IETF NSIS Working Group
Internet Draft

H. Tschofenig
Siemens
H. Schulzrinne
Columbia U.

Document: draft-tschofenig-rsvp-doi-01.txt
Expires: April 2002 October 2003

RSVP Domain of Interpretation for ISAKMP

Status of this Memo

Abstract

RSVP does not provide dynamic key management for the RSVP Integrity object. It is difficult to provide security for RSVP based on standard security protocols. This draft proposes the usage of the ISAKMP protocol with a new Domain of Interpretation (DoI) and allows to establish the necessary security parameters for the RSVP Integrity object. The Integrity object protects RSVP signaling messages at the application layer and uses this DoI to dynamically establish the necessary security associations.

This document also addresses the NSIS NTLP work and protocol design implications.

Table of Contents

1. Introduction

1.1 RSVP DoI

RSVP [RFC2205] offers security based on the RSVP Integrity object as described in [RFC2747] and based on the user identity representation document [RFC3182]. Unfortunately there is still room for improvement for a number of reasons. This document tries to fix some of them, namely providing dynamic authentication and key exchange in order to create the necessary security parameters for the RSVP Integrity object. The mechanism described in this document is executed independently of the RSVP message exchange to avoid modifications to the RSVP protocol itself. With this extension administrators have the choice between manual RSVP security establishment (see [RFC2205]) and dynamic authentication and key exchange based on shared secrets, Kerberos or public key based authentication.

A detailed discussion of security properties of RSVP is referred to [Tsc03]. This document tries to follow the current NSIS working group documents with regard to their requirements and framework thoughts (see [HF+03] and [Bru03]). Additionally security threats relevant for NSIS are applicable to this work (see [TK03]).

The basic procedure for establishing an RSVP security association can be described as follows:

Whenever an RSVP signaling message has to be sent to the next RSVP aware node and no security association is already available then a new one has to be dynamically established. RSVP therefore triggers the key management daemon. The RSVP daemon then constructs a RSVP message and interacts with the security association database using some sort of API (e.g. PF_KEY [RFC2367]) to retrieve the session key and other security parameters. Maintenance (creation, deletion, rekeying, possibly dead peer detection, etc.) of the RSVP security association is accomplished by the key management daemon. KINK [TV03], which uses Kerberos, can also be used as an authentication and key exchange protocol in addition to IKE.

Note that this draft is strongly aligned with [RFC2407] and reuses the same structure and (if appropriate) the same text.

1.2 Requirements for a DOI

Within ISAKMP, a Domain of Interpretation is used to group related protocols using ISAKMP to negotiate security associations. Security protocols sharing a DOI choose security protocol and cryptographic transforms from a common namespace and share key exchange protocol identifiers. They also share a common interpretation of DOI-specific payload data content, including the Security Association and Identification payloads.

Overall, ISAKMP places the following requirements on a DOI definition:

o define the naming scheme for DOI-specific protocol identifiers
o define the interpretation for the Situation field
o define the set of applicable security policies
o define the syntax for DOI-specific SA Attributes (Phase II)
o define the syntax for DOI-specific payload contents
o define additional Key Exchange types, if needed
o define additional Notification Message types, if needed

The remainder of this document describes the instantiation of these requirements for using the RSVP Security mechanism specified in [RFC2747] to provide authentication, integrity and replay protection.

2. Terminology

This document does not introduce new terms.

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119].

3. Definition

3.1 Naming Scheme

Within ISAKMP, all DOI's must be registered with the IANA in the "Assigned Numbers" RFC [RFC3232]. The IANA Assigned Number for the RSVP DOI is TBD (TBD). Within the RSVP DOI, all well-known identifiers MUST be registered with the IANA under the RSVP DOI. Unless otherwise noted, all tables within this document refer to IANA Assigned Numbers for the RSVP DOI. See Section 5 for further information relating to the IANA registry for the RSVP DOI.

All multi-octet binary values are stored in network byte order.

3.2 Situation Definition

Within ISAKMP, the Situation provides information that can be used by the responder to make a policy determination about how to process the incoming Security Association request. For the RSVP DOI, the Situation field is a four (4) octet bitmask with the following values.

