HIPRG
Internet-Draft
Expires: January 12, 2006

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

July 11, 2005

Using SRTP transport format with HIP
draft-tschofenig-hiprg-hip-srtp-00.txt

Status of this Memo

Copyright Notice

Abstract

The Host Identity Protocol is a signaling protocol which adds another layer to the Internet model and (optionally) establishes IPsec ESP SAs to protect subsequent data traffic. HIP is an end-to-end authentication and key exchange protocol, which supports security and mobility in a commendable manner. This draft explains a Secure Real

Time Protocol (SRTP) based mechanism for transmission of user data packets, to be used with the Host Identity Protocol (HIP).

Table of Contents

1. Introduction

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. Since different sizes for the public key are possible, it uses the Host Identity Tag (HIT), which is the hash of the public key, for operational representation. The HIP header carries HIT (128 bits long), which is similar to IPV6 addresses.

Transport connections and Security Associations between the communicating HIP hosts are bound to the HITs only. 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, it uses a readdressing mechanism to update the current location of the peer, thereby supporting mobility in a seamless manner.

Session Initiation Protocol (SIP) is an application layer protocol, which is capable of establishing modifying and terminating sessions between the hosts. The SIP architecture uses URIs to uniquely identify (maps) the user agents and has various infrastructure components like proxy server, redirect server etc., to achieve

personal mobility.

SIP, when combined with RTP, can effectively handle multimedia applications. SIP can also invite participants to already existing sessions, such as multicast conferences. Media can be added to (and removed from) an existing session. SIP relies on other security protocols like TLS, IPsec, HTTP Digest mechanisms to protect the SIP traffic.

HIP base exchange [I-D.ietf-hip-base] does not describe any transport formats for user data. This document proposes extensions to HIP by exporting the relevant parameters to support other key management

scheme, like MIKEY. SRTP proposes MIKEY [RFC3830] as a key management protocol. We propose to use the same key management scheme in HIP. HIP combined with MIKEY alike scheme can be used for SRTP as a key management protocol to exchange Master Keys and create a Cryptographic Context (SRTP RFC chapter 3.2). HIP has to satisfy the requirements of SRTP (SRTP RFC chapter 7/8) for a key management protocol and has to support the appropriate cryptographic algorithms within its transform parameters.

HIP base exchange provides a mutual authentication of the hosts, but does not specify any mechanism for protecting data packets for the actual communication. [I-D.ietf-hip-esp] draft proposes a way to use IPsec ESP format with HIP. In this document, we specify the use of SRTP for protecting user data traffic after the HIP base exchange.

SRTP mandates the use of a external key management protocol (like MIKEY) 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.

This document explains the compatibility of HIP and SIP together with the new KEY management scheme. Section 3 explains the revised base exchange, and Section 4 explains the key derivation and future work.

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 used the terminology defined in [I-D.ietf-hip-base] and [RFC3261].

The term MKI refers to Master Key Identifier used in SRTP packets. It is similar to SPIs in IPsec.

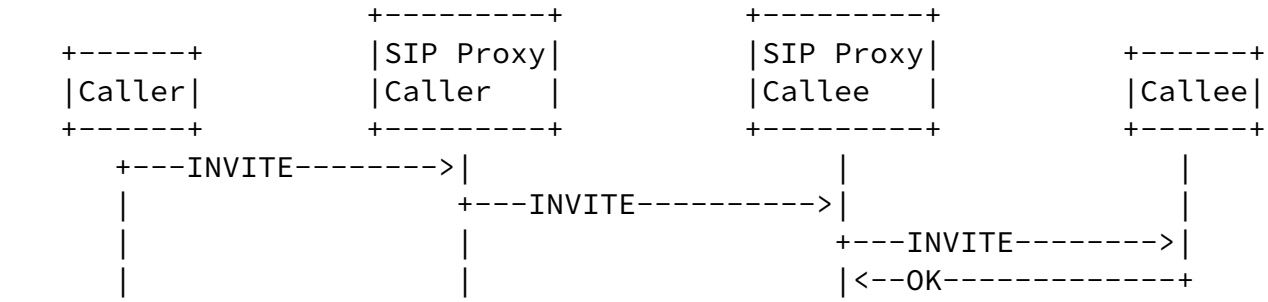
3. Message Flow

This section explains the integration of SIP and HIP. The motivation to combine HIP and SIP is defined in [I-D.ietf-hip-sip]. SIP uses URIs, which bind to an IP address or a host name. When HIP is used, SIP headers will make use of HITs instead of IPs i.e, SIP URIs will be bound to HITs. A HIT is derived from HI (public key), which identify users/hosts and IP addresses. The resolution of IP from HI/HITs can be done via DNS or other mechanisms. Also, the HI/HITs can be exchanged using SIP/SDP mechanism as desribed in [I-D.ietf-hip-sip].

