NSIS
Internet Draft   Hannes Tschofenig
Document: Siemens
draft-ietf-nsis-rsvp-sec-properties-02.txt
Expires: December 2003   June 2003

RSVP Security Properties
<draft-ietf-nsis-rsvp-sec-properties-02.txt>

Status of this Memo

Abstract

This document summarizes the security properties of RSVP. The goal of this analysis is to benefit from previous work done with RSVP and to capture the knowledge about past activities.

Table of Contents

1. Introduction..2
2. Terminology and Architectural Assumptions......................3
3. Overview..5
   3.1 The RSVP INTEGRITY Object...................................5
   3.2 Security Associations.......................................6
   3.3 RSVP Key Management Assumptions.............................7
   3.4 Identity Representation.....................................7
   3.5 RSVP Integrity Handshake...................................12
4. Detailed Security Property Discussion..........................13
   4.1 Discussed Network Topology.................................13
   4.2 Host/Router..13
   4.3 User to PEP/PDP..17
   4.4 Communication between RSVP aware routers..................25
5. Miscellaneous Issues...26
   5.1 First Hop Issue..26
   5.2 Next-Hop Problem...27
   5.3 Last-Hop Issue...29
   5.4 RSVP and IPsec protected data traffic......................30
   5.5 End-to-End Security Issues and RSVP........................32
   5.6 IPsec protection of RSVP signaling messages................32
   5.7 Authorization..33
6. Conclusions..34
7. Security Considerations..35
8. IANA considerations..35
9. Acknowledgments..35
10. Normative References..38
11. Informative References..39
Author's Contact Information......................................42
Full Copyright Statement..42

1. Introduction

As the work of the NSIS working group has begun there are also concerns about security and its implication for the design of a signaling protocol. In order to understand the security properties and available options of RSVP a number of documents have to be read. This document summarize the security properties of RSVP and is part of the overall process of analyzing other signaling protocols and to learn from their design considerations. This document should also provide a starting point for further discussions.

The content of this document is organized as follows:

Section 3 provides an overview of the security mechanisms provided by RSVP including the INTEGRITY object, a description of the identity representation within the POLICY_DATA object (i.e. user authentication) and the RSVP Integrity Handshake mechanism.

Section 4 provides a more detailed discussion of the used mechanism and tries to describe the mechanisms provided in detail.

Finally a number of miscellaneous issues are described which address first-hop, next-hop and last-hop issues. Furthermore the problem of IPsec security protection of data traffic and RSVP signaling message is discussed.

2. Terminology and Architectural Assumptions

This section describes some important terms and explains some architectural assumptions:

- Chain-of-Trust

The security mechanisms supported by RSVP [RFC2747] heavily relies on optional hop-by-hop protection using the built-in INTEGRITY object. Hop-by-hop security with the INTEGRITY object inside the RSVP message thereby refers to the protection between RSVP supporting network elements. Additionally there is the notion of policy aware network elements that additionally understand the POLICY_DATA element within the RSVP message. Since this element also includes an INTEGRITY object there is an additional hop-by-hop security mechanism that provides security between policy aware nodes. Policy ignorant nodes are not affected by the inclusion of this object in the POLICY_DATA element since they do not try to interpret it.

To protect signaling messages that are possibly modified by each RSVP router along the path it must be assumed that each incoming request is authenticated, integrity and replay protected. This provides protection against unauthorized nodes injecting bogus messages. Furthermore each RSVP-router is assumed to behave in the expected manner. Outgoing messages transmitted to the next hop network element experience protection according RSVP security processing.

Using the above described mechanisms a chain-of-trust is created whereby a signaling message transmitted by router A via router B and received by router C is supposed to be secure if router A and B and router B and C share a security association and all routers behave expectedly. Hence router C trusts router A although router C does not have a direct security association with router A. We can therefore conclude that the protection achieved with this hop-by-hop security for the chain-of-trust is as good as the weakest link in the chain.

If one router is malicious (for example because an adversary has control over this router) then it can arbitrarily modify messages and cause unexpected behavior and mount a number of attacks not only restricted to QoS signaling. Additionally it must be mentioned that

some protocols demand more protection than others (this depends between which nodes these protocols are executed). For example edge devices, where end-users are attached, may more likely be attacked in comparison to the more secure core network of a service provider. In some cases a network service provider may choose not to use the RSVP provided security mechanisms inside the core network because a different security protection is deployed.

Section 6 of [RFC2750] mentions the term chain-of-trust in the context of RSVP integrity protection. In Section 6 of [HH01] the same term is used in the context of user authentication with the INTEGRITY object inside the POLICY_DATA element. Unfortunately the term is not explained in detail and the assumption is not clearly specified.

- Host and User Authentication

The presence of the RSVP protection and a separate user identity representation leads to the fact that both user- and the host-identities are used for RSVP protection. Therefore user and host based security is investigated separately because of the different authentication mechanisms provided. To avoid confusion about the different concepts Section 3.4 will describe the concept of user authentication in more detail.

- Key Management

For most of the security associations required for the protection of RSVP signaling messages it is assumed that they are already available and hence key management was done in advance. There is however an exception with the support for Kerberos. Using Kerberos an entity is able to distribute a session key used for RSVP signaling protection.

- RSVP INTEGRITY and POLICY_DATA INTEGRITY Object

RSVP uses the INTEGRITY object in two places of the message. The first usage is in the RSVP message itself and covers the entire RSVP message as defined in [RFC2747] whereas the latter is included in the POLICY_DATA object and defined in [RFC2750]. In order to differentiate the two objects regarding their scope of protection the two terms RSVP INTEGRITY and POLICY_DATA INTEGRITY object are used. The data structure of the two objects however is the same.

- Hop vs. Peer

In the past there was considerable discussion about the terminology of a nodes that are addressed by RSVP. In particular two favorites have used: hop and peer. This document uses the term hop which is different to an IP hop. Two neighboring RSVP nodes communicating with

each other are not necessarily neighboring IP nodes (i.e. one IP hop away).

3. Overview

This section describes the security mechanisms provided by RSVP. Although the usage of IPsec is mentioned in Section 10 of [RFC2747] the security mechanisms primarily envisioned for RSVP are described.

3.1 The RSVP INTEGRITY Object

The RSVP INTEGRITY object is the major component of the RSVP security protection. This object is used to provide integrity and replay protect the content of the signaling message between two RSVP participating router. Furthermore the RSVP INTEGRITY object provides data origin authentication. The attributes of the object are briefly described:

- Flags field

The Handshake Flag is the only defined flag and is used to synchronize sequence numbers if the communication gets out-of-sync (i.e. for a restarting host to recover the most recent sequence number). Setting this flag to one indicates that the sender is willing to respond to an Integrity Challenge message. This flag can therefore be seen as a capability negotiation transmitted within each INTEGRITY object.

- Key Identifier

The Key Identifier selects the key used for verification of the Keyed Message Digest field and hence must be unique for the sender. Its length is fixed with 48-bit. The generation of this Key Identifier field is mostly a decision of the local host. [RFC2747] describes this field as a combination of an address, the sending interface and a key number. We assume that the Key Identifier is simply a (keyed) hash value computed over a number of fields with the requirement to be unique if more than one security association is used in parallel between two hosts (i.e. as it is the case with security association that have overlapping lifetimes). A receiving system uniquely identifies a security association based on the Key Identifier and the sender's IP address. The sender's IP address may be obtained from the RSVP_HOP object or from the source IP address of the packet if the RSVP_HOP object is not present. The sender uses the outgoing interface to determine which security association to use. The term outgoing interface might be confusing. The sender selects the security association based on the receiver's IP address (of the next RSVP capable router). To determine which node is the next capable

RSVP router is not further specified and is likely to be statically configured.

- Sequence Number

The sequence number used by the INTEGRITY object is 64-bits in length and the starting value can be selected arbitrarily. The length of the sequence number field was chosen to avoid exhaustion during the lifetime of a security association as stated in Section 3 of [RFC2747]. In order for the receiver to distinguish between a new and a replayed sequence number each value must be monotonically increasing modulo 2^64. We assume that the first sequence number seen (i.e. the starting sequence number) is stored somewhere. The modulo-operation is required because the starting sequence number may be an arbitrary number. The receiver therefore only accepts packets with a sequence number larger (modulo 2^64) than the previous packet. As explained in [RFC2747] this process is started by handshaking and agreeing on an initial sequence number. If no such handshaking is available then the initial sequence number must be part of the establishment of the security association.

The generation and storage of sequence numbers is an important step in preventing replay attacks and is largely determined by the capabilities of the system in presence of system crashes, failures and restarts. Section 3 of [RFC2747] explains some of the most important considerations.

- Keyed Message Digest

The Keyed Message Digest is an RSVP built-in security mechanism used to provide integrity protection of the signaling messages. Prior to computing the value for the Keyed Message Digest field the Keyed Message Digest field itself must be set to zero and a keyed hash computed over the entire RSVP packet. The Keyed Message Digest field is variable in length but must be a multiple of four octets. If HMAC-MD5 is used then the output value is 16 bytes long. The keyed hash function HMAC-MD5 [RFC2104] is required for a RSVP implementation as noted in Section 1 of [RFC2747]. Hash algorithms other than MD5 [RFC1321] like SHA [SHA] may also be supported.

The key used for computing this Keyed Message Digest may be obtained from the pre-shared secret which is either manually distributed or the result of a key management protocol. No key management protocol, however, is specified to create the desired security associations.