Situation	Value
SIT_IDENTITY_ONLY	0x01

3.2.1 SIT_IDENTITY_ONLY

The SIT_IDENTITY_ONLY type specifies that the security association will be identified by source identity information present in an associated Identification Payload. See Section 4.6.2 of [RFC2407] for a complete description of the various Identification types. All RSVP DOI implementations MUST support SIT_IDENTITY_ONLY by including an Identification Payload in at least one of the Phase I Oakley exchanges ([RFC2409], Section 5) and MUST abort any association setup that does not include an Identification Payload.

3.3 Security Policy Requirements

The RSVP DOI does not impose specific security policy requirements on any implementation. Host system policy issues are outside of the scope of this document.

However, the following sections touch on some of the issues that must be considered when designing an RSVP DOI implementation. This section should be considered only informational in nature.

3.3.1 Key Management Issues

It is expected that many systems choosing to implement ISAKMP will strive to provide a protected domain of execution for a combined IKE key management daemon. On protected-mode multi-user operating systems, this key management daemon will likely exist as a separate privileged process.

In such an environment, a formalized API such as PF_KEY [RFC2367] to communicate keying material (and other security relevant parameters) between the RSVP Daemon, the key management daemon and the key engine may be desirable.

3.3.1.1 Static Keying Issues

Systems that implement static keys, either for use directly by RSVP, or for authentication purposes (see [RFC2409] Section 5.4), should take steps to protect the static keying material when it is not residing in a protected memory domain or actively in use by the key engine.

Depending on the operating system and utility software installed, it may not be possible to protect the static keys once they are available to the keying engine or to the RSVP daemon, however they should not be trivially recoverable on initial system startup without having to satisfy some additional form of authentication.

3.3.1.2 Policy Issues

It is not realistic to assume that the transition to RSVP DOI will occur overnight. Incremental deployment is, however, possible particularly in intra-domain environments. Systems must be prepared to implement flexible policy lists that describe which systems they desire to speak securely with and which systems they require speak securely to them. This type of authorization behavior particularly addresses intra-domain environments where a strong trust relationship between individual RSVP nodes exists.

A minimal approach is probably a static list of IP addresses. For intra-domain communication such an approach might be sufficient in many cases due to the nature of RSVP signaling. A more flexible approach based on wildcard DNS names is given below and might simplify and reduce configuration overhead.

For inter-domain environments the authorization procedure must provide some mapping to an authorized identity for which a financial settlement between the interacting domains exist. For inter-domain environments it seems to be more appropriate not to use static lists of IP addresses. A more flexible implementation might consist of a list of wildcard DNS names (e.g. '*.foo.bar'). The wildcard DNS name would be used to match incoming or outgoing IP addresses.

The communication between end systems and the attached network is more difficult from an authorization point of view. The reader is referred to a detailed discussion in [TB+03a] and in [TB+03b]. For this version of the document RSVP DOI mainly addresses intra-domain and inter-domain communication instead of end host to network communication due to the authorization nature of QoS signaling protocols. A future version of the document might address these issues (and for non-QoS signaling applications) in an appropriate manner.

3.3.1.3 Certificate Management

Systems implementing a certificate-based authentication scheme will need a mechanism for obtaining and managing a database of certificates.

Secure DNS is to be one certificate distribution mechanism, however the pervasive availability of secure DNS zones, in the short term, is doubtful for many reasons. This is primarily applicable for inter-domain communication. For intra-domain environments secure DNS might be a reasonable choice.

The ability for RSVP nodes to import certificates that they acquire through secure, out-of-band mechanisms, is also a reasonable procedure.

However, manual certificate management should not be done so as to preclude the ability to introduce dynamic certificate discovery mechanisms and/or protocols as they become available.

In addition to certificate-based authentication and the distribution of pre-shared secrets between nodes for the purpose of authentication, Kerberos might be used. KINK [TV03] uses Kerberos and as such a trusted third party entity for authentication and key distribution. KINK replaces the first phase of IKE and represents a very efficient and fast alternative to IKE Phase I. Systems implementing the RSVP DOI SHOULD support this DOI using KINK.