Initially the caller sends an INVITE message to its proxy server, assuming that the caller is already connected. Then the caller proxy server locates the callee proxy server, possibly by performing a

particular type of DNS (Domain Name Service) lookup (DNS SRV record). DNS will return the HI/HIT of the callee together with one or more IP addresses of SIP proxies responsible for the callee. After resolution it forwards the message to a callee proxy server and adds a new entry in the SIP header (for route record routability). The callee proxy receiving the INVITE message consults a location database via a location service to resolve the HIT of the callee to current IP address. Finally, it forwards the INVITE to the callee.

There are several ways how to combine SIP messages with HIP base-exchange. SIP and HIP messages could be combined to reduce roundtrips or can be used separately.The latter will be explained in detail in this section.



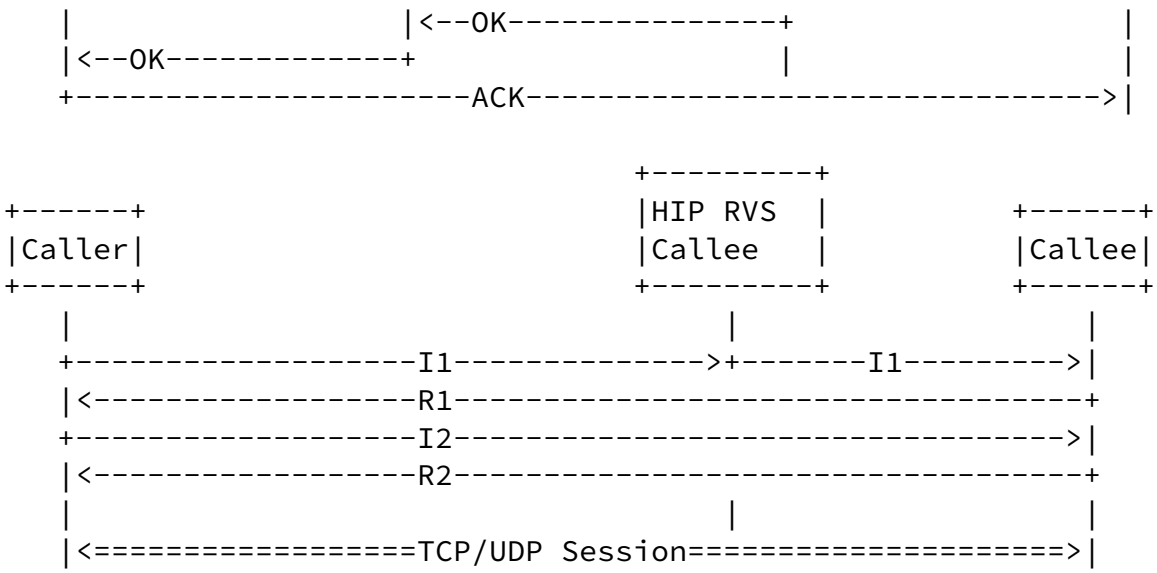


Figure 1: SIP and HIP Base Exchange

Session establishment works in known ways. First an INVITE is routed from the caller to the callee using SIP proxies. The callee then answers with 200 OK and the caller acknowledges with an ACK message directly to the callee. However in this scenario, the SDP of the SIP signalling traffic will not include any SRTP parameters (transforms), which will be decoupled and delegated to HIP. SIP only serves as a rendezvous protocol for HIP to exchange end-host IP addresses and negotiate HIP as the used end-to-end authentication and key exchange protocol.

3.1 Base Exchange

After HIP is chosen there are again two possibilities on how to proceed. Firstly HIP base-exchange may run directly between communication partners or secondly the callee might be using HIP rendezvous server which is shown in figure 1.

