HIPRG
Internet-Draft
Expires: April 26, 2006

H. Tschofenig
Siemens
F. Muenz
FH-Landshut
M. Shanmugam
TUHH

October 23, 2005

Using SRTP transport format with HIP
draft-tschofenig-hiprg-hip-srtp-01.txt

Status of this Memo

Copyright Notice

Abstract

The Host Identity Protocol (HIP) is a signaling protocol which adds a new layer between the traditional Transport and 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 genralized and purported to support different key

exchange mechanisms in order to provide confidentiality protection for the subsequent data traffic. In some cases it might not be desirable to establish IPsec security associations for protection of media traffic. This draft explains how keying material and parameters for usage with the Secure Real Time Protocol (SRTP) can be established using HIP.

Table of Contents

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 [I-D.ietf-hip-multi6].

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 and (optionally) establishes IPsec SAs to protect subsequent data traffic. It is a four-way handshake protocol, which supports end-to-end authentication and the data traffic may experience IPsec ESP encapsulation.

Transport connections and security associations between the communicating HIP hosts are bound to the HITs and IP addresses are used for routing purposes only. Therefore, changes to IP addresses do not change the connections or associations. So, when any of the peers move (mobility scenarios), it uses a readdressing mechanism to update the current location of the peer, thereby supporting mobility in a seamless manner.

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 a 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 the SRTP relevant parameters during the HIP base exchange. Besides this, we inherited the key derivation procedure of SRTP to show how the keys will be manipulated and maintained for the data traffic. Appendix A describes one possible use case to support this document.

This document is organized as follows. Section 3 explains the revised base exchange, Section 4 explains the rekeying scenario,

Section 5 presents the packet format and Section 6 explains the key derivation, and future work.

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 optionl parameter, refers to Master Key Identifier used in SRTP packets.

3. Goal

The HIP base exchange is used to set up a HIP association between two hosts. The base exchange provides two-way host authentication and key material generation, but it does not provide any means for protecting data communication between the hosts. In this document, we specify the use of SRTP for protecting user data traffic after the HIP base exchange. Note that we did not consider the key management issues in this draft.

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 possible KEYING Parameter 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 in the corresponding KEYING Parameter.

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

- define new parameter exchange for the relevant SRTP parameters.

- define the relevant packets structure and parameters.

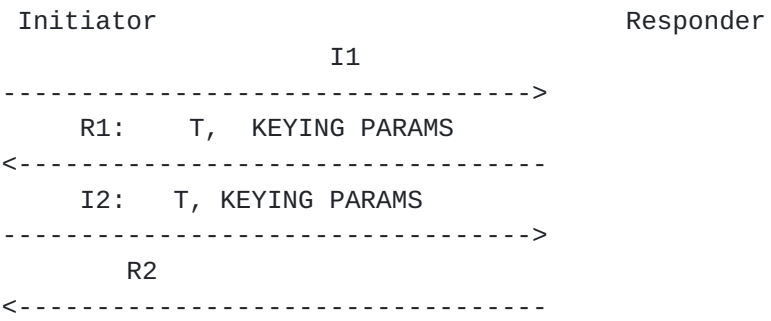
4. The Protocol

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

4.1. SRTP in HIP

4.1.1. 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.



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

T: The timestamp, used mainly to prevent replay attacks.

KEYING parameter contains

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

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.

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 MKI it is willing to use for the SRTP association.

The I2 message contains the response to an 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.

4.1.2. 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:

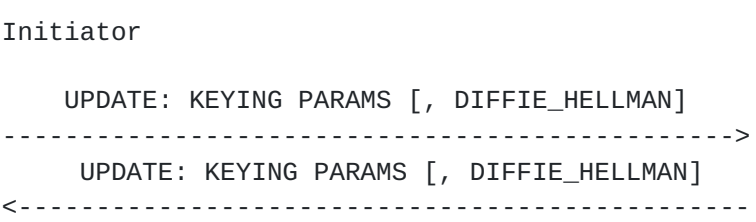


Fig 2:Rekeying mechanism

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. The same mechanism is followed here, the Initiator chooses its 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 concluding the rekeying procedure and now, both of the peers expect to receive the traffic in the new keying material.

