Hannes Tschofenig

Siemens AG

Internet Draft

Document: draft-ietf-nsis-rsvp-sec-properties-00.txt

Expires: April, 2003

October, 2002

RSVP Security Properties

<draft-ietf-nsis-rsvp-sec-properties-00.txt>

Status of this Memo

Abstract

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 tries to summarize the security properties of RSVP and
to view them from a different point of view. This work in NSIS is
part of the overall process of analyzing other protocols and to learn
from their design considerations. This document should also provide a
starting point for security discussions.

Table of Contents

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 tries to summarize the security properties of RSVP and to view them from a different point of view. This work in NSIS is

part of the overall process of analyzing other 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 the last Section briefly addresses issues like the discussion of the vulnerability of Kerberos against dictionary attacks and open issues in the context of RSVP and issues for further investigation.

2 Terminology

To begin with the description of the security properties of RSVP it is natural to describe some basic building-blocks.

- 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.

3 **Overview**

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 û 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, for example PF_KEY [RFC2367], 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
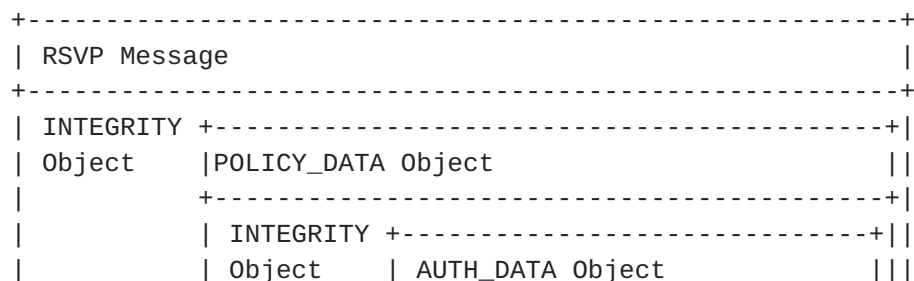Identifier.

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
ôProviding policy based admission control mechanism based on user
identities or application is one of the prime requirements.ö 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                   |||
|          |                +------------------------------+||
|          +-----------------------------------------+|
+-----------------------------------------------------+
```

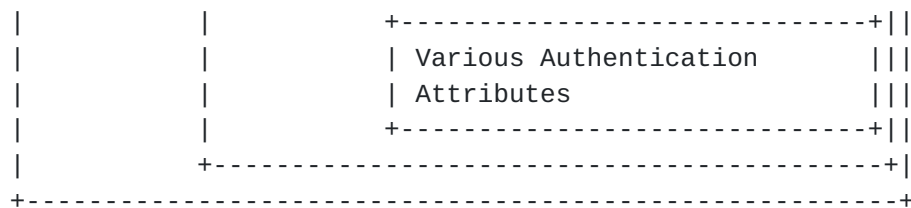   Figure 1: Security relevant Objects and Elements within the RSVP
                                    message

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.

A successful user authentication is the basis for doing 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 ôencryptedö
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   | Description                   |
|    Type      |                               |
+==============================================|
| 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 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 last error message allows the
user's host to discover the type of credentials supported although by
very inefficient means. 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. A host stores the latest sequence
number of a fresh and correctly authenticated packet. 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 packets 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)
```
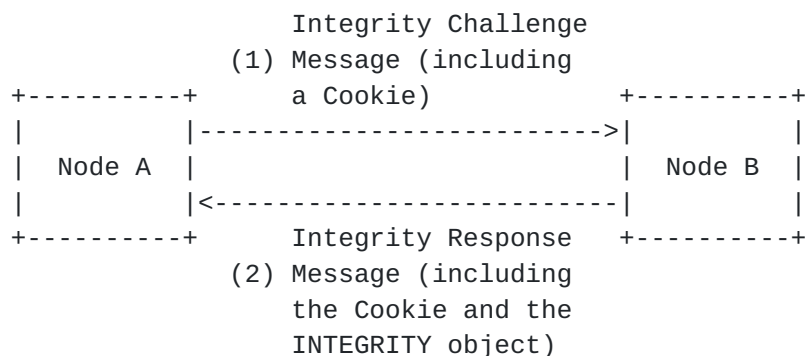
Figure 2: RSVP Integrity Handshake

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 ôChallenge Cookieö 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.