3.4 RSVP DOI assigned numbers

The following sections list the Assigned Numbers for the RSVP DOI: Situation Identifiers, Protocol Identifiers, Transform Identifiers, Security Association Attribute Type Values, Labeled Domain Identifiers, ID Payload Type Values, and Notify Message Type Values.

3.4.1 RSVP DOI Security Protocol Identifier

The ISAKMP proposal syntax was specifically designed to allow for the simultaneous negotiation of multiple Phase II security protocol suites within a single negotiation. As a result, the protocol suites listed below form the set of protocols that can be negotiated at the same time. It is a host policy decision as to what protocol suites might be negotiated together.

Protocol ID	Value
RESERVED	0
PROTO_ISAKMP	1
PROTO_RSVP_Integrity	2

All other values unused.

3.4.1.1 PROTO_ISAKMP

The PROTO_ISAKMP type specifies message protection required during Phase I of the ISAKMP protocol. The specific protection mechanism used for the RSVP DOI is described in [RFC2409]. All implementations within the RSVP DOI MUST support PROTO_ISAKMP.

NB: ISAKMP reserves the value one (1) across all DOI definitions.

3.4.1.2 PROTO_RSVP_Integrity

The PROTO_RSVP_Integrity type provides the necessary parameters for the security association used in RSVP (i.e. the RSVP Integrity Object [RFC2747]). This transform provides data origin authentication, integrity protection and replay detection.

Transforms for confidentiality protection are currently not defined. Support for confidentiality protection is currently not provided although useful. Furthermore, transforms providing payload compression do not seem to be useful for a signaling protocol due to the fact that other mechanisms have been standardized which provide similar techniques in a more efficient way (see [RFC2961]).

3.4.2 RSVP ISAKMP Transform Identifiers

As part of an ISAKMP Phase I negotiation, the initiator's choice of Key Exchange offerings is made using some host system policy description. The actual selection of Key Exchange mechanism is made using the standard ISAKMP Proposal Payload. The following table lists the defined ISAKMP Phase I Transform Identifiers for the Proposal Payload for the RSVP DOI.

Transform	Value
RESERVED	0
KEY_IKE	1

Within the ISAKMP and RSVP DOI framework it is possible to define key establishment protocols other than IKE (Oakley). Previous versions of this document defined types both for manual keying and for schemes based on use of a generic Key Distribution Center (KDC). These identifiers have been removed from the current document.

Extension of the RSVP DOI to include values for additional non-Oakley key establishment protocols (such as the Group Key Management Protocol (GKMP) [RFC2093]) is under consideration.

3.4.2.1 KEY_IKE

The KEY_IKE type specifies the hybrid ISAKMP/Oakley Diffie-Hellman key exchange (IKE) as defined in the [RFC2409] document. All implementations within the RSVP DOI MUST support KEY_IKE.

3.4.3 RSVP Integrity Transform Identifiers

The RSVP Integrity Object provides data origin authentication, integrity, and replay detection. It consists of the following fields:

- Flags

The Handshake Flag is the only defined flag and is used to synchronize sequence numbers if the communication gets out-of-sync. The separately defined mechanism is not required. Hence the Flags are set to zero.

- Key Identifier

The Key Identifier selects the key used for verification of the Keyed Message Digest field. Its length is fixed with 48-bit. According to [RFC2747] is the generation of this Key Identifier field mostly a local decision.

The 32-bit SPI field will be used to select the security association for verifying the keyed message digest. Hence the first 32 bit of the 48-bit Key Identifier are the SPI and the last 16 bit are set to zero.

- Sequence Number

[RFC2747] defines the sequence number used by the RSVP Integrity object as a 64-bit value for which the starting value can be selected arbitrarily. To prevent the need to communicate the sequence number the sequence number MUST start with zero for usage with this protocol.

- Keyed Message Digest

This field contains the keyed message digest and is a variable length field.

The following table lists the defined RSVP Integrity Transform Identifiers for the ISAKMP Proposal Payload for the RSVP DOI.