5. Parameter and Packet Formats

This section explains the relationship between the SRTP and KEYING parameter and presents the proposed packet format.

Master Key - derived from Diffie-Hellmann value

Master Salt - RAND in the KEYING parameter

MKI - Master Key Identifier

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

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

As discussed previously, KEYING parameters contains four element:

5.1. Timestamp

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

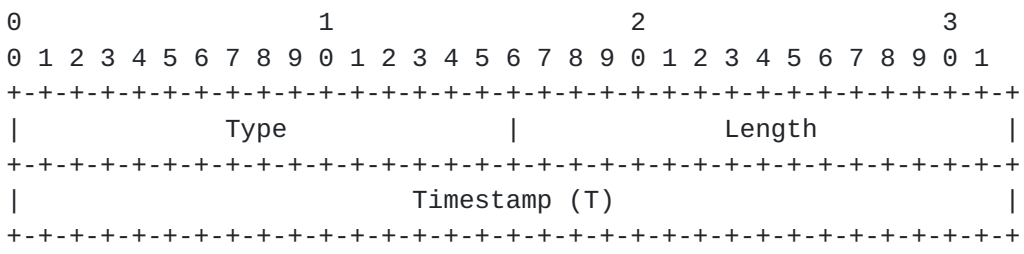


Fig 4: Timestamp parameter

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

5.2. Pseudo-random byte-string (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.

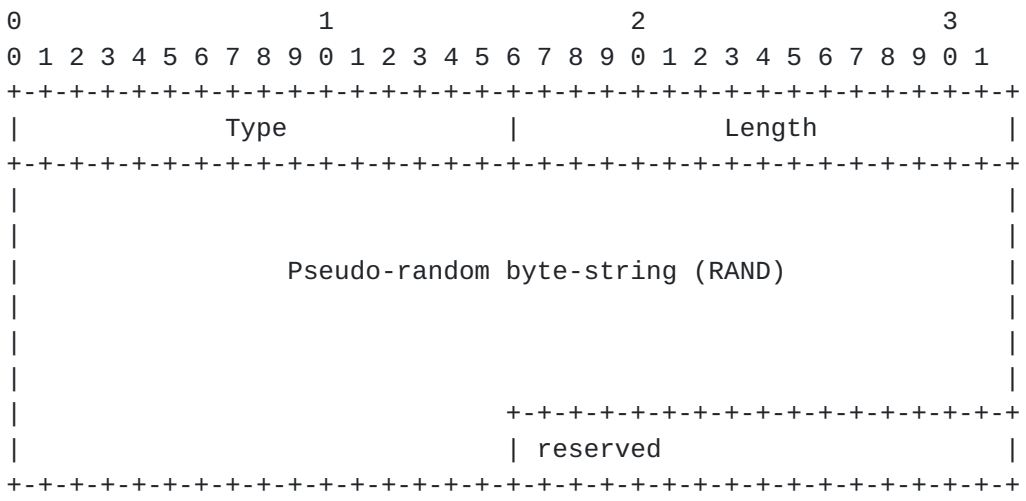


Fig 5: Pseudo-random byte-string parameter

Type: 40002 (experimental identifier range)
Length: 14
Value: Pseudo-random byte-string