Whether or not to use the RSVP Integrity Challenge/Response mechanism
is a site-local decision since it may not be needed in all network
environments. It is however recommended to use the RSVP Integrity
Handshake protocol.

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|
```
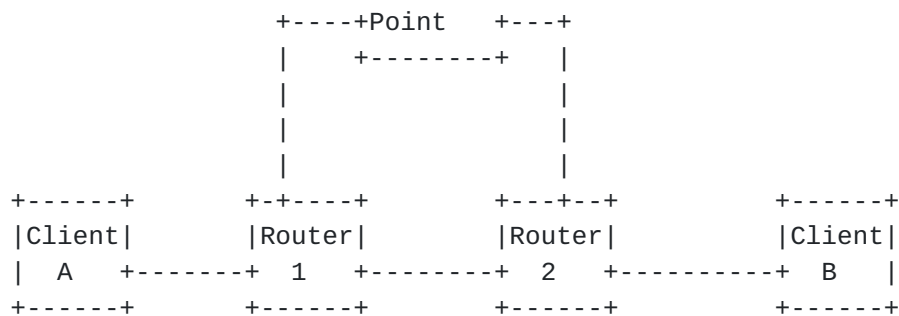
```
                   +----+Point   +---+
                   |    +--------+  |
                   |                |
                   |                |
                   |                |
   +------+      +-+----+        +---+--+         +------+
   |Client|      |Router|        |Router|         |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. Although not mentioned in [RFC2747] it is also
possible to use IPSec [RFC2401] to protect the RSVP signaling traffic
from the client to the first-hop router. If we use IPSec to protect
the interface between the userÆs host and the first hop router then
the optional RSVP INTEGRITY object may not be required. It may also
be possible (which requires a further investigation) whether an
existing IPSec security association may also be (re-)used for RSVP.

IPSec allows the key exchange protocol IKE [RFC2409] to be used to dynamically negotiate IPSec security associations. Note that KINK [FH+01] and other protocols are available that are also able to establish an IPSec security association. This text mainly refers to IKE since it is the most frequently used protocol for this purpose. A detailed description of IPSec and IKE is outside the scope of this document. Since IKE is computationally expensive it might create a computational burden to re-establish a new IPSec SA based of the movement of a mobile user host. Work at the SEAMOBY group tries to tackle this problem by using IPSec Context Transfer protocols. Hence in this case we would avoid triggering a separate key exchange protocol run for RSVP to protect messages at each layer if they terminate at the same node.

It is an open issue whether it is enough to provide IPSec protection of messages between the userÆs host and the first-hop router although different protocols (i.e. protocols executed at different protocol layers) (possibly) terminate at different endpoints.

- 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.

An additional protocol needs to be executed after each user is authenticated via Kerberos to establish a session key and to allow

multicast specific functionality like entering a group, leaving a
group to be executed securely. This would additionally allow
accounting and billing to be used efficiently and on a per-user
basis. This session key is then used to protect RSVP signaling
messages. These issues definitely need further investigation and are
not fully described in this version of the document.

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. Since the RSVP Integrity
object is an optional element of the RSVP message IPSec protection of
the signaling message to the router may also provide integrity
protection either with IPSec AH [RFC2402] or IPSec ESP [RFC2406] as
mentioned already in the previous paragraph.

Furthermore it is stated that other keyed hash functions apart from
HMAC-MD5 may be used within the RSVP INTEGRITY object and it is
obvious that both communicating entities must have security

associations indicating the algorithm used. 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] instead are known to provide better security properties.

c) Replay Protection

The main mechanism used for replay protection in RSVP are sequence
numbers whereby the sequence number is included in the RSVP INTEGRITY
object. The properties of this sequence number mechanisms 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.