Note: the Authentication Algorithm attribute MUST be specified to identify the appropriate protection suite. For example, AUTH_MD5 can best be thought of as a generic AH transform using MD5. To request the HMAC construction with AUTH, one specifies the AUTH_MD5 transform ID along with the Authentication Algorithm attribute set to HMAC-MD5. This is shown using the "Auth(HMAC-MD5)" notation in the following sections.

Transform ID	Value

RESERVED	0-1
AUTH_MD5	2
AUTH_SHA	3
AUTH_DES	4

Note: all mandatory-to-implement algorithms are listed as "MUST" implement (e.g. AUTH_MD5) in the following sections. All other algorithms are optional and MAY be implemented in any particular implementation.

3.4.3.1 AUTH_MD5

The AUTH_MD5 type specifies a generic AUTH transform using MD5. The actual protection suite is determined in concert with an associated SA attribute list. A generic MD5 transform is currently undefined.

All implementations within the RSVP DOI MUST support AUTH_MD5 along with the Auth(HMAC-MD5) attribute. HMAC-MD5 is described in [RFC2104].

Use of AUTH_MD5 with any other Authentication Algorithm attribute value is currently undefined.

3.4.3.2 AUTH_SHA

The AUTH_SHA type specifies a generic AUTH transform using SHA-1. The actual protection suite is determined in concert with an associated SA attribute list. A generic SHA transform is currently undefined.

All implementations within the RSVP DOI MUST support AUTH_SHA along with the Auth(HMAC-SHA) attribute. SHA-1 is described in [SHA1].

Use of AH_SHA with any other Authentication Algorithm attribute value is currently undefined.

3.4.3.3 AUTH_DES

The AUTH_DES type specifies a generic AUTH transform using DES. The actual protection suite is determined in concert with an associated SA attribute list. A generic DES transform is currently undefined.

The RSVP DOI defines AUTH_DES along with the Auth(DES-MAC) attribute to be a DES-MAC transform. Implementations are not required to support this mode.

Use of AUTH_DES with any other Authentication Algorithm attribute value is currently undefined.

3.5 RSVP Security Association Attributes

The following SA attribute definitions are used in Phase II of an IKE negotiation. Attribute types can be either Basic (B) or Variable-Length (V). Encoding of these attributes is defined in the base ISAKMP specification.

Attributes described as basic MUST NOT be encoded as variable. Variable length attributes MAY be encoded as basic attributes if their value can fit into two octets. See [RFC2409] for further information on attribute encoding in the RSVP DOI. All restrictions listed in [RFC2409] also apply to the RSVP DOI.

Attribute Types

class	value	type
SA Life Type	1	B
SA Life Duration	2	V
Group Description	3	B
Authentication Algorithm	4	B

Class Values

SA Life Type
SA Duration

Specifies the time-to-live for the overall security association. When the SA expires, all keys negotiated under the association must be renegotiated. The life type values are:

RESERVED	0
seconds	1
kilobytes	2

Values 3-61439 are reserved to IANA. Values 61440-65535 are for private use. For a given Life Type, the value of the Life Duration attribute defines the actual length of the component lifetime -- either a number of seconds, or a number of Kbytes that can be protected.

If unspecified, the default value shall be assumed to be 28800 seconds (8 hours).

An SA Life Duration attribute MUST always follow an SA Life Type which describes the units of duration.

Group Description

Specifies the Oakley Group to be used in a PFS QM negotiation. For a list of supported values, see Appendix A of [RFC2409].

Authentication Algorithm	
RESERVED	0
HMAC-MD5	1
HMAC-SHA	2
DES-MAC	3
KPDK	4

Values 5-61439 are reserved to IANA. Values 61440-65535 are for private use.

The default value for the Auth Algorithm is HMAC-MD5.

3.5.1 Required Attribute Support

To ensure basic interoperability, all implementations MUST be prepared to negotiate all of the following attributes.

