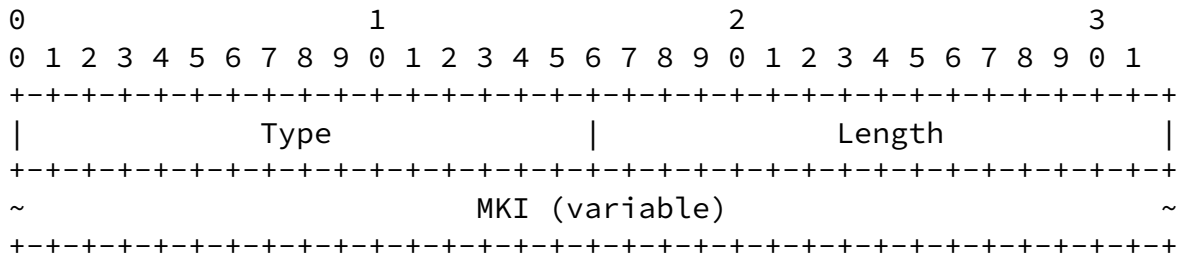


Fig 7: SRTP MKI parameter

Type: 40004 (experimental identifier range)  
Length: variable  
Value: Master Key Identifier (MKI)

5.6. NOTIFY Parameter

The HIP base specification defines a set of NOTIFY error types. The following error types are required for describing errors in SRTP security policy during negotiation.

NOTIFY PARAMETER - ERROR TYPES	Value
NO_SRTP_PROPOSAL_CHOSEN	TBD
None of the proposed SRTP Transform in the proposed security policy was acceptable.	
INVALID_SRTP_TRANSFORM_CHOSEN	TBD
The SRTP chosen parameters from the proposed security policy proposals do not correspond to those offered by the responder.	

6. Packet Processing

In general, packet processing is specified in the HIP base specification [[I-D.ietf-hip-base](#)]. This section provides an overview



of changes needed when using the new SRTP extensions.

#### [6.1.](#) Processing Outgoing Application Data

When the SRTP format is used, the outgoing application data will be encrypted using a cryptographic context. Details about the handling of outgoing SRTP traffic is described in [\[RFC3711\] Section 3.3](#).

#### [6.2.](#) Processing Incoming Application Data

When the SRTP format is used, the incoming application data will be decrypted using a cryptographic context. Details about the handling of incoming SRTP traffic is described in [\[RFC3711\] Section 3.3](#).

#### [6.3.](#) HMAC and SIGNATURE Calculation and Verification

The new HIP parameters described in this document, SRTP\_PARAM, SRTP\_T, SRTP\_RAND, SRTP\_SP and SRTP\_MKI must be protected using HMAC and signature calculations. In a typical implementation, they are included in R1, I2, and UPDATE packet HMAC and SIGNATURE calculations as described in [\[I-D.ietf-hip-base\]](#).

#### [6.4.](#) Processing Incoming SRTP Initialization (R1)

The SRTP key exchange is initialized in the R1 message. The receiving host (Initiator) selects the SRTP transforms from the presented values in the R1 packet and establishes its SRTP Cryptographic Context. For this the SRTP\_SP will provide the transforms and pseudo-random functions, the SRTP\_MAPPING parameters will provide information about what policy applies for which Crypto Session (CS). If no suitable value is found, the negotiation is terminated.

The selected values are subsequently used when generating and using session keys according to the negotiated SP, and when sending the reply packet I2. If the proposed alternatives are not acceptable to the system, it may abandon the SRTP establishment negotiation, or it may resend the I1 message within the retry bounds.

After creating cryptographic context, and performing other R1 processing, the system prepares and creates session keys derived from the exchanged master key complying to the negotiated SPs. The Initiator will then send the I2 packet with the selected SRTP transforms.

### [6.5.](#) Processing Incoming SRTP Reply (I2)

The following steps are required to process the incoming SRTP initialization reply in I2 for creating the correct Cryptographic Context on Responder side. It is assumed that the I2 packet has been accepted for processing (e.g., has not been dropped due to HIT comparisons as described in [[I-D.ietf-hip-base](#)]):

The SRTP\_SP and SRTP\_MAPPING parameters are verified and they MUST have the same number of proposed policies in R1 and each policy MUST match the values offered in the initialization packet.

The SRTP\_MKI field is parsed to obtain the MKI that will be used for selecting the appropriate master key.

The system creates session keys derived from the master key, master salt and security policy for a certain SRTP stream.

Upon successful processing of the initialization reply message, a possible old master key and session keys are dropped and the new ones are installed, and a finalizing packet, R2, is sent. Possible ongoing rekeying attempts are dropped.

### [6.6.](#) Dropping HIP Associations

When the system drops a HIP association, as described in the HIP base specification, the SRTP layer and any results from exchanges are not affected.