As explained in the previous sections, HIP allows the use of other key management protocols. Figure 2 explains how the new KEYING parameters fit into the HIP base exchange:

Initiator Responder

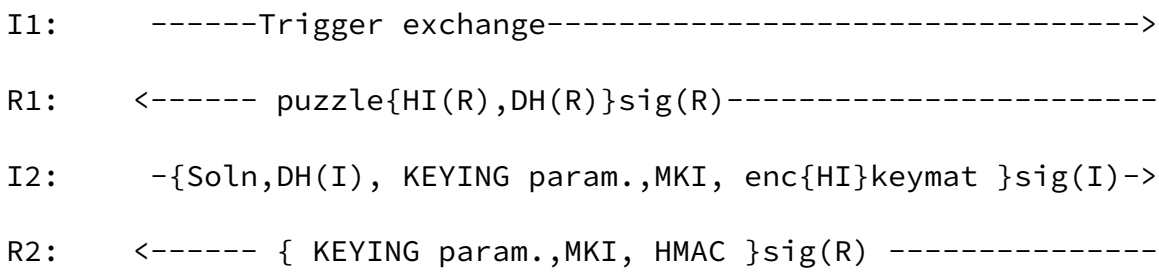


Fig 2: Base Exchange

The Initiator starts the HIP connection by sending the trigger message. This message is nothing but two IPs and HITs of the Initiator and the Responder respectively. The Responder answers with the R1 packet, the difference between the actual HIP exchange and the proposed mechanism is the removal of the cipher suites, because the transforms will be chosen via KEYING parameter. Since we have to avoid the state creation, it sends a precomputed packet.

Context id = <SSRC, destination HIT/address, destination port> This triple SHALL uniquely identiies a cryptographic context (SRTP RFC chapter 3.2.3.). This context id together with MKI will be mapped to the master key and cipher suites in KEYING management scheme to find the session keys to process the packet.

Any specific transform parameters needed for the SRTP cryptographic context will be exchanged by using SP parameter of KEYING parameter in HIP.

Master Key - derived from Diffie-Hellmann value

Master Salt - RAND in the KEYING parameter

MKI - Master Key Identifier

Upon receiving it , the Initiator solves the puzzle and sends the I2 packet with the Diffie Hellmann value and its KEYING parameter. For the explanation of KEYING parameter see above. The Initiator's HI is encrypted by the keying material derived from the master key (Diffie Hellmann value), so that the responder can also derive the same key using the negotiated cipher suites and Diffie Hellmann value to decrypt the HI. The key management and key derivation is up to the KEY scheme. The whole packet is signed by the Initiator's public key.

The Responder receives the packet verifies the solution, derives the key using the Diffie Hellmann value and the KEYING parameter, decrypts the HI, using the keying material , and verifies the Signature. The Responder derives all the encryption and authentication keys from the Initiator's master (Diffie Hellmann) and salt key (KEYING parameter RAND). The reason for this is that both the Initiator and Responder have the same key pairs for providing confidentiality for the data traffic.

Next, the Responder sends its KEYING parameter , the same time stamp, random no, the selected cipher suites and HMAC of the whole packet, the key for HMAC is derived from the Master Key. It sends its MKI to identify the incoming packet. The Initiator will check the HMAC and also the Signature to verify the integrity and authenticity of the packet. After this, the HIP association is established and both the hosts use their respective master key and it derived keys for protecting the traffic. The Master Key is 128 bit long, which can be exchanged using the Diffie Hellmann parameter.

3.2 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:

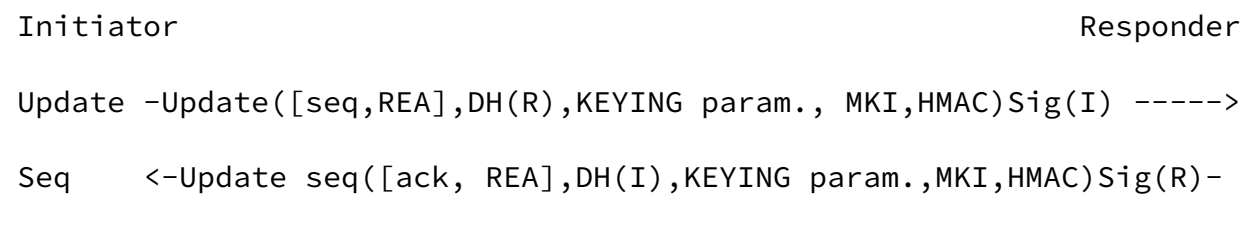


Fig 3:Rekeying mechanism

Figure 3 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. The key for HMAC has been derived from the old Master key. It also sends a new MKI value to identify the incoming packet.