SA Life Type
SA Duration
Auth Algorithm

3.5.2 Attribute Parsing Requirement (Lifetime)

To allow for flexible semantics, the RSVP DOI requires that a conforming ISAKMP implementation MUST correctly parse an attribute list that contains multiple instances of the same attribute class, so long as the different attribute entries do not conflict with one another. Currently, the only attributes which requires this treatment are Life Type and Duration.

To see why this is important, the following example shows the binary encoding of a four entry attribute list that specifies an SA Lifetime of either 100MB or 24 hours. (See Section 3.3 of [RFC2408] for a complete description of the attribute encoding format.)

Attribute #1:
0x80010001 (AF = 1, type = SA Life Type, value = seconds)

Attribute #2:
0x00020004 (AF = 0, type = SA Duration, length = 4 bytes)
0x00015180 (value = 0x15180 = 86400 seconds = 24 hours)

Attribute #3:
0x80010002 (AF = 1, type = SA Life Type, value = KB)

```
Attribute #4:
  0x00020004  (AF = 0, type = SA Duration, length = 4 bytes)
  0x000186A0  (value = 0x186A0 = 100000KB = 100MB)
```

If conflicting attributes are detected, an ATTRIBUTES-NOT-SUPPORTED Notification Payload SHOULD be returned and the security association setup MUST be aborted.

Note that this is the same treatment as suggested in [RFC2407].

3.5.3 Attribute Negotiation

If an implementation receives a defined RSVP DOI attribute (or attribute value) which it does not support, an ATTRIBUTES-NOT-SUPPORT SHOULD be sent and the security association setup MUST be aborted, unless the attribute value is in the reserved range.

If an implementation receives an attribute value in the reserved range, an implementation MAY choose to continue based on local policy.

3.5.4 Lifetime Notification

When an initiator offers an SA lifetime greater than what the responder desires based on their local policy, the responder has three choices:

1) fail the negotiation entirely;

2) complete the negotiation but use a shorter lifetime than what was offered;

3) complete the negotiation and send an advisory notification to the initiator indicating the responder's true lifetime. The choice of what the responder actually does is implementation specific and/or based on local policy.

To ensure interoperability in the latter case, the RSVP DOI requires the following only when the responder wishes to notify the initiator: if the initiator offers an SA lifetime longer than the responder is willing to accept, the responder SHOULD include an ISAKMP Notification Payload in the exchange that includes the responder's SA payload. Section 3.6.2.1 defines the payload layout for the RESPONDER-LIFETIME Notification Message type which MUST be used for this purpose.

3.6 RSVP DOI Payload Content

The following sections describe those ISAKMP payloads whose data representations are dependent on the applicable DOI.

RSVP DOI does not require any additional payloads. In particular it is not required to exchange Traffic Selector attributes within IKE Phase II as part of the Identification payload. The attributes used in the Phase I Identification payload are sufficient.

3.6.1 Identification Payload Content

The initiator of the negotiation is identified using the Identification Payload. The responder SHOULD use the initiator's identity to determine the correct security policy for creating SAs.

During Phase 1, the ID port and protocol fields MUST be set to zero or to the UDP port that the RSVP DOI is running on. If an implementation receives any other values, this MUST be treated as an error and the negotiation MUST be aborted.

The following diagram illustrates the content of the Identification Payload.

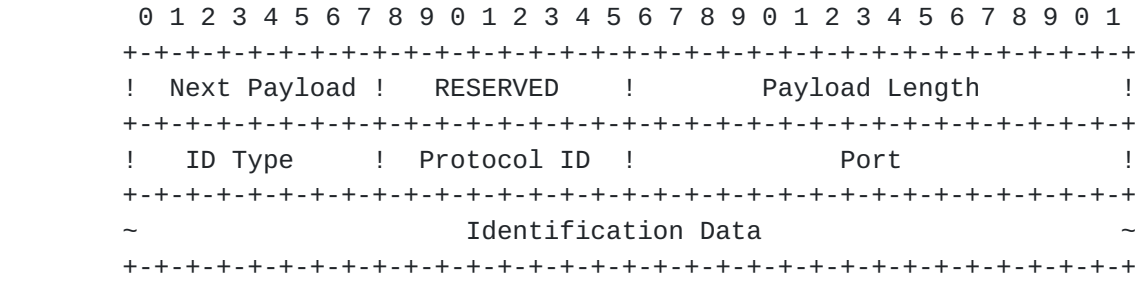

Figure 2: Identification Payload Format

The Identification Payload fields are defined as follows:

- Next Payload (1 octet) - Identifier for the type of the next payload in the message. If the current payload is the last in the message, this field will be zero (0).