### [6.7.](#) Initiating SRTP master key rekeying

During SRTP rekeying, the hosts exchange new master keys from which session keys will be derived. Use of the MKI for rekeying is RECOMMENDED

A system may initiate the SRTP rekeying procedure at any time. The system MUST NOT replace its current master key until the rekeying packet exchange successfully completes.

The rekeying procedure uses the UPDATE mechanism defined in [[I-D.ietf-hip-base](#)]. Because each peer must update their master keys, the rekeying process requires that each side both send and receive an UPDATE. A system will then rekey the session keys when it has sent parameters to the peer and has received both an ACK of the relevant UPDATE message and corresponding peer's parameters. It may be that the ACK and the required HIP parameters arrive in different UPDATE messages. This is always true if a system does not initiate

SRTP update but responds to an update request from the peer, but may

also occur if two systems initiate update nearly simultaneously. In such a case, if the system has an outstanding update request, it saves the one parameter and waits for the other before completing rekeying.

The following steps define the processing rules for initiating an SRTP update:

The system decides whether to continue to use the existing master key or to create a new one. In the latter case, the system **MUST** generate a new Diffie-Hellman public key. When using SRTP default transforms, the master key **MUST** be replaced before any of the index spaces are exhausted for any of the streams protected by one and the same master key.

The system creates an UPDATE packet, which contains the SRTP\_PARAM, SRTP\_T, SRTP\_RAND, SRTP\_SP and SRTP\_MKI parameter. In addition, the host **MUST** include DIFFIE\_HELLMAN parameter.

The system sends the UPDATE packet. For reliability, the underlying UPDATE retransmission mechanism **SHOULD** be used.

The system **MUST NOT** delete its existing master key, but continue using them if its policy still allows.

In case a protocol error occurs and the peer system acknowledges the UPDATE but does not itself send an SRTP parameters, the system may not finalize the outstanding session key update request. To guard against this, a system **MAY** re-initiate the SRTP update procedure after some time waiting for the peer to respond, or it **MAY** decide to abort the SRTP update after waiting for an implementation-dependent time. The system **MUST NOT** keep an outstanding SRTP update request for an indefinite time.

To simplify the state machine, a host **MUST NOT** generate new UPDATES while it has an outstanding SRTP update request, unless it is restarting the update process.

## [6.8.](#) Finalizing Rekeying

A system finalizes rekeying when it has both received the corresponding UPDATE acknowledgement packet from the peer and it has successfully received the peer's UPDATE. The following steps are taken:

If the received UPDATE messages contains a new Diffie-Hellman key, the system has a new Diffie-Hellman key from initiating SRTP update, or both, the system selects the new master key.

The system draws session keys from the new master key.

The system cancels any timers protecting the UPDATE.

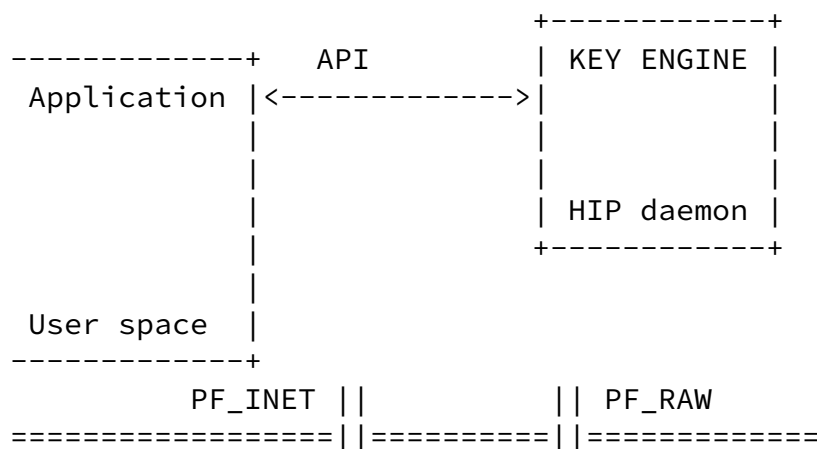
#### [6.9](#). Processing NOTIFY Packets

The processing of NOTIFY packets is described in the HIP base specification.

## 7. Implementation Considerations

This section explains the implementation considerations for the HIP-SRTP exchange. After the initial base exchange, both peers have the same master key, salt and agreed crypto transforms (including pseudo random function). When the application receives the data traffic after the base exchange, an API is invoked and asks the HIP daemon for the appropriate key to process the data packet.