If IPSec is used to protect RSVP messages then the optional IPSec replay protection mechanism may be used which is also based on sequence numbers with a window mechanism. This window mechanism may (theoretically) also cause problems whereby an adversary reorders messages. This is however very difficult to exploit since the signaling messages are exchanged at a relatively low rate compared to regular data traffic that may also be protected with IPSec.

- 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 ôdiscoveredö 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 directly supported by RSVP. However, IPSec can provide confidentiality by encrypting the transmitted signaling traffic with IPSec ESP.

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. via AAA protocols (for example using RADIUS [RFC2865] or DIAMETER [CA+02]) and authorization information is downloaded to one or more network elements for example to the access router/first hop router to modify filter rules to enable the IP traffic forwarding. The access router is therefore acting as a firewall with dynamically created filter rules based on a successful host or user authentication. Issues related to network access 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. Without a policy element being embedded inside the RSVP message no policy-based admission control can be done.

The interaction between the two access authorization procedures (and the filter-installation at the various network devices) will likely be investigated in more detail in the MIDCOM working group.

f) Performance

The computation of the keyed message digest for a RSVP INTEGRITY object does not represent a performance problem. The same is true for IPSec AH (or IPSec ESP). 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 because of the characteristics of the keyed message digest routine.

The 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 the 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.

Section 4.5.3 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 that PGP is used. 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 a OpenPGP [RFC2440]) and there

has been no indication which version should be used. When thinking
about CMS support for RSVP the main question that has to be answered
is whether a public key based authentication (and key agreement
mechanism) should be supported for a QoS signaling protocol.

Especially the risks of denial of service attacks, large processing,
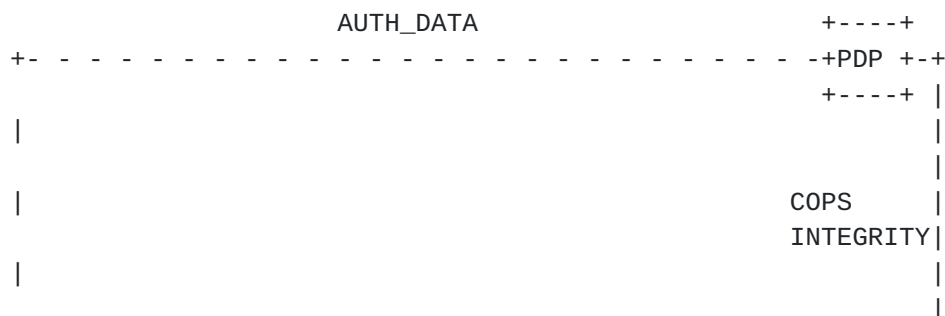memory and bandwidth utilization should be considered.