- RESERVED (1 octet) - Unused, must be zero (0).

- Payload Length (2 octets) - Length, in octets, of the identification data, including the generic header.

- Identification Type (1 octet) - Value describing the identity information found in the Identification Data field.

- Protocol ID (1 octet) - Value specifying an associated Transport Layer Protocol ID (e.g. UDP/TCP). Value zero means

that the Protocol ID field should be ignored. If raw IP is used then this value is set to zero. RSVP also allows UDP to be used.

- Port (2 octets) - Value specifying an associated port. A value of zero means that the Port field should be ignored. If raw IP is used with RSVP then the concept of a port is not used.

- Identification Data (variable length) - Value, as indicated by the Identification Type.

  The legal Identification Type field values in Phase 1 are as defined in [IPSDOI]. The table lists the assigned values for the Identification Type field found in the Identification Payload.

ID Type	Value
RESERVED	0
ID_IPV4_ADDR	1
ID_FQDN	2
ID_USER_FQDN	3
ID_IPV4_ADDR_SUBNET	4
ID_IPV6_ADDR	5
ID_IPV6_ADDR_SUBNET	6
ID_IPV4_ADDR_RANGE	7
ID_IPV6_ADDR_RANGE	8
ID_DER_ASN1_DN	9
ID_DER_ASN1_GN	10
ID_KEY_ID	11

The values of the individual Identification Types are described in Section 5.6.2.1 of [RFC2407].

3.6.2 RSVP DOI Notify Message Types

ISAKMP defines two blocks of Notify Message codes, one for errors and one for status messages. ISAKMP also allocates a portion of each block for private use within a DOI. The RSVP DOI defines the following private message types for its own use.

Notify Messages - Error Types	Value
RESERVED	8192

Notify Messages - Status Types	Value
RESPONDER-LIFETIME	24576
INITIAL-CONTACT	24578

Notification Status Messages MUST be sent under the protection of an ISAKMP SA: either as a payload in the last Main Mode exchange; in a separate Informational Exchange after Main Mode or Aggressive Mode processing is complete; or as a payload in any Quick Mode exchange. These messages MUST NOT be sent in Aggressive Mode exchange, since Aggressive Mode does not provide the necessary protection to bind the Notify Status Message to the exchange.

Note that a Notify payload is fully protected only in Quick Mode, where the entire payload is included in the HASH(n) digest. In Main Mode, while the notify payload is encrypted, it is not currently included in the HASH(n) digests. As a result, an active substitution attack on the Main Mode ciphertext could cause the notify status message type to be corrupted. (This is true, in general, for the last message of any Main Mode exchange.) While the risk is small, a corrupt notify message might cause the receiver to abort the entire negotiation thinking that the sender encountered a fatal error. Implementation Note: the ISAKMP protocol does not guarantee delivery of Notification Status messages when sent in an ISAKMP Informational Exchange. To ensure receipt of any particular message, the sender SHOULD include a Notification Payload in a defined Main Mode or Quick Mode exchange which is protected by a retransmission timer.

3.6.2.1 RESPONDER-LIFETIME

The RESPONDER-LIFETIME status message may be used to communicate the SA lifetime chosen by the responder.

When present, the Notification Payload MUST have the following format:

- Payload Length - set to length of payload + size of data (var)
- DOI - set to RSVP DOI (TBD)
- Protocol ID - set to selected Protocol ID from chosen SA
- SPI Size - set to either sixteen (16) (two eight-octet ISAKMP cookies) or four (4) (one SPI)
- Notify Message Type - set to RESPONDER-LIFETIME (Section 4.6.3)
- SPI - set to the two ISAKMP cookies or to the sender's inbound SPI
- Notification Data - contains an ISAKMP attribute list with the responder's actual SA lifetime(s)

Implementation Note: saying that the Notification Data field contains an attribute list is equivalent to saying that the Notification Data field has zero length and the Notification Payload has an associated attribute list.