The Responder chooses its Diffie Hellmann value, verifies the HMAC

and Signature. The other parameters are explained in [I-D.ietf-hip-base] draft. The Responder checks the return routability by sending

the Update seq message containing its 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.

3.3 Packet Format

This section explains the packet format for the KEYING parameter in more detail.

KEYING parameter contains

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

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

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.

MKI : It is similar to SPI i.e, to identify the Master key and also security associations.

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

session keys is derived using Master key and SP

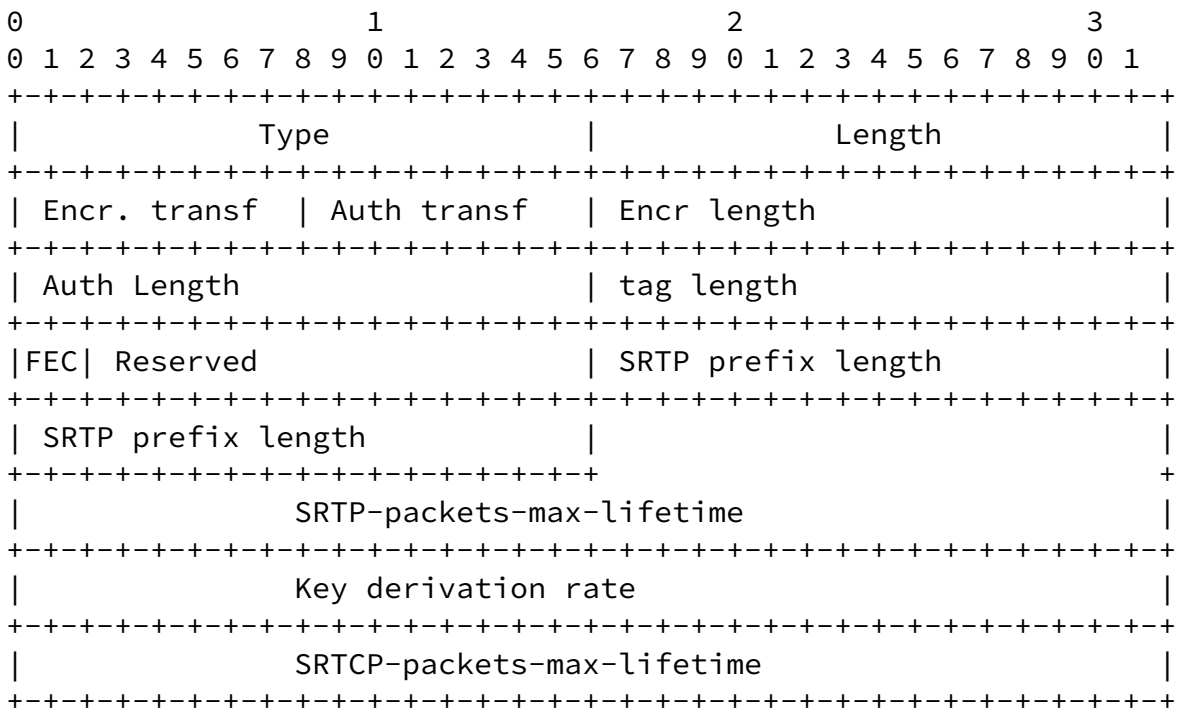


Fig 4: Key management parameters

Type: 40000 (experimental identifier range)
Length: 256 bit
Value:

Type/Meaning	Possible values
SRTP and SRTCP encr transf	see below
SRTP and SRTCP auth transf.	see below
tag length	80

```
        SRTP prefix_length                | variable (default 0)
        Key derivation PRF                | see below
        encr session key length           | 128
        auth session key length           | 160
        key derivation rate               | variable  (default 0)
        SRTP-packets-max-lifetime         | variable
        SRTCP-packets-max-lifetime        | variable
        Forward Error Control             | 2-bits
```



```
   SRTP and SRTCP encr transf. | Value
   ---------------------------------
   NULL                        |  0
   AES_CM                      |  1
   AES_f8                      |  2

   SRTP and SRTCP auth transf. | Value
   ---------------------------------
   NULL                        |  0
   HMAC-SHA1                   |  1

   Key derivation PRF          | Value
   ---------------------------------
   NULL                        |  0
   AES_CM                      |  1
```