3.2 Security Associations

Different attributes are stored for security associations of sending and receiving systems (i.e. unidirectional security associations).

The sending system needs to maintain the following attributes in such a security association [RFC2747]:

- Authentication algorithm and algorithm mode
- Key
- Key Lifetime
- Sending Interface
- Latest sequence number (sent with this key identifier)

The receiving system has to store the following fields:

- Authentication algorithm and algorithm mode
- Key
- Key Lifetime
- Source address of the sending system
- List of last n sequence numbers (received with this key identifier)

Note that the security associations need to have additional fields to indicate their state. It is necessary to have an overlapping lifetime of security associations to avoid interrupting an ongoing communication because of expired security associations. During such a period of overlapping lifetime it is necessary to authenticate either one or both active keys. As mentioned in [RFC2747] a sender and a receiver might have multiple active keys simultaneously.
If more than one algorithm is supported then the algorithm used must be specified for a security association.

3.3 RSVP Key Management Assumptions

[RFC2205] assumes that security associations are already available. Manual key distribution must be provided by an implementation as noted in Section 5.2 of [RFC2747]. Manual key distribution however has different requirements to a key storage
\- - a simple plaintext
ASCII file may be sufficient in some cases. If multiple security associations with different lifetimes should be supported at the same time then a key engine would be more appropriate. Further security requirements listed in Section 5.2 of [RFC2747] are the following:

- The manual deletion of security associations must be supported.
- The key storage should persist a system restart.
- Each key must be assigned a specific lifetime and a specific Key Identifier.

3.4 Identity Representation

In addition to host-based authentication with the INTEGRITY object inside the RSVP message user-based authentication is available as introduced with [RFC2750]. Section 2 of [RFC3182] stated that

"Providing policy based admission control mechanism based on user identities or application is one of the prime requirements." To identify the user or the application, a policy element called AUTH_DATA, which is contained in the POLICY_DATA object, is created by the RSVP daemon at the userÆs host and transmitted inside the RSVP message. The structure of the POLICY_DATA element is described in [RFC2750]. Network nodes like the PDP then use the information contained in the AUTH_DATA element to authenticate the user and to allow policy-based admission control to be executed. As mentioned in [RFC3182] the policy element is processed and the policy decision point replaces the old element with a new one for forwarding to the next hop router.

A detailed description of the POLICY_DATA element can be found in [RFC2750]. The attributes contained in the authentication data policy element AUTH_DATA, which is defined in [RFC3182], are briefly explained in this Section. Figure 1 shows the abstract structure of the RSVP message with its security relevant objects and the scope of protection. The RSVP INTEGRITY object (outer object) covers the entire RSVP message whereas the POLICY_DATA INTEGRITY object only covers objects within the POLICY_DATA element.

```
+--------------------------------------------------------+
| RSVP Message                                           |
+--------------------------------------------------------+
| INTEGRITY +-------------------------------------------+|
| Object    |POLICY_DATA Object                         ||
|           +-------------------------------------------+|
|           | INTEGRITY +------------------------------+||
|           | Object    | AUTH_DATA Object             |||
|           |           +------------------------------+||
|           |           | Various Authentication       |||
|           |           | Attributes                   |||
|           |           +------------------------------+||
|           +-------------------------------------------+|
+--------------------------------------------------------+
```

Figure 1: Security relevant Objects and Elements within the RSVP message

The AUTH_DATA object contains information for identifying users and applications together with credentials for those identities. The main purpose of those identities seems to be the usage for policy based admission control and not for authentication and key management. As noted in Section 6.1 of [RFC3182] an RSVP may contain more than one POLICY_DATA object and each of them may contain more than one AUTH_DATA object. As indicated in the Figure above and in [RFC3182] one AUTH_DATA object contains more than one authentication attribute.

A typical configuration for a Kerberos-based user authentication includes at least the Policy Locator and an attribute containing the Kerberos session ticket.

Successful user authentication is the basis for executing policy-based admission control. Additionally other information such as time-of-day, application type, location information, group membership etc. may be relevant for a policy.

The following attributes are defined for the usage in the AUTH_DATA object:

a) Policy Locator

The policy locator string that is a X.500 distinguished name (DN) used to locate the user and/or application specific policy information. The following types of X.500 DNs are listed:

- ASCII_DN
- UNICODE_DN
- ASCII_DN_ENCRYPT
- UNICODE_DN_ENCRYPT

The first two types are the ASCII and the Unicode representation of the user or application DN identity. The two "encrypted" distinguished name types are either encrypted with the Kerberos session key or with the private key of the userÆs digital certificate (i.e. digitally signed). The term encrypted together with a digital signature is easy to misconceive. If user identity confidentiality shall be provided then the policy locator has to be encrypted with the public key of the recipient. How to obtain this public key is not described in the document. Such an issue may be specified in a concrete architecture where RSVP is used.

b) Credentials

Two cryptographic credentials are currently defined for a user: Authentication with Kerberos V5 [RFC1510], and authentication with the help of digital signatures based on X.509 [RFC2495] and PGP [RFC2440]. The following list contains all defined credential types currently available and defined in [RFC3182]:

Credential Type	Description
ASCII_ID	User or application identity encoded as an ASCII string
UNICODE_ID	User or application identity encoded as an Unicode string
KERBEROS_TKT	Kerberos V5 session ticket
X509_V3_CERT	X.509 V3 certificate
PGP_CERT	PGP certificate

Table 1: Credentials Supported in RSVP

The first two credentials only contain a plaintext string and therefore they do not provide cryptographic user authentication. These plaintext strings may be used to identify applications, which are included for policy-based admission control. Note that these plain-text identifiers may, however, be protected if either the RSVP INTEGRITY and/or the INTEGRITY object of the POLICY_DATA element is present. Note that the two INTEGRITY objects can terminate at different entities depending on the network structure. The digital signature may also provide protection of application identifiers. A protected application identity (and the entire content of the POLICY_DATA element) cannot be modified as long as no policy ignorant nodes are used in between.

A Kerberos session ticket, as previously mentioned, is the ticket of a Kerberos AP_REQ message [RFC1510] without the Authenticator. Normally, the AP_REQ message is used by a client to authenticate to a server. The INTEGRITY object (e.g. of the POLICY_DATA element) provides the functionality of the Kerberos Authenticator, namely replay protection and shows that the user was able to retrieve the session key following the Kerberos protocol. This is, however, only the case if the Kerberos session was used for the keyed message digest field of the INTEGRITY object. Section 7 of [RFC2747] discusses some issues for establishment of keys for the INTEGRITY object. The establishment of the security association for the RSVP INTEGRITY object with the inclusion of the Kerberos Ticket within the AUTH_DATA element may be complicated by the fact that the ticket can be decrypted by node B whereas the RSVP INTEGRITY object terminates at a different host C. The Kerberos session ticket contains, among many other fields, the session key. The Policy Locator may also be

encrypted with the same session key. The protocol steps that need to be executed to obtain such a Kerberos service ticket are not described in [RFC3182] and may involve several roundtrips depending on many Kerberos related factors. The Kerberos ticket does not need to be included in every RSVP message as an optimisation as described in Section 7.1 of [RFC2747]. Thus the receiver must store the received service ticket. If the lifetime of the ticket is expired then a new service ticket must be sent. If the receiver lost his state information (because of a crash or restart) then he may transmit an Integrity Challenge message to force the sender to re-transmit a new service ticket.

If either the X.509 V3 or the PGP certificate is included in the policy element then a digital signature must be added. The digital signature computed over the entire AUTH_DATA object provides authentication and integrity protection. The SubType of the digital signature authentication attribute is set to zero before computing the digital signature. Whether or not a guarantee of freshness with the replay protection (either timestamps or sequence numbers) is provided by the digital signature is an open issue as discussed in Section 4.3.

c) Digital Signature

The digital signature computed over the data of the AUTH_DATA object must be the last attribute. The algorithm used to compute the digital signature depends on the authentication mode listed in the credential. This is only partially true since for example PGP again allows different algorithms to be used for computing a digital signature. The algorithm identifier used for computing the digital signature is not included in the certificate itself. The algorithm identifier included in the certificate only serves the purpose to allow the verification of the signature computed by the certificate authority (except for the case of self-signed certificates).

d) Policy Error Object

The Policy Error Object is used in the case of a failure of the policy based admission control or other credential verification. Currently available error messages allow to notify if the credentials are expired (EXPIRED_CREDENTIALS), if the authorization process disallowed the resource request (INSUFFICIENT_PRIVILEGES) and if the given set of credentials is not supported (UNSUPPORTED_CREDENTIAL_TYPE). The latter error message returned by the network allows the user's host to discover the type of credentials supported. Particularly for mobile environments this might be quite inefficient. Furthermore it is unlikely that a user supports different types of credentials. The purpose of the error

message IDENTITY_CHANGED is unclear. The protection of the error message is not discussed in [RFC3182].

3.5 RSVP Integrity Handshake

The Integrity Handshake is a protocol that was designed to allow a crashed or restarted host to obtain the latest valid challenge value stored at the receiving host. Due to the absent key management it must be guaranteed that two messages do not use the same sequence number with the same key. A host stores the latest sequence number of a cryptographically verified message. An adversary can replay eavesdropped packets if the crashed host has lost its sequence numbers. A signaling message from the real sender with a new sequence number would therefore allow the crashed host to update the sequence number field and prevent further replays. Hence if there is a steady flow of RSVP protected messages between the two hosts an attacker may find it difficult to inject old messages since new authenticated messages with high sequence numbers arrive and get stored immediately.