5.3. Security Policies (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 I2/R2 packet.

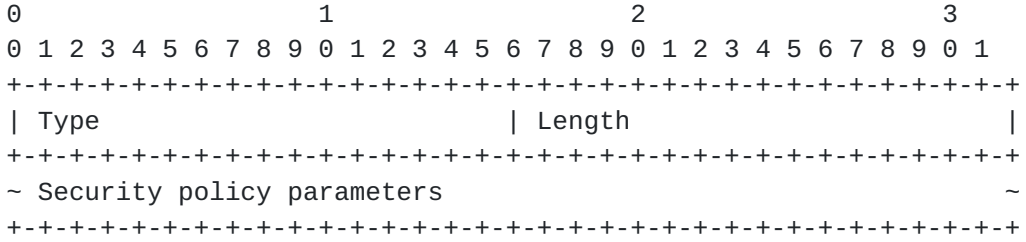
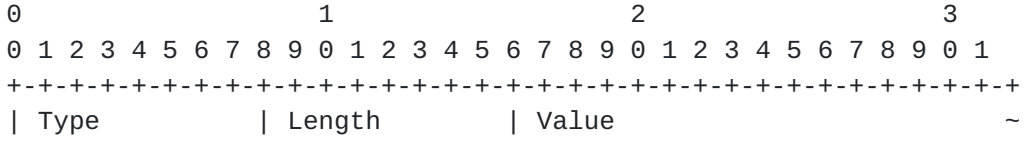


Fig 6: Security policy parameters parameters

Type: 40004 (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).

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

5.4. Master Key Identifier (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.

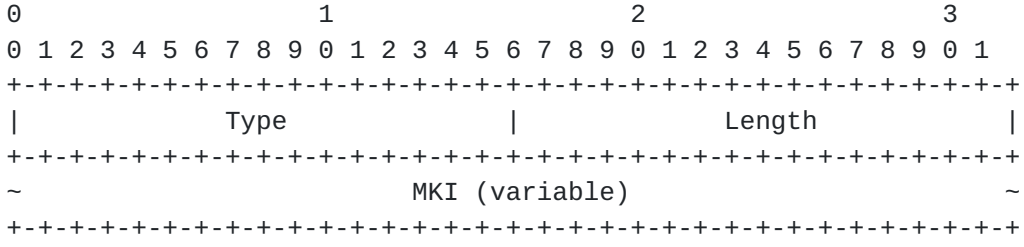


Fig 7: SRTP MKI parameter

Type: 40001 (experimental identifier range)
Length: variable
Value: Master Key Identifier (MKI)

6. Key management

This section explains how the key management scheme can be used for the data traffic. 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.

The SRTP based key derivation helps to generate the session keys for both peers, so that they have the same keys in possession for encrypting/decrypting the incoming packets. It generates three keys namely encryption key to provide confidentiality for the data packets, authentication key for providing integrity and salt key for the AES counter mode. For that, it uses the master key, salt and crypto transforms together with the packet index.

Figure 6 depicts the example implementation architecture of the proposed mechanism:

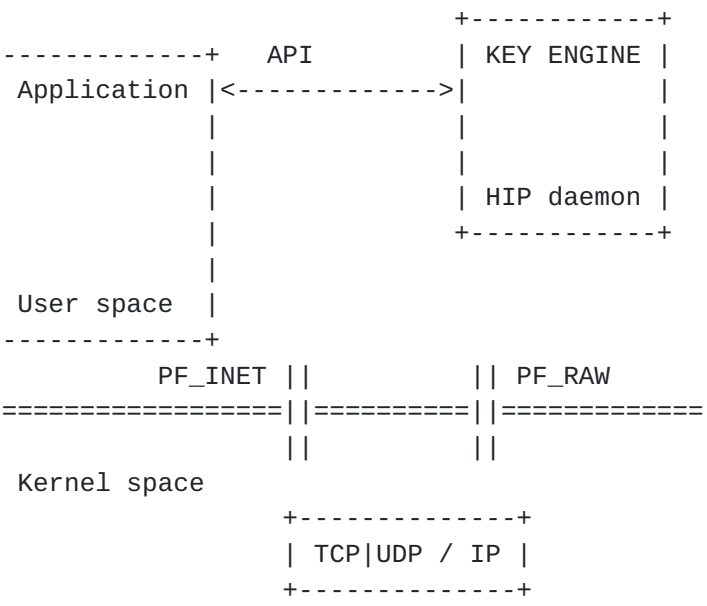


Fig 5: Example Implementation Architecture

Figure 6 depicts the key derivation, for example, when the peer receives a packet it gets the packet index, MKI, which is used for identifying the relevant master key and transforms. Then, the key derivation function, which is explained below, will generate the required keys.