Figure 15 depicts the example implementation architecture of the proposed mechanism:



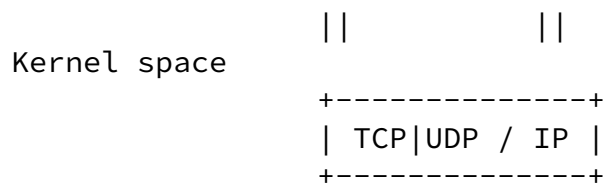


Figure 15: Example Implementation Architecture

## [8.](#) Security Considerations

Security is considered throughout this document.

The initial keying material is generated using using Diffie-Hellman procedure. This document extends the usage of UDPATE packet, defined in the base specification, for rekeying. The hosts may rekey for the generation of new keying material using Diffie-Hellman procedure. This mechanism enjoys the security protection provided by base exchange using HMAC and signature verifications.

In this approach, we have tried to extend the HIP base exchange to support SRTP based key management scheme. We have listed the following security mechanisms that are incorporated with this idea:

### [8.1.](#) Denial of Service

Threat:

A denial of service attack typically overloads the attacked nodes by exploiting any state creation, CPU intensive calculation or simply overloading their maximum available bandwidth.

Countermeasures:

This approach enjoys the merits of HIP like resisting CPU and memory exhaustive DoS attacks by forcing the caller to calculate the solution for a cryptographic puzzle. This provides only a basic DoS protection for the callee.

## [8.2.](#) Man in the Middle Attack

Threat:

An adversary might want to modify the parameters that are exchanged.

Countermeasures:

HIP uses authenticated Diffie-Hellmann key exchange, which prevents the man-in-the-middle (MitM) attacks. The exchanged parameters are protected in the same fashion as IPSec parameters are when HIP is used for setting up IPSec security associations.

## [8.3.](#) Eavesdropping

Threat:

A possible passive attack of an adversary is placing itself between

communication partners and collect data, that is exchanged between them. As a result the adversary may learn about communication and encryption details.

Countermeasures:

Since the data traffic is encrypted, it is unreadable for the attackers.

## [8.4.](#) Authentication

Threat:

A malicious node may impersonate another node and perform actions on behalf of this node for it's own needs.

Countermeasures:

Both peers are authenticated using asymmetric key (signature verification) cryptography assuming that public keys can be acquired by secure ways.

## [9.](#) IANA Considerations

This document defines several new name spaces associated with the HIP payloads. This section summarizes the name spaces for which IANA is requested to manage the allocation of values. IANA is requested to



record the pre-defined values defined in the given sections for each name space. IANA is also requested to manage the definition of additional values in the future.

This document defines five new HIP parameters, namely SRTP\_PARAM, SRTP\_T, SRTP\_RAND, SRTP\_SP and SRTP\_MKI. These parameters currently use the experimental identifier range of the HIP protocol. A decision must be made on the final values.

The name spaces for the fields in the Security Policies parameter (from [Section 5.4](#)) are requested to be managed by IANA. All proposed values are summarized in the tables given in this section.

The name spaces for the fields in the Notify parameter (from [Section 5.6](#)) are requested to be managed by IANA.

## 10. Acknowledgements

The authors would like to gratefully acknowledge Pekka Nikander and Jari Arkko for their comments to this document.

This document is a byproduct of the Ambient Networks Project, partially funded by the European Commission under its Sixth Framework Programme. It is provided "as is" and without any express or implied warranties, including, without limitation, the implied warranties of fitness for a particular purpose. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Ambient Networks Project or the European Commission.

## [11.](#) References

### [11.1.](#) Normative References

#### [I-D.ietf-hip-base]

Moskowitz, R., Nikander, P., Jokela, P., and T. Henderson, "Host Identity Protocol", [draft-ietf-hip-base-06](#) (work in progress), June 2006.

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

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

### [11.2.](#) Informative References

#### [I-D.ietf-hip-esp]

Moskowitz, R., Nikander, P., and P. Jokela, "Using ESP transport format with HIP", [draft-ietf-hip-esp-03](#) (work in progress), June 2006.

[RFC3261] Schulzrinne, H., Camarillo, G., Rosenberg, J., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "Session Initiation Protocol", February 2005.

Internet-Draft      Using SRTP transport format with HIP

October 2006

#### Authors' Addresses

Hannes Tschofenig  
Siemens Networks GmbH & Co KG  
Otto-Hahn-Ring 6  
Munich, Bavaria 81739  
Germany

Phone: +49 89 636 40390  
Email: Hannes.Tschofenig@siemens.com  
URI: <http://www.tschofenig.com>

Murugaraj Shanmugam  
Siemens Networks GmbH & Co KG  
Otto-Hahn-Ring 6  
Munich, Bayern 81739  
Germany

Email: murugaraj.shanmugam@siemens.com

Franz Muenz

Email: franz.muenz@thirdwave.de

Internet-Draft

Using SRTP transport format with HIP

October 2006

#### Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

#### Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

#### Acknowledgment

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