The following description explains the details of the RSVP Integrity Handshake that is started by Node A after recovering from a synchronization failure:

```
                  Integrity Challenge
              (1) Message (including
 +----------+     a Cookie)           +----------+
 |          |-------------------------->|          |
 |  Node A  |                           |  Node B  |
 |          |<--------------------------|          |
 +----------+     Integrity Response  +----------+
              (2) Message (including
                  the Cookie and the
                  INTEGRITY object)
```

Figure 2: RSVP Integrity Handshake

The details of the messages are described below:

```
CHALLENGE= (Key Identifier, Challenge Cookie)
Integrity Challenge Message:=(Common Header, CHALLENGE)
Integrity Response Message:=(Common Header, INTEGRITY, CHALLENGE)
```

The "Challenge Cookie" is suggested to be a MD5 hash of a local secret and a timestamp [RFC2747].

The Integrity Challenge message is not protected with an INTEGRITY object as show in the protocol flow above. As explained in Section 10

of [RFC2747] this was done to avoid problems in situations where both communication parties do not have a valid starting sequence number.

It is recommended to use the RSVP Integrity Handshake protocol although it is not mandatory (since it may not be needed in all network environments).

4. Detailed Security Property Discussion

The purpose of this section is to describe the security protection of the RSVP provided mechanisms individually for authentication, authorization, integrity and replay protection, user identity confidentiality, confidentiality of the signaling messages.

4.1 Discussed Network Topology

The main purpose of this paragraph is to show the basic interface of a simple RSVP network architecture. The architecture below assumes that there is only a very single domain and that two routers are RSVP and policy aware. These assumptions are relaxed in the individual paragraphs as necessary. Layer 2 devices between the clients and their corresponding first hop routers are not shown. Other network elements like a Kerberos Key Distribution Center and for example an LDAP server where the PDP retrieves his policies are also omitted. The security of various interfaces to the individual servers (KDC, PDP, etc.) depends very much on the security policy of a specific network service provider.

```
                        +--------+
                        |Policy  |
                        |Decision|
                   +----+Point   +---+
                   |    +--------+   |
                   |                 |
                   |                 |
                   |                 |
 +------+       +-+----+        +---+--+          +------+
 |Client|       |Router|        |Router|          |Client|
 |  A   +-------+  1   +--------+  2   +----------+  B   |
 +------+       +------+        +------+          +------+
```

Figure 3: Simple RSVP Architecture

4.2 Host/Router

When talking about authentication in RSVP it is very important to make a distinction between user and host authentication of the

signaling messages. By using the RSVP INTEGRITY object the host is authenticated while credentials inside the AUTH_DATA object can be used to authenticate the user. In this section the focus is on host authentication whereas the next section covers user authentication.

a) Authentication

We use the term host authentication above since the selection of the security association is bound to the hostÆs IP address as mentioned in Section 3.1 and 3.2. Depending on the key management protocol used to create this security association and the identity used it is also possible to bind a user identity to this security association. Since the key management protocol is not specified it is difficult to evaluate this part and hence we speak about data origin authentication based on the hostÆs identity for RSVP INTEGRITY objects. The fact that the host identity is used for selecting the security association has already been described in Section 3.1.

Data origin authentication is provided with the keyed hash value computed over the entire RSVP message excluding the keyed message digest field itself. The security association used between the userÆs host and the first-hop router is, as previously mentioned, not established by RSVP and must therefore be available before the signaling is started.

- Kerberos for the RSVP INTEGRITY object

As described in Section 7 of [RFC2747] Kerberos may be used to create the key for the RSVP INTEGRITY object. How to learn the principal name (and realm information) of the other node is outside the scope of [RFC2747]. Section 4.2.1 of [RFC2747] states that the required identities can be obtained statically or dynamically via a directory service or DHCP. [HA01] describes a way to distribute principal and realm information via DNS which can be used for this purpose (assuming that the FQDN or the IP address of the other node is known for which this information is desired). It is only required to encapsulate the Kerberos ticket inside the policy element. It is furthermore mentioned that Kerberos tickets with expired lifetime must not be used and the initiator is responsible for requesting and exchanging a new service ticket before expiration.

RSVP multicast processing in combination with Kerberos requires additional thoughts:

Section 7 of [RFC2747] states that in the multicast case all receivers must share a single key with the Kerberos Authentication Server i.e. a single principal used for all receivers). From a personal discussion with Rodney Hess it seems that there is currently

no other solution available in the context of Kerberos. Multicast handling therefore leaves some questions open in this context.

In case that one entity crashed the established security association is lost and therefore the other node must retransmit the service ticket. The crashed entity can use an Integrity Challenge message to request a new Kerberos ticket to be retransmitted by the other node. If a node receives such a request then a reply message must be returned.

b) Integrity Protection

Integrity protection between the userÆs host and the first hop router is based on the RSVP INTEGRITY object. HMAC-MD5 is the preferred although other keyed hash functions may also be used within the RSVP INTEGRITY object. In any case both communicating entities must have a security association which indicates the algorithm to use. This may be however difficult since there is no negotiation protocol defined to agree on a specific algorithm. Hence it is very likely that HMAC-MD5 is the only usable algorithm for the RSVP INTEGRITY object if RSVP is used in a mobile environment and only in local environments it may be useful to switch to a different keyed hash algorithm. The other possible alternative is that every implementation must support the most important keyed hash algorithms for example MD5, SHA-1, RIPEMD-160 etc. HMAC-MD5 was mainly chosen because of the performance characteristics. The weaknesses of MD5 [DBP96] are known and described in [Dob96]. Other algorithms like SHA-1 [SHA] and RIPEMD-160 [DBP96] provide better security properties.

c) Replay Protection

The main mechanism used for replay protection in RSVP is based on sequence numbers whereby the sequence number is included in the RSVP INTEGRITY object. The properties of this sequence number mechanism are described in Section 3.1. The fact that the receiver stores a list of sequence numbers is an indicator for a window mechanism. This somehow conflicts with the requirement that the receiver only has to store the highest number given in Section 3 of [RFC2747]. We assume that this is a typo. Section 4.1 of [RFC2747] gives a few comments about the out-of-order delivery and the ability of an implementation to specify the replay window.

- Integrity Handshake

The mechanism of the Integrity Handshake is explained in Section 3.5. The Cookie value is suggested to be hash of a local secret and a timestamp. The Cookie value is not verified by the receiver. The mechanism used by the Integrity Handshake is a simple

Challenge/Response message which assumes that the key shared between the two hosts survives the crash. If the security association is however dynamically created then this assumption may not be true.

In Section 10 of [RFC2747] the authors note that an adversary can create faked Integrity Handshake message including challenge cookies. Subsequently he would store the received response. Later he tries to replay these responses while a responder recovers from a crash or restart. If this replayed Integrity Response value is valid and has a lower sequence number than actually used then this value is stored at the recovering host. In order for this attack to be successful the adversary must either have collected a large number of challenge/response value pairs or the adversary "discovered" the cookie generation mechanism (for example by knowing the local secret). The collection of Challenge/Response pairs is even more difficult since they depend on the Cookie value, on sequence number included in the response message and on the shared key which is used by the INTEGRITY object.

d) Confidentiality

Confidentiality is not considered to be a security requirement for RSVP. Hence it is not supported by RSVP.

e) Authorization

The task of authorization consists of two subcategories: Network access authorization and RSVP request authorization. Access authorization is provided when a node is authenticated to the network e.g. using EAP [RFC2284] in combination with AAA protocols (for example using RADIUS [RFC2865] or DIAMETER [CA+02]). Issues related to network access authentication and authorization are outside the scope of RSVP.

The second authorization refers to RSVP itself. Depending on the network configuration
- the router either forwards the received RSVP request to the policy decision point e.g. by using COPS (see [RFC2748] and [RFC2749]) and to request admission control procedure to be executed or
- the router supports the functionality of a PDP and therefore there is no need to forward the request or
- the router may already be configured with the appropriate policy information to decide locally whether to grant this request or not.

Based on the result of the admission control the request may be granted or rejected. Information about the resource requesting entity must be available to provide policy-based admission control.

f) Performance

The computation of the keyed message digest for a RSVP INTEGRITY object does not represent a performance problem. The protection of signaling messages is usually not a problem since these messages are transmitted at a low rate. Even a high number of messages does not cause performance problems for a RSVP routers due to the efficiency of the keyed message digest routine.

Dynamic key management, which is computationally more demanding, is more important for scalability. Since RSVP does not specify a particular key exchange protocol to be used it is difficult to estimate the effort to create the required security associations. Furthermore the number of key exchanges to be triggered depends on security policy issues like lifetime of a security association, required security properties of the key exchange protocol, authentication mode used by the key exchange protocol etc. In a stationary environment with a single administrative domain the manual security association distribution may be acceptable and provides the best performance characteristics. In a mobile environment asymmetric authentication methods are likely to be used with a key exchange protocol and some sort of certificate verification needs to be supported.