3.6.2.2 INITIAL-CONTACT

The INITIAL-CONTACT status message may be used when one side wishes to inform the other that this is the first SA being established with the remote system. The receiver of this Notification Message might then elect to delete any existing SA's it has for the sending system under the assumption that the sending system has rebooted and no longer has access to the original SA's and their associated keying material. When used, the content of the Notification Data field SHOULD be null (i.e. the Payload Length should be set to the fixed length of Notification Payload).

When present, the Notification Payload MUST have the following format:

- Payload Length - set to length of payload + size of data (0)
- DOI - set to RSVP DOI (TBD)
- Protocol ID - set to selected Protocol ID from chosen SA
- SPI Size - set to sixteen (16) (two eight-octet ISAKMP cookies)
- Notify Message Type - set to INITIAL-CONTACT
- SPI - set to the two ISAKMP cookies
- Notification Data - <not included>

4. Security Considerations

This entire memo pertains to the Internet Key Exchange protocol [RFC2409], which combines ISAKMP [RFC2408] and Oakley [RFC2412] to provide for the derivation of cryptographic keying material in a secure manner. Specific discussion of the various security protocols and transforms identified in this document can be found in the indicated documents.

It is important to mention that this document is based on the assumption that two RSVP nodes know each other already before they exchange RSVP messages (and therefore knows which security parameter to select). Unfortunately this is not true for all scenarios. Thus in these cases where this assumption does not hold it is not possible to use this mechanisms directly. This assumption mainly addresses the nature of PATH alike signaling messages (i.e. messages which are addressed to the end host and carry a Router Alert Option [RFC2113]).

Security requirements, threats, framework issues and authorization aspects are found in separate documents (see [TK03], [HF+03], [TB+03a] and [TB+03b]).

5. IANA Considerations

This document contains many "magic" numbers to be maintained by the IANA.

A future version of the document will contain a list of the required numbers.

6. Key Derivation

The RSVP Integrity object requires keying material with can either be procedure by IKE or KINK or other authentication and key exchange protocols supporting the Domain of Interpretation framework. For IKE the key derivation procedure defined in Section 5.5 of [RFC2409] is used. For KINK the key derivation procedure described in Section 8 of [TV03] is applicable.

7. Open Issues

A number of open issues have been identified. Some of these issues result from the fact that reusing of RSVP within NSIS is under investigation.

- This document tries to reuse the security of RSVP (namely the RSVP Integrity object) without modifications. Currently RSVP only supports data origin authentication, integrity protection and replay detection based on the RSVP Integrity object. Steve Bellovin expressed interest in adding support for confidentiality protection at the 55th IETF. Confidentiality protection is not included in this document since it would require a new RSVP security object. For this version of the document no parameter negotiation for confidentiality protection is therefore provided.

- The Keyed Message Digest field is variable in length but must be a multiple of four octets. The truncated HMAC-SHA-1-96 or the HMAC-MD5-96 does not work with this restriction. Furthermore, it might be desirable specify other integrity algorithms such as RIPEMD-160.

- Currently no compression profiles are defined for usage with RSVP. It seems that no such profile is required.

- The REPLAY-STATUS notification is not required since replay protection is mandatory. However, in cases of multicast and in case of selective object protection between non-neighboring RSVP nodes it might need to be introduced. This version of the document does not address the security of multicast RSVP signaling messages.

- The Identification payload contains the same values as [RFC2407]. It remains for further study whether it might be possible to limit the list.

8. Normative References

[TK03] H. Tschofenig and D. Kroeselberg: "Security Threats for NSIS", Internet Draft, Internet Engineering Task Force, June 2003. Work in progress.

[Tsc03] H. Tschofenig: "RSVP Security Properties", Internet Draft, Internet Engineering Task Force, March 2003. Work in progress.