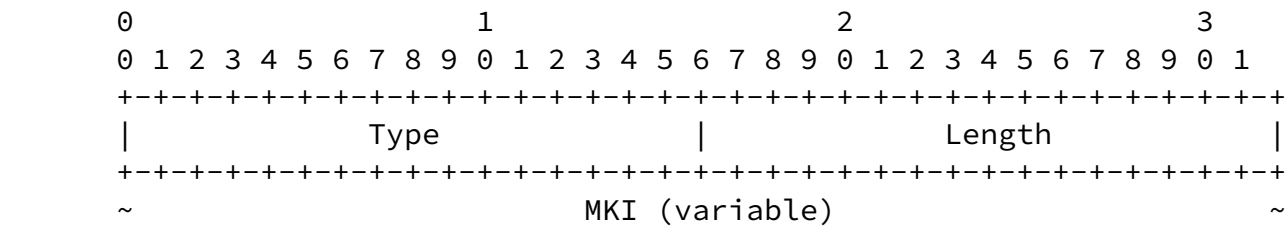


Fig 5: SRTP MKI parameter

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

4. 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:

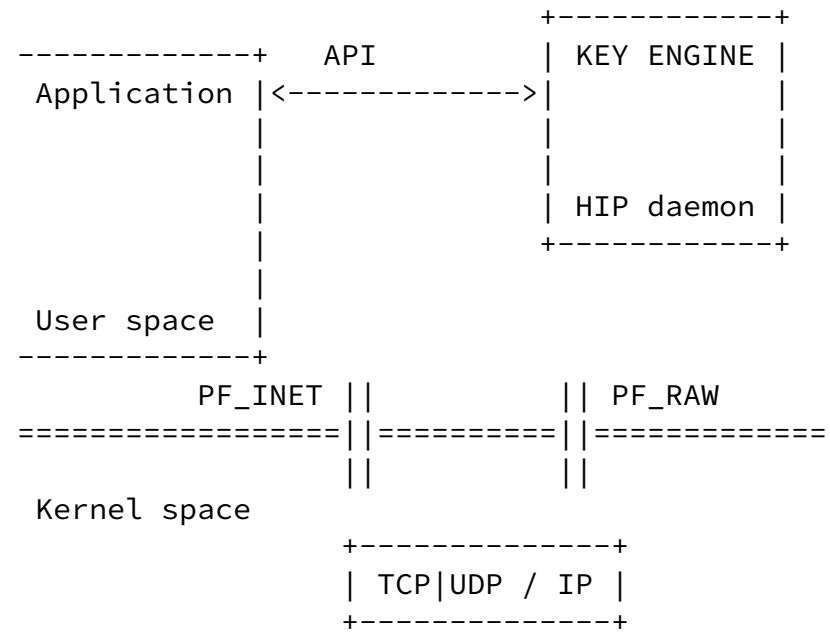


Fig 6: Example Implementation Architecture

Figure 7 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.

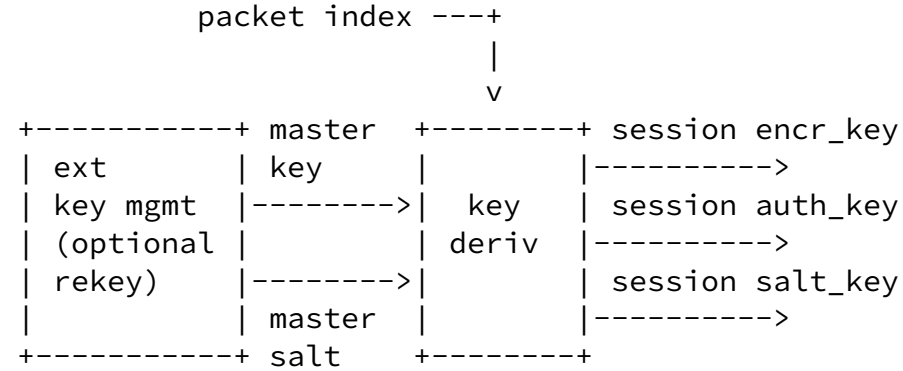


Fig 7: SRTP Key Derivation

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).

5. 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:

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.

6. References

6.1 Normative References

[I-D.ietf-hip-base]
Moskowitz, R., Nikander, P., Jokela, P., and T. Henderson, "Host Identity Protocol", draft-ietf-hip-base-02 (work in progress), February 2005.

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

6.2 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.

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

[RFC3830] Arrko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", draft-ietf-hip-base-02 (work in progress), August 2004.

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.