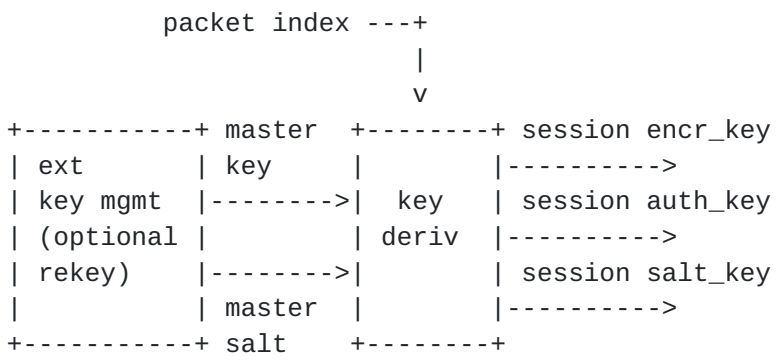


Fig 6: SRTP Key Derivation

For single key derivation (key_derivation_rate = 0), we define x for later use in calculating keys using PRF and length of PRF bit string output like shown in the following table:

X	ROC || SEQ	Usage	PRF output length n
0x00	000000000000	SRTP encryption	128 bit
0x01	000000000000	SRTP message auth.	160 bit
0x02	000000000000	SRTP salting key	112 bit
0x03	000000000000	SRTCP encryption	128 bit
0x04	000000000000	SRTCP message auth.	160 bit
0x05	000000000000	SRTCP salting key	112 bit

PRF_n (master_key, x)

For multiple key derivation (key_derivation_rate = 1,2,...2E24)
x must be calculated according to the following sequence:

r = index / key_derivation_rate
(with "/" defines r = 0 for key_derivation_rate = 0)

with index is a 48-bit concatenation of the 32 bit Roll Over Counter (ROC) and the 16 bit sequence number of the SRTP packet given in the SRTP header (ROC||SEQ)

r must be the same length like index, which results in leading zeros.

Next concatenate an 8-bit label for selecting the usage with r
key_id = <label> concatenated with r.

where <label> is one of the following>
0x00 for SRTP encryption
0x01 for SRTP message authentication
0x02 for SRTP salting key
0x03 for SRTCP encryption key
0x04 for SRTCP authentication key
0x05 for SRTCP salting key

Finally, x is calculated by performing key_id XOR master_salt, where key_id and master_salt are aligned so that their least significant bits agree (right-alignment).

7. Security Considerations

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:

DoS: 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.

MitM: HIP uses authenticated Diffie-Hellmann key exchange, which prevents the man-in-the-middle (MitM) attacks.

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

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

8. Acknowledgements

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

9. References

9.1. Normative References

[I-D.ietf-hip-base]
Moskowitz, R., Nikander, P., Jokela, P., and T. Henderson, "Host Identity Protocol", draft-ietf-hip-base-03 (work in progress), June 2005.

[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.

9.2. Informative References

[I-D.ietf-hip-esp]
Moskowitz, R., Nikander, P., and P. Jokela, "Host Identity Protocol", draft-ietf-hip-esp-00 (work in progress), June 2005.

[I-D.ietf-hip-multi6]
Tschofenig, H. and A. Nagarajan, "Comparative Analysis of Multi6 Proposals using a Locator/Identifier Split", October 2004.

[I-D.ietf-hip-sip]
Tschofenig, H., Schulzrinne, H., Henderson, T., Torvinen, V., Camarillo, G., and J. ott, "Exchanging Host Identities in SIP", October 2004.

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

Authors' Addresses

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

Email: Hannes.Tschofenig@siemens.com

Franz Muenz
University of Applied Sciences
Lurzenhof 1
Landshut, Bayern 84036
Germany

Email: franz.muenz@fh-landshut.de

Murugaraj Shanmugam
Technical University Hamburg-Harburg
Schwarzenbergstrasse 95
Harburg, Hamburg 21075
Germany

Email: murugaraj.shanmugam@tuhh.de

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.