4.3 User to PEP/PDP

As noted in the previous section both user and host based authentication is supported by RSVP. Using RSVP, a user may authenticate to the first hop router or to the PDP as specified in [RFC2747] depending on the infrastructure provided by the network domain or on the architecture used (e.g. the integration of RSVP and Kerberos V5 into the Windows 2000 Operating System [MADS01]). Another architecture where RSVP is tightly integrated is the one specified by the PacketCable organization. The interested reader is referred to [PKTSEC] for a discussion of their security architecture.

a) Authentication

When a user sends a RSVP PATH or RESV message then this message may include some information to authenticate the user. [RFC3182] describes how user and application information is embedded into the RSVP message (AUTH_DATA object) and how to protect it. A router receiving such a message can use this information to authenticate the client and forward the user/application information to the policy decision point (PDP). Optionally the PDP itself can authenticate the user, which is described in the next section. In order to be able to authenticate the user, to verify the integrity and to check for replays the entire POLICY_DATA element has to be forwarded from the router to the PDP e.g. by including the element into a COPS message.

It is assumed that the INTEGRITY object within the POLICY_DATA element is sent to the PDP along with all other attributes although not clearly specified in [RFC3182].

Certificate Verification

Using the policy element as described in [RFC3182] it is not possible to provide a certificate revocation list or other information to proof the validity of the certificate inside the policy element. A specific mechanism for certificate verification is not discussed in [RFC3182] and hence a number of them can be used for this purpose. For certificate verification the network element (a router or the policy decision point), which has to authenticate the user, could frequently download certificate revocation lists or should use a protocol like the Online Certificate Status Protocol (OCSP) [RFC2560] and the Simple Certificate Validation Protocol (SCVP) [MHHF01] to determine the current status of a digital certificate.

User Authentication to the PDP

This alternative authentication procedure uses the PDP to authenticate the user instead of the first hop router. In Section 4.2.1 in [RFC3182] the choice is given for the user to either obtain a session ticket for the next hop router or for the PDP. As noted in the same Section the identity of the PDP or the next hop router is statically configured or dynamically retrieved. Subsequently user authentication to the PDP is considered.

Kerberos-based Authentication to the PDP

If Kerberos is used to authenticate the user then first a session ticket for the PDP needs to be requested. If the user roams between different routers in the same administrative domain then he does not need to request a new service ticket since the PDP is likely to be used by most or all first-hop routers within the same administrative domain. This is different if a session ticket for a router has to be obtained and authentication to a router is required. The router therefore plays a passive role of forwarding the request only to the PDP and executing the policy decision returned by the PDP.

Appendix B describes one example of user-to-PDP authentication.

User authentication with the policy element only provides unilateral authentication where the client authenticates to the router or to the PDP. If a RSVP message is sent to the userÆs host and public keyed based authentication is used then the message does not contain a certificate and digital signature. Hence no mutual authentication can be assumed. In case of Kerberos mutual authentication may be accomplished if the PDP or the router transmits a policy element with

an INTEGRITY object computed with the session key retrieved from the Kerberos ticket or if the Kerberos ticket included in the policy element is also used for the RSVP INTEGRITY object as described in Section 4.2. This procedure only works if a previous message was transmitted from the end host to the network and such key is already established. [RFC3182] does not discuss this issue and therefore there is no particular requirement dealing with transmitting network specific credentials back to the end-user's host.

b) Integrity Protection

The integrity protection of the RSVP message and the POLICY_DATA element are protected separately as shown in Figure 1. In case of a policy ignorant node along the path the RSVP INTEGRITY object and the INTEGRITY object inside the policy element terminate at different nodes. Basically the same is true for the credentials of the user if they are verified at the policy decision point instead of the first hop router.

- Kerberos

If Kerberos is used to authenticate the user to the first hop router then the session key included in the Kerberos ticket may be used to compute the INTEGRITY object of the policy element. It is the keyed message digest that provides the authentication. The existence of the Kerberos service ticket inside the AUTH_DATA object does not provide authentication and a guarantee of freshness for the receiving host. Authentication and guarantee of freshness is provided by the keyed hash value of the INTEGRITY object inside the POLICY_DATA element. The user thereby shows that he actively participated in the Kerberos protocol and that he was able to obtain the session key to compute the keyed message digest. The Authenticator used in the Kerberos V5 protocol provides similar functionality but replay protection is based on timestamps (or based on sequence number if the optional seq-number field inside the Authenticator is used for KRB_PRIV/KRB_SAFE messages as described in Section 5.3.2 of [RFC1510]).

- Digital Signature

If public key based authentication is provided then user authentication is accomplished with the digital signature. As explained in Section 3.3.3 of [RFC3182] the DIGITAL_SIGNATURE attribute must be the last attribute in the AUTH_DATA object and the digital signature covers the entire AUTH_DATA object. Which hash algorithm and public key algorithm is used for the digital signature computation is described in [RFC2440] in case of PGP. In case of X.509 credentials the situation is more complex since different mechanisms like CMS [RFC2630] or PKCS#7 [RFC2315] may be used for the

digitally signing the message element. X.509 only provides the standard for the certificate layout which seems to provide insufficient information for this purpose. Therefore X.509 certificates are supported for example by CMS and PKCS#7. [RFC3182], however, does not make any statements about the usage of CMS and PKCS#7. Currently there is no support for CMS or PKCS#7 described in [RFC3182], which provides more than only public key based authentication (e.g. CRL distribution, key transport, key agreement, etc.). Furthermore the usage of PGP in RSVP is vague since there are different versions of PGP (including OpenPGP [RFC2440]) and there has been no indication which version should be used.

Supporting public key based mechanisms in RSVP might increase the risks of denial of service attacks. Additionally the large processing, memory and bandwidth utilization should be considered. Fragmentation might also be an issue here.

If the INTEGRITY object is not included in the POLICY_DATA element or not sent to the PDP then we have to make the following observation:

a) For the digital signature case only the replay protection provided by the digital signature algorithm can be used. It is however not clear whether this usage was anticipated or not. Hence we might assume that the replay protection is based on the availability of RSVP INTEGRITY object used with a security association that is established by other means.

b) Including only the Kerberos session ticket is insufficient since freshness is not provided (since the Kerberos Authenticator is missing). Obviously there is no guarantee that the user actually followed the Kerberos protocol and was able to decrypt the received TGS_REP (or in rare cases the AS_REP if a session ticket is requested with the initial AS_REQ).

c) Replay Protection

Figure 4 shows the interfaces relevant for replay protection of signaling messages in a more complicated architecture. The client therefore uses the policy data element with PEP2 since PEP1 is not policy aware. The interfaces between the client and the PEP1 and between the PEP1 and PEP2 are protected with the RSVP INTEGRITY object. The link between the PEP2 and the PDP is protected for example by using the COPS built-in INTEGRITY object. The dotted line between the Client and the PDP indicates the protection provided by the AUTH_DATA element which has no RSVP INTEGRITY object included.

```
                        AUTH_DATA                      +----+
   +- - - - - - - - - - - - - - - - - - - - - - - - - -+PDP +-+
                                                       +----+ |
   |                                                          |
                                                              |
   |                                                 COPS     |
                                                     INTEGRITY|
   |                                                          |
                                                              |
   |                                                          |
+--+---+   RSVP INTEGRITY  +----+    RSVP INTEGRITY    +----+ |
|Client+-------------------+PEP1+----------------------+PEP2+-+
+--+---+                   +----+                      +-+--+
   |                                                     |
   +-----------------------------------------------------+
                    POLICY_DATA INTEGRITY
```

Figure 4: Replay Protection

Host authentication with the RSVP INTEGRITY object and user authentication with the INTEGRITY object inside the POLICY_DATA element both use the same replay mechanism. The length of the Sequence Number field, sequence number rollover and the Integrity Handshake is already explained in Section 3.1.

Section 9 in [RFC3182] states "RSVP INTEGRITY object is used to protect the policy object containing user identity information from security (replay) attacks.". Using public key based authentication RSVP based replay protection is not supported since the digital signature does not cover the POLICY_DATA INTEGRITY object with its Sequence Number field. The digital signature covers the entire AUTH_DATA object only.

The usage of public key cryptography within the AUTH_DATA object complicates replay protection. Digital signature computation with PGP is described in [PGP] and in [RFC2440]. The data structure preceding the signed message digest includes information about the message digest algorithm used and a 32-bit timestamp when the signature was created ("Signature creation time"). The timestamp is included in the computation of the message digest. The IETF standardized OpenPGP version [RFC2440] contains more information and describes the different hash algorithms (MD2, MD5, SHA-1, RIPEMD-160) provided. [RFC3182] does not make any statements whether the "Signature creation time" field is used for replay protection. Using timestamps for replay protection requires different synchronization mechanisms in case of clock-screws. Traditionally "loosely" synchronized clocks are assumed in those cases but also requires specifying a replay-window.

If the "Signature creation time" is not used for replay protection then a malicious policy ignorant node can use this weakness to replace the AUTH_DATA object without destroying the digital signature. It is therefore assumed that replay protection of the user credentials is not considered as an important security requirement since the hop-by-hop processing of the RSVP message protects the message against modification by an adversary between two communicating nodes.

The lifetime of the Kerberos ticket is based on the fields starttime and endtime of the EncTicketPart structure of the ticket as described in Section 5.3.1 of [RFC1510]. Since the ticket is created by the KDC located at the network of the verifying entity it is not difficult to have the clocks roughly synchronized for the purpose of lifetime verification. Additional information about clock-synchronization and Kerberos can be found at [DG96].

If the lifetime of the Kerberos ticket expires then a new ticket must be requested and used. Rekeying is implemented with this procedure.

d) (User Identity) Confidentiality

This section discusses privacy protection of identity information transmitted inside the policy element. Especially user identity confidentiality is of interest because there is no built-in RSVP mechanism for encrypting the POLICY_DATA object or the AUTH_DATA elements. Encryption of one of the attributes inside the AUTH_DATA element - of the POLICY_LOCATOR attribute is discussed.