[RFC2408] D. Maughan, M. Schertler, M. Schneider and J. Turner: "Internet Security Association and Key Managment Protocol (ISAKMP)", RFC 2408, November 1998.

[RFC2407] D. Piper: "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.

[RFC2205] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation protocol (RSVP) -- version 1 functional specification", RFC 2205, Internet Engineering Task Force, Sept. 1997.

[TV03] M. Thomas and J. Vilhuber: "Kerberized Internet Negotiation of Keys (KINK)", Internet Draft, Internet Engineering Task Force, Jan. 2003.

[Bru03] M. Brunner: "Requirements for QoS signaling protocols", Internet Draft, Internet Engineering Task Force, August 2003. Work in progress.

[RFC2409] D. Harkins and D. Carrel: "The Internet Key Exchange (IKE)", RFC 2409, Internet Engineering Task Force, Nov. 1998. IKE

[RFC2104] H. Krawczyk, M. Bellare, and R. Canetti: "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, March 1996.

[SHA1] NIST, FIPS PUB 180-1: Secure Hash Standard,
April 1995.
http://csrc.nist.gov/fips/fip180-1.txt (ascii)
http://csrc.nist.gov/fips/fip180-1.ps (postscript)

[RFC2408] D. Maughan, M. Schertler, M. Schneider and J. Turner: "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, Internet Engineering Task Force, November 1998.

[RFC2412] H. Orman: "The OAKLEY Key Determination Protocol", RFC 2412, Internet Engineering Task Force, November 1998.

[RFC3232] J. Reynolds: "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", RFC 3232, Internet Engineering Task Force, Jan. 2002.

[RFC2747] F. Baker, B. Lindell and M. Talwar: "RSVP Cryptographic Authentication", RC 2747, Internet Engineering Task Force, January, 2000.

[RFC2119] S. Bradner: "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, Internet Engineering Task Force, March 1997.

9. Informative References

[RFC2367] D. McDonald, C. Metz, and B. Phan: "PF_KEY key management API, version 2", RFC 2367, Internet Engineering Task Force, July 1998.

[TB+03a] H. Tschofenig, M. Buechli, S. Van den Bosch and H. Schulzrinne: "NSIS Authentication, Authorization and Accounting Issues", Internet Draft, Internet Engineering Task Force, March 2003. Work in progress.

[TB+03b] H. Tschofenig, M. Buechli, S. Van den Bosch, H. Schulzrinne and T. Chen: "QoS NSLP Authorization Issues", Internet Draft, Internet Engineering Task Force, June 2003. Work in progress.

[RFC2093] H. Harney and C. Muckenhirn: "Group Key Management Protocol (GKMP) Specification", RFC 2093, Internet Engineering Task Force, July 1997.

[HF+03] R. Hancock, I. Freytsis, G. Karagiannis, J. Loughney and S. Van den Bosch: "Next Steps in Signaling: Framework", Internet Draft, Internet Engineering Task Force, September 2003. Work in progress.

[RFC2113] D. Katz: "IP router alert option", RFC 2113, Internet Engineering Task Force, Feb. 1997.

[RFC3182] S. Yadav, R. Yavatkar, R. Pabbati, P. Ford, T. Moore, S. Herzog and R., Hess: "Identity Representation for RSVP", RFC 3182, Internet Engineering Task Force, October, 2001.

[RFC2961] L. Berger, D. Gan, G. Swallow, P. Pan, F. Tommasi, and S. Molendini: "RSVP refresh overhead reduction extensions," RFC 2961, Internet Engineering Task Force, Apr. 2001.

10. Acknowledgments

This document is largely based on [RFC2407] and hence we would like to thank the author Derrell Piper and the IETF members, which provided input to RFC 2407, for their work.

Additionally, we would like to thank Jorge Cuellar and Gerald Volkmann for their comments to the draft.

11. Author's Addresses

Hannes Tschofenig
Siemens AG
Otto-Hahn-Ring 6
81739 Munich
Germany
EMail: Hannes.Tschofenig@siemens.com

Henning Schulzrinne
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, NY 10027
USA
EMail: schulzrinne@cs.columbia.edu