Internet Draft
File: <draft-ietf-rap-rsvp-identity-00.txt>

Satyendra Yadav
Intel
Ramesh Pabbati
Microsoft
Tim Moore
Microsoft
Shai Herzog
IPHighway

Identity Representation for RSVP

November 1998

Status of this Memo

1. Abstract

This document describes the representation of identity information in POLICY_DATA object [POL-EXT] for supporting policy based admission control in RSVP. The goal of identity representation is to allow a process on a system to securely identify the owner of the communicating process (e.g. user id) and convey this information in RSVP messages (PATH or RESV) in a secure manner. We describe the encoding of identities as RSVP policy element. We describe the processing rules to generate identity policy elements for multicast merged flows. Subsequently, we describe representations of user identities for Kerberos and Public Key based user authentication mechanisms. In summary we describe the use of this identity

information in an operational setting.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC-2119].

3. Introduction

RSVP [RFC 2205] is a resource reservation setup protocol designed for an integrated services Internet [RFC 1633]. RSVP is used by a host to request specific quality of service (QoS) from the network for particular application data streams or flows. RSVP is also used by routers to deliver QoS requests to all nodes along the path(s) of the flows and to establish and maintain state to provide the requested service. RSVP requests will generally result in resources being reserved in each node along the data path. RSVP allows particular users to obtain preferential access to network resources, under the control of an admission control mechanism. Permission to make a reservation is based both upon the availability of the requested resources along the path of the data and upon satisfaction of policy rules. Providing policy based admission control mechanism based on user identity is one of the prime requirements.

In order to solve these problems and implement user based policy control it is required to identify the user making an RSVP request. This document proposes a mechanism for sending identification information in the RSVP requests and enables authorization decisions based on policy and identity of the user requesting resources from the network.

We describe the authentication policy element (AUTH_DATA) contained in the POLICY_DATA object. User process generates an AUTH_DATA policy element and gives it to RSVP process (service) on the originating host. RSVP process inserts AUTH_DATA into the RSVP message to identify the owner (user) making the request for network resources. Network elements, such as routers, authenticate user using the credentials presented in the AUTH_DATA and admit the RSVP message based on admission policy. After a request has been authenticated, first hop router installs the RSVP state and forwards the new policy element returned by the policy server.

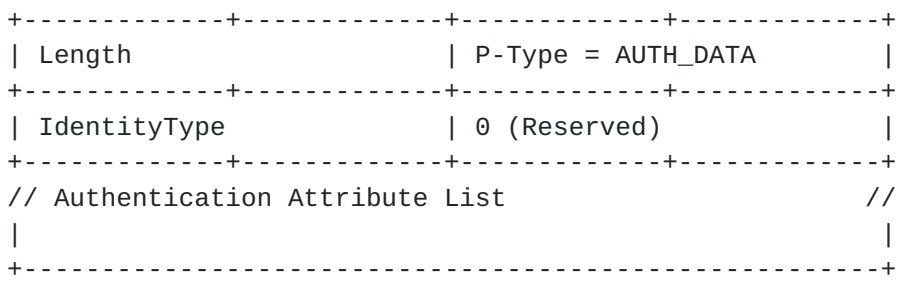
4. Policy Element for Authentication Data

4.1 Policy Data Object Format

POLICY_DATA objects contain policy information and are carried by RSVP messages. A detail description of the format of POLICY_DATA object can be found in RSVP Extensions for Policy Control [POL-EXT].

4.2 Authentication Data Policy Element

In this section, we describe a policy element (PE) called authentication data (AUTH_DATA). AUTH_DATA policy element contains a list of authentication attributes. Policy object containing AUTH_DATA MUST be protected against replay attacks using INTEGRITY object option as described in the [POL-EXT].



Length
The length of the policy element (including the Length and P-Type) is in number of octets and indicates the end of the authentication attribute list.

AUTH_DATA
Policy element type (P-type) for the authentication data as registered with Internet Assigned Numbers Authority (IANA).

IdentityType
This field describes the authentication method being used.
Following types are currently defined.