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) If a Kerberos session ticket is included but without using the
Kerberos session key then the analogon of 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 below 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
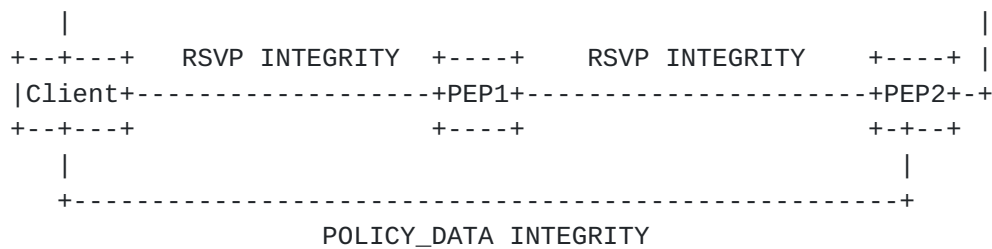the AUTH_DATA element which has no RSVP INTEGRITY object included.

```
                    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 ôRSVP INTEGRITY object is used to
protect the policy object containing user identity information from
security (replay) attacks.ö. Hence the public key based
authentication does not support the RSVP based replay protection
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.

The use of public key systems 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 user's credentials without destroying the digital
signature. Additionally the RSVP initiating host, where multiple
users may have access, must be trustworthy even if a smartcard is

used since otherwise, replay attacks with a recorded AUTH_DATA object
are possible. Note that this however violates the hop-by-hop security
assumption. 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.

There are two additional issues related to a Kerberos based user
authentication in the context of replay protection. 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 we assume that the Kerberos session key is used for RSVP then
there may be a need for rekeying. If we assume that a policy at the
user's host indicates when to rekey then the next RSVP message
includes a new Kerberos session ticket that is then used by the
verifying entity. If the lifetime of the Kerberos ticket or other
policies do not affect rekeying then an RSVP security association may
never require rekeying at all because of the large sequence number
space.

d) (User Identity) Confidentiality

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

There has often been the discussion whether the effort for protecting
user identity is worth the additional complexity. With the increasing
privacy awareness there must be at least a discussion on the
mechanisms provided by the given protocol. The main question in this
context is about the threat model i.e. against which entity the user
identity should be protected. Since RSVP does not make any
assumptions about the underlying key management protocol for most
parts it is difficult to make a judgment. However for the identity

representation part of the protocol it is possible to make some observations. We assume that the most important threat for a user is to reveal his identity to an adversary located between the userÆs host and the first-hop router. 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. We cannot however provide user identity confidentiality against the network provider to which the user is attached. Different mechanisms must be deployed to disallow the network provider to create a profile of the user. These mechanisms are outside the scope of this document since there is a strong involvement with the initial authentication and key agreement protocol executed between the user and the visited network.

If the link between the userÆs host and the first hop router is protected with IPSec ESP then confidentiality of the entire signaling messages is provided. Note however that the IPSec protection may terminate at the different node than the RSVP policy aware signaling does. The focus of this Section is, however, the functionality provided by RSVP.

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].

There is no option to encrypt the user or application identity without Kerberos or public key mechanisms are used since the selection of an appropriate security association is not possible.

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) that were published in the context of the AES competition.

The task of evaluating the confidentiality provided for the user requires to look 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 user identity used with Kerberos. 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 users 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 accounting might be deployed
in the near future that do not require the any type of inter-domain
communication. Obviously there is a strong interrelationship between
the authorization and the process of accounting. Note that the term
accounting in this context is not only related to process of
metering. Metering is only the process of measuring and collecting
resource usage information. Instead the term unites metering,
pricing, charging and billing.

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. IPSec may also provide RSVP signaling message protection. 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.

If IPSec is used in a hop-by-hop fashion then the required security associations may be manually created or dynamically distributed with IKE by either using symmetric or asymmetric authentication modes. A description of the existing IKE authentication modes and IKE security properties is outside the scope of this document. The reader is referred to the relevant documents at the IPSec working group.

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. In case of IPSec host authentication is accomplished with the keyed message digest included in the Authentication Data field of the IPSec Authentication Header included in every IP packet.