To protect the users privacy it is important not to reveal the users identity to an adversary located between the userÆs host and the first-hop router (e.g. on a wireless link). User identities should furthermore not be transmitted outside the domain of the visited network provider i.e. the user identity information inside the policy data element should be removed or modified by the PDP to prevent revealing information to other (non-authorized) entities along the signaling path. It is not possible (with the offered mechanisms) to hide the user identity in such a way that it is not visible to the first policy aware RSVP node (or to the attached network in general).

The ASCII or Unicode distinguished name of user or application inside the POLICY_LOCATOR attribute of the AUTH_DATA element may be encrypted as specified in Section 3.3.1 of [RFC3182]. The user (or application) identity is then encrypted with either the Kerberos session key or with the private key in case of public key based authentication. Since the private key is used we usually speak of a digital signature which can be verified by everyone possessing the public key. Since the certificate with the public key is included in

the message itself this is no obstacle. Furthermore the included certificate provides enough identity information for an eavesdropper together with the additional (unencrypted) information provided in the RSVP message. Hence the possibility of encrypting the policy locator in case of public key based authentication is less obvious. To encrypt the identities using asymmetric cryptography the userÆs host must be able to somehow retrieve the public key of the entity verifying the policy element (i.e. the first policy aware router or the PDP). Currently no such mechanism is defined in [RFC3182].

The algorithm used to encrypt the POLICY_LOCATOR with the Kerberos session key is assumed to be the same as the one used for encrypting the service ticket. The information about the used algorithm is available in the etype field of the EncryptedData ASN.1 encoded message part. Section 6.3 of [RFC1510] lists the supported algorithms. [Rae01] defines new encryption algorithms (Rijndael, Serpent, and Twofish).

Evaluating user identity confidentiality requires also looking at protocols executed outside of RSVP (for example to look at the Kerberos protocol). The ticket included in the CREDENTIAL attribute may provide user identity protection by not including the optional cname attribute inside the unencrypted part of the Ticket. Since the Authenticator is not transmitted with the RSVP message the cname and the crealm of the unencrypted part of the Authenticator are not revealed. In order for the user to request the Kerberos session ticket, for inclusion in the CREDENTIAL attribute, the Kerberos protocol exchange must be executed. Then the Authenticator sent with the TGS_REQ reveals the identity of the user. The AS_REQ must also include the user identity to allow the Kerberos Authentication Server to respond with an AS_REP message that is encrypted with the user's secret key. Using Kerberos, it is therefore only possible not to reveal content of the encrypted policy locator, which is only useful if this value differs from the Kerberos principal name. Hence using Kerberos it is not "entirely" possible to provide user identity confidentiality.

It is important to note that information stored in the policy element may be changed by a policy aware router or by the policy decision point. Which parts are changed depends upon whether multicast or unicast is used, how the policy server reacts, where the user is authenticated and whether he needs to be re-authenticated in other network nodes etc. Hence user and application specific information can leak after the messages leave the first hop within the network where the user's host is attached. As mentioned at the beginning of this Section this information leakage is assumed to be intentional.

e) Authorization

Additional to the description of the authorization steps of the Host/Router interface, user based authorization is added with the policy element providing user credentials. The inclusion of user and application specific information enables policy-based admission control with special user policies that are likely to be stored at a dedicated server. Hence a Policy Decision Point can query for example a LDAP server for a service level agreement stating the amount of resources a certain user is allowed to request. Additional to the user identity information group membership and other non-security related information may contribute to the evaluation of the final policy decision. If the user is not registered to the currently attached domain then there is the question of how much information the home domain of the user is willing to exchange. This also impacts the user's privacy policy. In general the user may not want to distribute much of his policy information. Furthermore the missing standardized authorization data format may create interoperability problems when exchanging policy information. Hence we can assume that the policy decision point may use information from an initial authentication and key agreement protocol which may already required cross-realm communication with the user's home domain to only assume that the home domain knows the user and that the user is entitled to roam and to be able to forward accounting messages to this domain. This represents the traditional subscriber based accounting scenario. Non-traditional or alternative means of access might be deployed in the near future that do not require the any type of inter-domain communication.

Additional discussions are required to determine the expected authorization procedures. [TB+03a] and [TB+03b] discuss authorization issues for QoS signaling protocols. Furthermore a number of mobililty implications for the policy handling in RSVP are described in [Tho02].

f) Performance

If Kerberos is used for user authentication then a Kerberos ticket must be included in the CREDENTIAL Section of the AUTH_DATA element. The Kerberos ticket has a size larger than 500 bytes but only needs to be sent once since a performance optimization allows the session key to be cached as noted in Section 7.1 of [RFC2747]. It is assumed that subsequent RSVP messages only include the POLICY_DATA INTEGRITY object with a keyed message digest that uses the Kerberos session key. This however assumes that the security association required for the POLICY_DATA INTEGRITY object is created after (or modified) to allow the selection of the correct key. Otherwise it difficult to say which identifier is used to index the security association.

When Kerberos is used as an authentication system then, from a performance perspective, then the message exchange to obtain the session key needs to be considered although the exchange only needs to be done once in a long time frame depending on the lifetime of the session ticket. This is particularly true in a mobile environment with a fast roaming user's host.

Public key based authentication usually provides the best scalability characteristics for key distribution but the protocols are performance demanding. A major disadvantage of the public key based user authentication in RSVP is the non-existing possibility to derive a session key. Hence every RSVP PATH or RESV message includes the certificate and a digital signature, which is a huge performance and bandwidth penalty. For a mobile environment with low performance devices, high latency and low bandwidth links this seems to be less encouraging. Note that a public key infrastructure is required to allow the PDP (or the first-hop router) to verify the digital signature and the certificate. To check for revoked certificates, certificate revocation lists or protocols like the Online Certificate Status Protocol [RFC2560] and the Simple Certificate Validation Protocol [MHHF01]. Then the integrity of the AUTH_DATA object via the digital signature is verified.

4.4 Communication between RSVP aware routers

a) Authentication

RSVP signaling messages are data origin authenticated and protected against modification and replay using the RSVP INTEGRITY object. The RSVP message flow between routers is protected based on the chain of trust and hence each router only needs to have a security association with its neighboring routers. This assumption was made because of performance advantages and because of special security characteristics of the core network where no user hosts are directly attached. In the core network the network structure does not change frequently and the manual distribution of shared secrets for the RSVP INTEGRITY object may be acceptable. The shared secrets may be either manually configured or distributed by using network management protocols like SNMP.

Independent of the key distribution mechanism host authentication with RSVP built-in mechanisms is accomplished with the keyed message digest in the RSVP INTEGRITY object computed using the previously exchanged symmetric key.

b) Integrity Protection

Integrity protection is accomplished with the RSVP INTEGRITY object with the variable length Keyed Message Digest field.

c) Replay Protection

Replay protection with the RSVP INTEGRITY object is extensively described in previous sections.

To enable crashed hosts to learn the latest sequence number used the Integrity Handshake mechanism is used in RSVP.

d) Confidentiality

Confidentiality is not provided by RSVP.

e) Authorization

Depending on the RSVP network QoS resource authorization at different routers may need to contact the PDP again. Since the PDP is allowed to modify the policy element, a token may be added to the policy element to increase the efficiency of the re-authorization procedure. This token is used to refer to an already computed policy decision. The communications interface from the PEP to the PDP must be properly secured.

f) Performance

The performance characteristics the protection of the RSVP signaling messages is largely determined by the key exchange protocol since the RSVP INTEGRITY object is only used to compute a keyed message digest of the transmitted signaling messages.

The security associations within the core network i.e. between individual routers (in comparison to the security association between the userÆs host and the first-hop router or with the attached network in general) can be established more easily because of the strong trust assumptions. Furthermore it is possible to use security associations with an increased lifetime to avoid too frequent rekeying. Hence there is less impact for the performance compared to the user to network interface. The security association storage requirements are also less problematic.

5. Miscellaneous Issues

This section describes a number of issues which illustrate some of the short-comings of RSVP with respect to security.

5.1 First Hop Issue

In case of end-to-end signaling an end host starts signaling to its attached network. The first-hop communication is often more difficult

because of the different requirements and a missing trust relationship. An end host must therefore obtain some information to start RSVP signaling:

- Does this network support RSVP signaling?
- Which node supports RSVP signaling?
- To which node is authentication required?
- Which security mechanisms are used for authentication?
- Which algorithms have to be used?
- Where should the keys/security association come from?
- Should a security association be established?

RSVP, as specified today, is used as a building block. Hence these questions have to be answered as part of overall architectural considerations. Without giving an answer to this question "ad-hoc" RSVP communication by an end host roaming to an unknown network is not possible. A negotiation of security mechanisms and algorithms is not supported for RSVP.