b) Integrity Protection

Integrity protection is either accomplished with the RSVP INTEGRITY object with the variable length Keyed Message Digest field or with the IPSec Authentication Header. A description of the IPSec AH is found in [RFC2402] and IPSec ESP [RFC2406] with null encryption is found in [RFC2410]. The main difference between IPSec and RSVP protection is the layer at which the security is applied.

c) Replay Protection

Replay protection with the RSVP INTEGRITY object is extensively described in previous Sections. IPSec provides an optional window-based replay protection, which may cause problems if a strict message ordering of RSVP messages is required. This problem was already discussed in a previous Section and a possible solution is to include the RSVP INTEGRITY object without a key, which reduces the RSVP integrity protection to a simple MD5 hash. This modification must however be integrated into an existing implementation and it is not clear whether the RSVP standard allows this modification. If the RSVP implementation is able to access the IPSec Security Association Database and retrieve the required security association then no such modification to RSVP is required and IKE is only used to distribute the security associations. This however requires the RSVP implementation to trigger the IKE exchange.

To enable crashed hosts to learn the latest sequence number used the Integrity Handshake mechanism is used in RSVP as explained in a Section above. IPSec does not provide such a mechanism since a crashed host looses its negotiated security associations and therefore has to re-negotiate them using IKE. Note that manually configured IPSec security associations do not provide replay protection because a sequence number rollover would require the establishment of a new SA. This is obviously not possible when using manually configured IPSec SAs. Using IKE with pre-shared secrets is therefore a simple solution.

d) Confidentiality

Confidentiality is not provided by RSVP but using IPSec ESP in a hop-by-hop mode can provide it. The usage of IPSec ESP for RSVP is not recommended because of the additional overhead for little additional security benefit if we think of the underlying assumed trust model of chain of trust. Hence there must be a good reason why to require

confidentiality in a hop-by-hop fashion in the core network of the
same administrative domain. If the RSVP network spawns different
provider networks then it is possible to encapsulate RSVP messages
between RSVP networks over a non-RSVP cloud similar to a VPN. Such a
configuration is mainly determined by the network structure of a
provider.

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 or IPSec AH are only used to compute a keyed
message digest of the transmitted messages. Furthermore only RSVP
signaling messages are protected and the protection of the
application data stream is outside the scope of RSVP. IPSec ESP
provides a performance penalty but may only be rarely used. A network
administrator may however use IPSec ESP in transport mode with NULL
encryption to provide the same functionality as IPSec AH but with the
chance of better hardware support.

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.

4.5 Miscellaneous Issues

4.5.1   Dictionary Attacks and Kerberos

This Section addresses issues related to Kerberos and its
vulnerability against dictionary attacks since there often seems to

be a misunderstanding. The reason for including this discussion in this document is that Kerberos seems to be one of the most widely supported authentication and key distribution systems available.

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 like the Encrypted Key Exchange protocol (EKE) to avoid leaking of user password information. B. Jaspan investigated the use of EKE for Kerberos V5 called ôDual-workfactor Encrypted Key Exchangeö [Jas96] which is described below.

With the PA-ENC-DH pre-authentication Jaspan included the Diffie-Hellman ôpublic keyö of the client encrypted with the user password in the initial AS_REQ to the Authentication Server. Additionally the modulus m is included since the client can choose this value dynamically.

It is interesting to note that pre-authentication was orginally introduced to allow the user to authenticate to the AS with the inital AS_REQ message . The use of the Encrypted Key Exchange protocol [BM92] as a pre-authentication mechanism does not allow the Authentication Server to authenticate the client since this would require the client to include verifiable data (e.g. a keyed message digest for data origin authentication) but this destroys the properties of EKE. EKE was designed to create a strong-password based

authentication protocol that is resistant against dictionary attacks.
Hence after the second message the Authentication Server is
authenticated to the client by showing that he was able to compute
the shared key k(a,as) used to encrypt the first part of message (2).
The client is not authenticated to the Authentication Server.

It is obvious that both the client and the Authentication Server must
be able to provide good random numbers for the creation of the
Diffie-Hellman key pair. Jaspan additionally noted that the timestamp
in the response from the Authentication Server (AS_REP message) can
be used to eliminate the dependency on time synchronization of the
Kerberos protocol. The client can use this value to adjust his clock
after successful authentication of the Authentication Server.

The vulnerability against denial of service attacks is a disadvantage
common to many strong-password based authenticated key agreement
protocols. Nothing prevents an adversary from flooding the
Authentication Server with bogus AS_REQ messages using the pre-
authentication method PA-ENC-DH. This forces the Authentication
Server to create a Diffie-Hellman public/private key pair, to decrypt
the received response and to compute the session key k(a,as) and to
return a message to the source IP address of the previously received
message. Even if the Authentication Server does not re-create a new
public/private key pair with every session he still has to compute
the session key which requires multiprecision operations and this is
time consuming.

Jaspan furthermore noted that the missing client authentication can
be used by an undetectable on-line password guessing attack as
described in [DH95]. An adversary sends an AS_REQ for a user B
encrypted with a password k(bÆ). The Authentication Server decrypts
the value of the pre-authentication field with the real user password
k(b) and encrypts his response to the adversary. If the adversary
correctly guessed the password of user B then the receive response
verifies correctly. Jaspan proposed to modify the KDC to allow only a
certain number of requests per day but this can be used by an
attacker to mount a denial of service attack against such users to
lock their accounts by sending a number of incorrect requests to the
KDC. The KDC would then reject Ticket Granting Ticket or even a
service ticket from legitimate users.

Tom Wu mentioned in [Wu99] the use of a variant of SRP [Wu98] and the
use of SPEKE [Jab96] to be used in the pre-authentication process as
possible candidates to prevent dictionary attacks. Unfortunately Wu
does not explain the proposals in detail.

Currently only PKINIT is available for preventing off-line dictionary attacks. Other proposals described above like SPEKE, SRP etc. are not included in the current Kerberos version. IPR issues may be one of the reasons.

4.5.2   **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.

4.5.3   Open Issues

The following issues have often been mentioned in the context of RSVP. However a design decision with regard to end-to-end security and a framework for accounting and charging cannot be found in the main RSVP documents.

a) 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 which turned out to be quite difficult between two arbitrary hosts. 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 actually has to pay for the reservation then no end-to-end security is likely to be required. End-to-end security if introduced into RSVP would then cause problem 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.

b) Accounting/Charging Framework

Many documents have been published in the context of accounting and charging for RSVP/IntServ, pricing, business models etc. The reasons for large number of proposals and the ôrareö number of used mechanisms are manifold. The lack of a defined framework makes it difficult to argument whether the processing of credentials within the policy element and a possible forwarding to other network domains is required. Forwarding user credentials would allow other networks to authenticate the identity acting as a signaling source. If

credentials are however removed then no such behavior can be achieved and each neighboring domain only exchanges accounting data to the next domain without taking the length of the real number of visited domains into consideration. Scalability problems in the core network speak against solutions that verify the user credentials by every network along the path or solutions that create an analogon to a long-distance call. A long-distance call in terms of RSVP can be simulated by adding a monetary value for the requested resource at each network along the path. Issues related to accounting will receive further attention in the NSIS framework discussion.

5 Conclusions

It is often argued that RSVP cannot be used in particular environments. Whether this is true or not cannot be answered by the author but what can be observed is the following: RSVP should be seen as a building block that has to be adapted to provide the desired services for a given architecture. The point to stress is "architecture". Hence it is difficult to state whether RSVP provides the adequate security for a given architecture without a particular framework. The author represents the opinion that the RSVP designers and architects did a good job in providing the necessary blocks (including security relevant parts) that allows RSVP to be easily adapted to most architectures. By including some RSVP extensions additional flexibility and features are provided.

This document aims to provide more insights into the security of RSVP explained with different words from a different view. It must not be interpreted as a pass or fail evaluation of the security provided by RSVP.

                    RSVP Security Properties   October 2002

Certainly this document is not complete to describe all issues related to RSVP but it serves as a starting point. 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. As stated in the previous Section the interaction with accounting/charging issues are worth a closer look.