5.2 Next-Hop Problem

Throughout the document it was always assumed that the next RSVP node along the path is always known. Knowing your next hop is important to be able to select the correct key for the RSVP Integrity object to provide proper protection. In case that an RSVP node assumes to know which node is the next hop then the following protocol exchange can occur:

```
                                Integrity
                                 (A<->C)    +------+
                                  (3)       | RSVP |
                             +------------->+ Node |
                             |              |  B   |
                Integrity    |              +--+---+
                 (A<->C)     |                 |
      +------+    (2)     +--+----+            |
 (1)  | RSVP +----------->+Router |            |  Error
----->| Node |            | or    +<-----------+ (I am B)
      |  A   +<-----------+Network|       (4)
      +------+    (5)     +--+----+
                 Error       .
                (I am B)     .              +------+
                             .              | RSVP |
                             ...............+ Node |
                                            |  C   |
                                            +------+
```

Figure 5: Next-Hop Issue

When RSVP node A in Figure 5 receives an incoming RSVP Path message then standard RSVP message processing takes place. Node A then has to decide which key to select to protect the signaling message. We assume that some mechanism which is not further specified is used to make this decision. In this example node A assumes that the message will travel to RSVP node C. However because of some reasons (e.g. a route change, inability to learn the next RSVP hop along the path, etc.) the message travels to node B via a non-RSVP supporting router which cannot verify the integrity of the message (or cannot decrypt the Kerberos service ticket). The processing failure causes a PathErr message to be returned to the originating sender of the Path message. This error message also contains information about the node recognizing the error. In many cases a security association might not be available. Node A receiving the PathErr message might use the information returned with the PathErr message to select a different security association (or to establish one).

Figure 5 describes a behavior which might help node A to learn that an error occured. However, the description of Section 4.2 of [RFC2747] describes in step (5) that a signaling message is silently discarded if the receiving host cannot properly verify the message: "If the calculated digest does not match the received digest, the message is discarded without further processing." For RSVP Path alike messages this functionality is not really helpful.

The RSVP Path message therefore provides a number of functions: path discovery, detecting route changes, learning of QoS capabilities along the path using the Adspec object, (with some interpretation) next-hop discovery and possibly security association establishment (for example in case of Kerberos).

From a security point of view there is a conflict between

- Idempotent messages delivery and efficiency

Especially the RSVP Path message performs a number of functions. Supporting idempotent message delivery somehow contradicts with security association establishment and efficient message delivery and size. For example a "real" idempotent signaling message would contain enough information to perform security processing without depending on a previously executed message exchange. Adding a Kerberos ticket with every signaling message is, however, very inefficient. Using public key based mechanisms is even more inefficient when included in every signaling message. With public key based protection for idempotent messages there is additionally a risk of introducing denial of service attacks.

- RSVP Path message functionality and next-hop discovery

To protect an RSVP signaling message (and a RSVP Path message in particular) it is necessary to know the identity of the next RSVP aware node (and some other parameters). Without a mechanism for next-hop discovery an RSVP Path message is also responsible for this task. Without knowing the identity of the next hop the Kerberos principal name is also unknown. The so-called Kerberos user-to-user authentication mechanism is not supported which would allow the receiver to trigger the process of establishing Kerberos authentication is not supported. This issue will again be discussed in relationship with the last-hop problem.

It is fair to assume that a RSVP supporting node might not have a security association with all immediately neighboring RSVP nodes. Especially for inter-domain signaling, IntServ over DiffServ or for some new applications such as firewall signaling the next RSVP aware node might not be known in advance. The number of next RSVP nodes might be considerably large if they are separated by a large number of non-RSVP aware nodes. Hence a node transmitting a RSVP Path message might experience difficulties to properly protect the message if it serves as a mechanism to detect both the next RSVP node (i.e. Router Alert Option added to the signaling message and addressed to the destination address) and to detect route changes. It is fair to note that in an intra-domain case this might be possible due to manual configuration in case of a dense distribution of RSVP nodes.

There is nothing which prevents an adversary from continuously flooding an RSVP node with bogus PathErr messages. It might be possible to protect the PathErr message with an existing security association if available. A legitimate RSVP node would believe that a change in the path took place. Hence this node would try to select a different security association or try to create one with the indicated node. Hence an adversary can send a PathErr message at any time to confuse an RSVP node. If an adversary is located somewhere along the path then it might also be possible to act as a man-in-the-middle adversary if either authentication or authorization is not performed with the necessary accuracy.

5.3 Last-Hop Issue

This section tries to address practical difficulties when authentication and key establishment is accomplished with a protocol which shows some asymmetry in message processing when executed between two nodes. Kerberos is such a protocol and also the only supported protocol which provides dynamic session key establishment for RSVP. For first-hop communication authentication is typically done between a user and some network in the network (for example the access router). Especially in a mobile environment it is not feasible to authenticate end hosts based on their IP or MAC address. To show

the problem the typical processing steps for Kerberos are shown for first-hop communication:

a) The end host A learns the identity (i.e. Kerberos principal name) of some entity B. This entity B is either the next RSVP node or a PDP or the next policy aware RSVP node.

b) Entity A then requests a ticket granting ticket for the network domain. This assumes that the identity of the network domain is known.

c) Entity A then requests a service ticket for entity B which was learned in step (a).

d) Entity A includes the service ticket to the RSVP signaling message (inside the policy object). The Kerberos session key is used to protect the entire RSVP signaling message.

For last-hop communication this processing step theoretically has to be reversed; entity A is then a node in the network (for example the access router) and entity B is the other end host. This assumes that RSVP signaling is accomplished between two end hosts and not between an end host and a application server. The access router might however in step (a) not be able to learn the identity of the user's principal name since this information might not be available. Entity A could reverse the process by triggering an IAKERB exchange. This would cause entity B to request a service ticket for A as described above. IAKERB is however not supported.

5.4 RSVP and IPsec protected data traffic

QoS signaling requires flow information to be established at routers along a path. This flow identifier installed at each device tells the router which data packets should experience QoS treatment. RSVP typically establishes a flow identifier based on the 5-tuple (source IP address, destination IP address, transport protocol type, source port and destination port). If this 5-tuple information is not available then other identifiers have to be used. IPsec protected data traffic is such an example where the transport protocol and the port numbers are not accessible. Hence the IPsec SPI is used as a substitute for them. RFC 2207 considers these IPsec implications for RSVP and is based on three assumptions:

a) An end host, which initiates the RSVP signaling message exchange, has to be able to retrieve the SPI for given flow. This requires some interaction with the IPsec SADB and SPD. An application usually does not know the SPI of the protected flow and cannot provide the desired values. It can provide the signaling protocol daemon with flow identifiers. The signaling daemon would then need to query the IPsec

security association database by providing the flow identifiers as input parameters and the SPI as an output parameter.

b) RFC 2207 assumes an end-to-end IPsec protection of the data traffic. In IPsec is applied in a nested fashion then parts of the path do not experience QoS treatment. This problem can be treated as a tunneling problem but is initiated by the end host. A figure better illustrates the problem in case of enforcing secure network access:

```
+------+          +---------------+      +---------+          +------+
| Host |          | Security      |      | Router |           | Host |
|  A   |          | Gateway (SGW) |      |   Rx   |           |  B   |
+--+---+          +-------+-------+      +----+---+           +--+---+
   |                      |                   |                  |
   |IPsec-Data(           |                   |                  |
   | OuterSrc=A,          |                   |                  |
   | OuterDst=SGW,        |                   |                  |
   | SPI=SPI1,            |                   |                  |
   | InnerSrc=A,          |                   |                  |
   | OuterDst=B,          |                   |                  |
   | Protocol=X,          |IPsec-Data(        |                  |
   | SrcPort=Y,           | SrcIP=A,          |                  |
   | DstPort=Z)           | DstIP=B,          |                  |
   |=====================>| Protocol=X,       |IPsec-Data(       |
   |                      | SrcPort=Y,        | SrcIP=A,         |
   | --IPsec protected->  | DstPort=Z)        | DstIP=B,         |
   |    data traffic      |------------------>| Protocol=X,      |
   |                      |                   | SrcPort=Y,       |
   |                      |                   | DstPort=Z)       |
   |                      |                   |----------------->|
   |                      |                   |                  |
   |                      |     --Unprotected data traffic->     |
   |                      |                   |                  |
```

Figure 6: RSVP and IPsec protected data traffic

Host A transmitting data traffic would either indicate a 3-tuple <A, SGW, SPI1> or a 5-tuple <A, B, X, Y, Z>. In any case it is not possible to make a QoS reservation for the entire path. Similar examples are remote access using a VPN, protection of data traffic between the home agent (or a security gateway in the home network) and the mobile node and other. With a nested application of IPsec (for example IPsec between A and SGW and between A and B) the same problem occurs.

One possible solution to this problem is to change the flow identifier along the path to capture the new flow identifier after an IPsec endpoint.

IPsec tunnels which neither start nor terminate at one of the signaling end points (for example between two networks) should be addressed differently by recursively applying an RSVP signaling exchange for the IPsec tunnel. RSVP signaling within tunnels is addressed in [RFC2746].

c) It is assumed that SPIs do not change during the lifetime of the established QoS reservation. If a new IPsec SA is created then a new SPI is allocated for the security association. To reflect this change either a new reservation has to be established or the flow identifier of the existing reservation has to be updated. Since IPsec SAs have a longer lifetime this issue does not seem to be a major issue. IPsec protection of SCTP data traffic might more often require an IPsec SA (and an SPI) change to reflect added and removed IP addresses from an SCTP association.

5.5 End-to-End Security Issues and RSVP

End-to-end security for RSVP has not been discussed throughout the document. In this context end-to-end security refers to credentials transmitted between the two end hosts using RSVP. It is obvious that care must be taken to ensure that routers along the path are able to process and modify the signaling messages according to the processing procedure. Some objects however could be used for end-to-end protection. The main question however is what the benefit of such an end-to-end security is. First there is the question how to establish the required security association. Between two arbitrary hosts on the Internet this might turn out to be quite difficult. Furthermore it depends on an architecture where RSVP is deployed whether it is useful to provide end-to-end security. If RSVP is only used to signal QoS information into the network and other protocols have to be executed beforehand to negotiate the parameters and to decide which entity is charged for the QoS reservation then no end-to-end security is likely to be required. Introducing end-to-end security to RSVP would then cause problems with extensions like RSVP proxy [GD+02], Localized RSVP [MS+02] and others which terminate RSVP signaling somewhere along the path without reaching the destination end host. Such a behavior could then be interpreted as a man-in-the-middle attack.

5.6 IPsec protection of RSVP signaling messages

In this document it was assumed that RSVP signaling messages can also be protected by IPsec [RFC2401] in a hop-by-hop fashion between two adjacent RSVP nodes. RSVP uses a special processing of signaling messages which complicates IPsec protection. As we explain in this section IPsec should only be used for protection of RSVP signaling messages in a point-to-point communication environment (i.e. a RSVP message can only reach one RSVP router and not possibly more than one). This circumstance is caused by the combination of signaling message delivery and discovery into a single message. Furthermore the end-to-end addressing complicates IPsec handling considerably. This section tries to describe these complications.

RSVP messages are transmitted as raw IP packets with protocol number 46. It might be possible to encapsulate them in UDP as described in Appendix C of [RFC2205]. Some RSVP messages (Path, PathTear, and ResvConf) must have the Router Alert IP Option set in the IP header. These messages are addressed to the (unicast or multicast) destination address and not to the next RSVP node along the path. Hence an IPsec traffic selector can only use these fields for IPsec SA selection. If there is only a single path (and possibly every traffic is protected) then there is no problem for IPsec protection of signaling messages. This type of protection is not common and might only be used to secure network access between an end host and its first-hop router. Since the described RSVP messages are addressed to the destination address instead of the next RSVP node it is not possible to use IPsec ESP [RFC2406] or AH [RFC2402] in transport mode - only IPsec in tunnel mode is possible.

If there is more than one possible path which an RSVP message can take then the IPsec engine will experience difficulties to protect the message. Even if the RSVP daemon installs a traffic selector with the destination IP address then still there is no distinguishing element which allows to select the correct security association of one of the possible RSVP nodes along. Even if it possible to apply IPsec protection (in tunnel mode) for RSVP signaling messages by incorporating some additional information then there is still the possibility that the tunneled messages do not recognize a path change in a non-RSVP router. Then the signaling messages would simply follow different path than the data.

RSVP messages like RESV can be protected by IPsec since they are contain enough information to create IPsec traffic selectors which allow a differentiation between different next RSVP nodes. A traffic selector would then contain the protocol number and the source / destination address pair of the two communicating RSVP nodes.

The benefit of using IPsec is the available key management using either IKE [RFC2409], KINK [FH+01] or IKEv2 [IKEv2].

5.7 Authorization

In [TB+03a] two trust models (NJ Turnpike and NJ Parkway model) and two authorization models (per-session and per-channel financial settlement). The NJ Turnpike model gives a justification for the hop-by-hop security protection. RSVP supports the NJ Parkway model and per-channel financial settlement to some extend only. The communication procedures defined for policy object [Her95] can be improved to support the more efficient per-channel financial settlement by avoiding policy handling between inter-domain networks at a signaling message granularity. Additional information about expected behavior of policy handling in RSVP can also be obtained in [Her96].

[TB+03b] and [Tho02] provide additional information on authorization.

6. Conclusions

RSVP was the first QoS signaling protocol which provided some security protection. Whether RSVP provides enough security protection heavily depends on the environment where it is deployed. As RSVP is specified today should be seen as a building block that has to be adapted to a given architecture.

This document aims to provide more insights into the security of RSVP. It cannot not be interpreted as a pass or fail evaluation of the security provided by RSVP.

Certainly this document is not complete to describe all security issues related to RSVP. Some issues that require further considerations are RSVP extensions (for example [RFC2207]), multicast issues and other security properties like traffic analysis etc. Additionally the interaction with mobility protocols (micro- and macro-mobility) from a security point of view demands further investigation.

What can be learned from a practical protocol experience and from the increased awareness regarding security is that some of the available credential types have received more acceptance. Kerberos is such a system which is integrated in many IETF protocols today.
Public key based authentication techniques are however still considered to be too heavy-weight (computationally and from a bandwidth perspective) to be used for a per-flow signaling. The increased focus on denial of service attacks additionally demands a closer look on public key based authentication.

The following list briefly summarizes a few security or architectural issues which desire improvement:

* Discovery and signaling message delivery should be separated.

* For some applications and scenarios it cannot be assumed that neighboring RSVP aware nodes know each other. Hence some in-path discovery mechanism should be provided.

* Addressing for signaling messages should be done in a hop-by-hop fashion.

* Standard security protocols (IPsec, TLS or CMS) should be used whenever possible. Authentication and key exchange should separated from signaling message protection. In general it is necessary to provide key management to dynamically establish a security association for signaling message protection. Relying on manually configured keys between neighboring RSVP nodes is insufficient.

* The usage of public key cryptography for authorization tokens, identity representation, selective object protection, etc. is likely to cause fragmentation and problems.

* Public key authentication and user identity confidentiality provided with RSVP require some improvement.

* Public key based user authentication only provides entity authentication. An additional security association is required to protect the signaling message.

* Data origin authentication should not be provided by non-RSVP nodes (such as the PDP). Such a procedure could be accomplished by entity authentication during the authentication and key exchange phase.

* Authorization and charging should be better integrated in the base protocol.

* Selective message protection should be provided. A protected message should be recognizable from a flag in the header.

* Confidentiality protection is missing and should therefore be added to the protocol.

* Parameter and mechanism negotiation should be provided.

7. Security Considerations

This document discusses security properties of RSVP and as such, it is concerned entirely with security.

8. IANA considerations

This document does not address any IANA considerations.

9. Acknowledgments

I would like to thank Jorge Cuellar, Robert Hancock, Xiaoming Fu and Guenther Schaefer for their valuable comments. Additionally I would

like to thank Robert and Jorge for their time to discuss various issues with me. Furthermore I would like to thank Marc De Vuyst and Jukka Manner for their comments to this draft.

Appendix A: Dictionary Attacks and Kerberos

Kerberos might be used with RSVP as described in this document. Since dictionary attacks are often mentioned in relationship with Kerberos a few issues are addressed.

The initial Kerberos AS_REQ request (without pre-authentication, various extensions and without PKINIT) is unprotected. The response message AS_REP is encrypted with the client's long-term key. An adversary can take advantage of this fact by requesting AS_REP messages to mount an off-line dictionary attack. Using pre-authentication ([Pat92]) can be used to reduce this problem. However pre-authentication does not entirely prevent dictionary attacks by an adversary since he can still eavesdrop Kerberos messages if being located at the path between the mobile node and the KDC. With mandatory pre-authentication for the initial request an adversary cannot request a Ticket Granting Ticket for an arbitrary user. On-line password guessing attacks are still possible by choosing a password (e.g. from a dictionary) and then transmitting an initial request including pre-authentication data field. An unsuccessful authentication by the KDC results in an error message and the gives the adversary a hint to try a new password and restart the protocol again.

There are however some proposals that prevent dictionary attacks from happening. The use of Public Key Cryptography for initial authentication [TN+01] (PKINIT) is one such solution. Other proposals use strong-password based authenticated key agreement protocols to protect the user's password during the initial Kerberos exchange. In [Wu99] Tom Wu discusses the security of Kerberos and also discusses mechanisms to prevent dictionary attacks.

Appendix B: Example of User-to-PDP Authentication

The following Section describes an example of user-to-PDP authentication. Note that the description below is not fully covered by the RSVP specification and hence it should only be seen as an example.

Windows 2000, which integrates Kerberos into RSVP, uses a configuration with the user authentication to the PDP as described in [MADS01]. The steps for authenticating the user to the PDP in an intra-realm scenario are the following:

- Windows 2000 requires the user to contact the KDC and to request a Kerberos service ticket for the PDP account AcsService in the local realm.

- This ticket is then embedded in the AUTH_DATA element and included in either the PATH or the RESV message. In case of MicrosoftÆs implementation the user identity encoded as a distinguished name is encrypted with the session key provided with the Kerberos ticket. The Kerberos ticket is sent without the Kerberos authdata element that contains authorization information as explained in [MADS01].

- The RSVP message is then intercepted by the PEP who forwards it to the PDP. [MADS01] does not state which protocol is used to forward the RSVP message to the PDP.

- The PDP who finally receives the message decrypts the received service ticket. The ticket contains the session key which was used by the user's host to
a) Encrypt the principal name inside the policy locator field of the AUTH_DATA object and to
b) Create the integrity protected Keyed Message Digest field in the INTEGRITY object of the POLICY_DATA element. The protection described here is between the user's host and the PDP. The RSVP INTEGRITY object on the other hand is used to protect the path between the users host and the first-hop router since the two message parts terminate at a different node and a different security association must be used. The interface between the message intercepting first-hop router and the PDP must be protected as well.
c) The PDP does not maintain a user database and [MADS01] describes that the PDP may query the Active Directory (a LDAP based directory service) for user policy information.

Appendix C: Literature on RSVP Security

Very few documents address the security of RSVP signaling. This section briefly describes some important documents.