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.

6 Security Considerations

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

7 IANA considerations

This document does not address any IANA considerations.

8 Open Issues

A future version of this document might include a description of the
following issues:

.   Security Algorithm Negotiation
.   Denial of Service Attacks
.   IPSec implications for RSVP
o   VPN issues
o   Implications for flow identification
o   IPSec protection of signaling messages
o   Nested IPSec Tunnels
.   First-Peer Problem
.   Next-Peer Problem
.   Last-Peer problem
.   Possibly: Reservation Ownership Issues

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 for
his comments to the draft.

10   References

[BM92]   Bellovin, B., Merrit, M.: ôEncrypted Key Exchange:
Password-based protocols secure against dictionary attacksö, in
ôProceedings of the IEEE Symposium on Research in Security and
Privacyö, May, 1992.

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

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

[DG96]   Davis, D., Geer, D.: ôKerberos With Clocks Adrift:
History, Protocols and Implementationö, in ôUSENIX Computing Systems
Volume 9 no. 1, Winterö, 1996.

[DH95]   Ding, Y., Horster, P.: ôUndetectable On-line Password
Guessing Attacksö, Operating Systems Review, 29(No. 4), pp. 77-86,
1995.

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

[FH+01]   Thomas, M., Vilhuber, J.: "Kerberized Internet
Negotiation of Keys (KINK)", <draft-ietf-kink-kink-03.txt>, (work in
progress), May, 2002.

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

[HA01]   Hornstein, K., Altman, J.: "Distributing Kerberos KDC
and Realm Information with DNS", <draft-ietf-krb-wg-krb-dns-locate-
03.txt>, (work in progress), 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.: ôStrong password-only authenticated key
exchangeô, Computer Communication Review, 26(5), pp. 5-26, October,
1996.

[Jas96]   Jaspan, B.: ôDual-workfactor Encrypted Key Exchange:
Efficiently Preventing Password Chaining and Dictionary Attacksö, in
ôProceedings of the Sixth Annual USENIX Security Conferenceö, pp. 43-
50, July, 1996.

[MADS01]   ôMicrosoft Authorization Data Specification v. 1.0 for
Microsoft Windows 2000 Operating Systemsö, April, 2000, available at:

http://www.microsoft.com/technet/security/kerberos/default.asp,
February, 2001.

[MHHF01]   Malpani, A., Hoffman, P., Housley, R., Freeman, T.:
ôSimple Certificate Validation Protocol (SCVP)ö, <draft-ietf-pkix-
scvp-09.txt>, (work in progress), June, 2002.

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

[Rae01]   Raeburn, K.: "Rijndael, Serpent, and Twofish
Cryptosystems for Kerberos 5", <draft-raeburn-krb-rijndael-krb-
01.txt>, (expired), July, 2001.

[RF2367]   McDonald, D., Metz, C., Phan, B.: ôPF_KEY Key
Management API, Version 2ö, RFC 2367, July, 1998.

[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.: ôHMAC: Keyed-
Hashing for Message Authenticationö, RFC 2104, February, 1997.

[RFC2205]   Braden, R., Zhang, L., Berson, S., Herzog, S., Jamin,
S.: äResource ReSerVation Protocol (RSVP) û Version 1 Functional
Specificationô, RFC 2205, September 1997.

[RFC2207]   Berger, L., OÆMalley, T.: äRSVP Extensions for IPSEC
Data Flowsô, RFC 2207, September 1997.

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

[RFC2367]   McDonald, D., Metz, C., Phan, B.: "PF_KEY Key
Management API, Version 2", RFC 2367, July, 1998.

[RFC2401]   Kent, S., Atkinson, R.: "Security Architecture for the Internet Protocol", RFC 2401, November, 1998.

[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.: ôThe Internet Key Exchange (IKE)ö, RFC 2409, November, 1998.

[RFC2410]   Glenn, R., Kent, S.: "The NULL Encryption Algorithm and Its Use With IPsec", RFC 2410, November, 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.: ôX.509 Internet Public Key Infrastructure Online Certificate Status Protocol û OCSPö, RFC 2560, June, 1999.

[RFC2630]   Housley, R.: ôCryptographic Message Syntaxö, RFC 2630, June, 1999.

[RFC2747]   Baker, F., Lindell, B., Talwar, M.: ôRSVP Cryptographic Authenticationö, RC 2747, January, 2000.

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

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

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

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

[RFC3182]   Yadav, S., Yavatkar, R., Pabbati, R., Ford, P., Moore,

T., Herzog, S., Hess, R.: ôIdentity Representation for RSVPö, RFC
3182, October, 2001.

[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.: ôPublic Key Cryptography for
Initial Authentication in Kerberosö, <draft-ietf-cat-kerberos-pk-
init-16.txt>, (work in progress), October, 2001.

[Wu98]   Wu, T.: ôThe Secure Remote Password Protocolô, in
ôProceedings of the Internet Society Network and Distributed System
Security Symposiumö, pp. 97-111, March, 1998.

[Wu99]   Wu, T.: ôA Real-World Analysis of Kerberos Password
Securityö, in ôProceedings of the 1999 Network and Distributed System
Securityö, February, 1999.

11   Author's Contact Information

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

12   Full Copyright Statement

revoked by the Internet Society or its successors or assigns.

Acknowledgement