Improvements to RSVP are proposed in [WW+99] to deal with insider attacks. Insider attacks are caused by malicious RSVP routers modifying RSVP signaling messages in such a way that they cause harm to the nodes participating in the signaling message exchange.

As a solution non-mutuable RSVP objects are digitally signed by the sender. This digital signature is added to the RSVP PATH message. Additionally the receiver attaches an object to the RSVP RESV message containing a "signed" history. This value allows intermediate RSVP routers (together with the previously signed value) to detect a malicious RSVP node.

A few issues are, however, left open in the document. Replay attacks are not covered and it is therefore assumed that timestamp-based replay protection is used. In order to detect a malicious node it is necessary that all routers along the path are able to verify the digital signature. This requires a global public key infrastructure and also a client-side PKI. Furthermore the computational requirements to verify and compute digital signatures with each signaling message might place a burden on a real-world deployment. Authorization is not considered in the document which might have an influence on the implication of signaling message modification. Hence the chain-of-trust relationship (or step towards a different direction) should be considered in relationship with authorization.

In [TN00] the above described idea of detecting malicious RSVP nodes is improved by addressing the performance aspects. The proposed solution is somewhat between hop-by-hop security and the above described approach by separating the end-to-end path into individual networks. Furthermore some additional RSVP messages (i.e. feedback messages) are introduced to implement a mechanism call "delayed integrity checking". In [TN+01] the approach presented with [TN00] is enhanced.

10. Normative References

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

[RFC2750] Herzog, S.: "RSVP Extensions for Policy Control", RFC 2750, January, 2000.

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

[RFC2748] Boyle, J., Cohen, R., Durham, D., Herzog, S., Rajan, R., Sastry, A.: "The COPS(Common Open Policy Service) Protocol", RFC 2748, January, 2000.

[RFC2749] Boyle, J., Cohen, R., Durham, D., Herzog, S., Rajan, R., Sastry, A.: "COPS usage for RSVP", RFC 2749, January, 2000.

[RFC2207] Berger, L., OÆMalley, T.: "RSVP Extensions for IPSEC Data Flows", RFC 2207, September 1997.

[RFC1321] Rivest, R.: "The MD5 Message-Digest Algorithm", RFC 1321, April, 1992.

[RFC1510] Kohl, J., Neuman, C.: "The Kerberos Network Authentication Service (V5)", RFC 1510, September 1993.

[RFC2104] Krawczyk, H., Bellare, M., Canetti, R.: "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February, 1997.

[RFC2205] Braden, R., Zhang, L., Berson, S., Herzog, S., Jamin, S.: "Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification", RFC 2205, September 1997.

11. Informative References

[CA+02] Calhoun, P., Arkko, J., Guttman, E., Zorn, G., Loughney, J.: "DIAMETER Base Protocol", <draft-ietf-aaa-diameter-17.txt>, (work in progress), December, 2002.

[DBP96] Dobbertin, H., Bosselaers, A., Preneel, B.: "RIPEMD-160: A strengthened version of RIPEMD", in "Fast Software Encryption, LNCS Vol 1039, pp. 71-82", 1996.

[DG96] Davis, D., Geer, D.: "Kerberos With Clocks Adrift: History, Protocols and Implementation", in "USENIX Computing Systems Volume 9 no. 1, Winter", 1996.

[Dob96] Dobbertin, H.: "The Status of Md5 After a Recent Attack," RSA Laboratories' CryptoBytes, Volume 2, Number 2, 1996.

[GD+02] Gai, S., Dutt, D., Elfassy, N., Bernet, Y.: "RSVP Proxy", <draft-ietf-rsvp-proxy-03.txt>, (expired), March, 2002.

[HA01] Hornstein, K., Altman, J.: "Distributing Kerberos KDC and Realm Information with DNS", <draft-ietf-krb-wg-krb-dns-locate-03.txt>, (expired), July, 2002.

[HH01] Hess, R., Herzog, S.: "RSVP Extensions for Policy Control", <draft-ietf-rap-new-rsvp-ext-00.txt>, (expired), June, 2001.

[Jab96] Jablon, D.: "Strong password-only authenticated key exchange", Computer Communication Review, 26(5), pp. 5-26, October, 1996.

[MADS01] "Microsoft Authorization Data Specification v. 1.0 for Microsoft Windows 2000 Operating Systems", April, 2000.

[RFC2284] Blunk, L. and J. Vollbrecht, "PPP Extensible Authentication Protocol (EAP)", RFC 2284, March 1998.

[MHHF01] Malpani, A., Hoffman, P., Housley, R., Freeman, T.: "Simple Certificate Validation Protocol (SCVP)", <draft-ietf-pkix-scvp-11.txt>, (work in progress), December, 2002.

[MS+02] Manner, J., Suihko, T., Kojo, M., Liljeberg, M., Raatikainen, K.: "Localized RSVP", <draft-manner-lrsvp-00.txt>, (expired), May, 2002.

[Pat92] Pato, J., "Using Pre-Authentication to Avoid Password Guessing Attacks", Open Software Foundation DCE Request for Comments 26, December, 1992.

[PGP] "Specifications and standard documents", http://www.pgpi.org/doc/specs/ (March, 2002).

[PKTSEC] PacketCable Security Specification, PKT-SP-SEC-I01-991201, Cable Television Laboratories, Inc., December 1, 1999, http://www.PacketCable.com/ (June, 2003).

[Rae01] Raeburn, K.: " Encryption and Checksum Specifications for Kerberos 5", <draft-ietf-krb-wg-crypto-05.txt>, (work in progress), June, 2003.

[RFC2315] Kaliski, B.: " PKCS #7: Cryptographic Message Syntax Version 1.5", RFC 2315, March, 1998.

[RFC2440] Callas, J., Donnerhacke, L., Finney, H., Thayer, R.: "OpenPGP Message Format", RFC 2440, November, 1998.

[RFC2495] Housley, R., Ford, W., Polk, W., Solo, D.: "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", RFC 2459, January, 1999.

[RFC2560] Myers, M., Ankney, R., Malpani, A., Galperin, S., Adams, C.: "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - - OCSP", RFC 2560, June, 1999.

[RFC2630] Housley, R.: "Cryptographic Message Syntax", RFC 2630, June, 1999.

[RFC2865] Rigney, C., Willens, S., Rubens, A., Simpson, W.: "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June, 2000.

[SHA] NIST, FIPS PUB 180-1, "Secure Hash Standard", April, 1995.

[TN+01] Tung, B., Neuman, C., Hur, M., Medvinsky, A., Medvinsky, S., Wray, J., Trostle, J.: "Public Key Cryptography for Initial Authentication in Kerberos", <draft-ietf-cat-kerberos-pk-init-16.txt>, (expired), October, 2001.

[Wu99] Wu, T.: "A Real-World Analysis of Kerberos Password Security", in "Proceedings of the 1999 Network and Distributed System Security", February, 1999.

[TB+03a] H. Tschofenig, M. Buechli, S. Van den Bosch, H. Schulzrinne: "NSIS Authentication, Authorization and Accounting Issues", <draft-tschofenig-nsis-aaa-issues-01.txt>, (work in progress), March, 2003.

[TB+03b] H. Tschofenig, M. Buechli, S. Van den Bosch, H. Schulzrinne, T. Chen: "QoS NSLP Authorization Issues", <draft-tschofenig-nsis-qos-authz-issues-00.txt>, (work in progress), June, 2003.

[Her95] Herzog, S.: "Accounting and Access Control in RSVP", <draft-ietf-rsvp-lpm-arch-00.txt>, (expired), November, 1995.

[Her96] S. Herzog: "Accounting and Access Control for Multicast Distributions: Models and Mechanisms", PhD Dissertation, University of Southern California, June 1996, available at: http://www.policyconsulting.com/publications/USC%20thesis.pdf, (June, 2003).

[Tho02] M. Thomas: "Analysis of Mobile IP and RSVP Interactions", <draft-thomas-nsis-rsvp-analysis-00.txt>, (work in progress), October 2002.

[FH+01] Thomas, M., Vilhuber, J.: "Kerberized Internet Negotiation of Keys (KINK)", <draft-ietf-kink-kink-05.txt>, (work in progress), January, 2003.

[RFC2402] Kent, S., Atkinson, R.: "IP Authentication Header", RFC 2402, November, 1998.

[RFC2406] Kent, S., Atkinson, R.: "IP Encapsulating Security Payload (ESP)", RFC 2406, November, 1998.

[RFC2409] Harkins, D., Carrel, D.: "The Internet Key Exchange (IKE)", RFC 2409, November, 1998.

[IKEv2] C. Kaufman: "Internet Key Exchange (IKEv2) Protocol", Internet Draft, <draft-ietf-ipsec-ikev2-08.txt>, (work in progress), June, 2003.

[WW+99] Wu, T., Wu, F. and Gong, F.: "Securing QoS: Threats to RSVP Messages and Their Countermeasures", in "IEEE IWQoS, pp. 62-64, 1999.

[TN00] Talwar, V. and Nahrstedt, K.: "Securing RSVP For Multimedia Applications", in "Proceedings of ACM Multimedia (Multimedia Security Workshop)", Los Angeles, November, 2000.

[TN+01] Talwar, V., Nath, S., Nahrstedt, K.: "RSVP-SQoS : A Secure RSVP Protocol", in "International Conference on Multimedia and Exposition", Tokyo , Japan, August 2001.

Author's Contact Information

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

Full Copyright Statement

Acknowledgement

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