1	AUTH_USER	authentication scheme to identify users
2	AUTH_APP	authentication scheme to identify applications

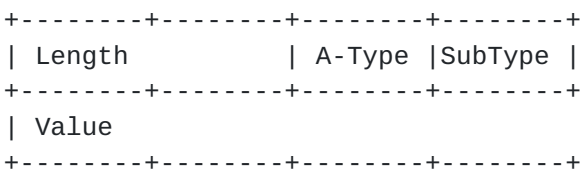
Reserved
Must be set to 0.

Authentication Attribute List

Authentication attributes contain information specific to authentication methods. The policy element provides the mechanism for grouping a collection of authentication attributes.

4.3 Authentication Attributes

Authentication attributes must be encoded as a multiple of 4 octets, attributes that are not a multiple of 4 octets long must be padded to a 4-octet boundary.



Length
The length field is two octets and indicates the actual length of the attribute (including the Length and A-Type fields) in number of octets. The length does not include any bytes padding the attribute to make it multiple of 4 octets long.

A-Type
Authentication attribute type (A-Type) field is one octet. The following values are defined:

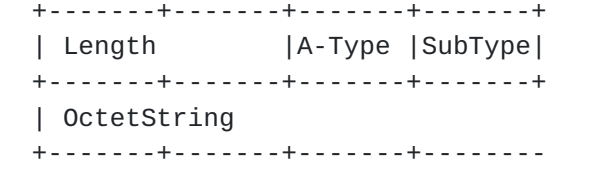
1	POLICY_LOCATOR	Unique string for locating the admission policy (such as X.500 DN described in [RFC 1779]).
2	CREDENTIAL_	User credential such as Kerberos ticket, or digital certificate. Application credential such as application ID.
3	DIGITAL_SIGNATURE	Digital signature of the authentication data policy element.

SubType
Authentication attribute sub-type field is one octet. Value of SubType depends on A-type.

Value:
The value field contains 0-65351 octets.

4.3.1 Policy Locator

POLICY_LOCATOR is used to locate the admission policy for the user or application. Distinguished Name (DN) is unique for each policy hence a DN is used as policy locator.



Length
> 4

A-Type
POLICY_LOCATOR

SubType
Following sub types for POLICY_LOCATOR are defined.

1	X500_DN	OctetString contains the X.500 DN as described in the RFC 1779 as an ASCII string.

2 UNICODE DN OctetString contains the X.500 DN described in the RFC 1779 as an UNICODE string.

3 X500_DN ENCRYPT OctetString contains the encrypted X.500 DN. The Kerberos session key or digital certificate private key is used for encryption. For Kerberos encryption the format is the same as returned from gss_seal [RFC 1509].

4 X500_UNICODE_DN ENCRYPT OctetString contains the encrypted UNICODE X.500 DN. The Kerberos session key or digital certificate private key is used for encryption. For Kerberos encryption the format is the same as returned from gss_seal [RFC 1509].

OctetString
The OctetString field contains the DN.

4.3.2 Credential

CREDENTIAL indicates the credential of the user or application to be authenticated. For Kerberos authentication method the CREDENTIAL object contains the Kerberos session ticket. For public key based authentication this field contains a digital certificate.

A summary of the CREDENTIAL attribute format is shown below. The fields are transmitted from left to right.

```
+-------+-------+-------+-------+
| Length        |A-Type |SubType|
+-------+-------+-------+-------+
| OctetString
+-------+-------+-------+--------
```

Length
> 4

A-Type
_CREDENTIAL

SubType

Following sub types for CREDENTIAL are defined.

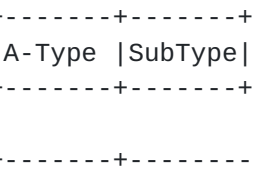
1	ASCII_ID	OctetString contains user or application identification in plain ASCII text.
2	UNICODE_ID	OctetString contains user or application identification in plain UNICODE text.
3	KERBEROS_TKT	OctetString contains Kerberos ticket.
4	X509_V3_CERT	OctetString contains X.509v3 digital certificate [X.509].
5	PGP_CERT	OctetString contains PGP digital certificate.

OctetString
The OctetString contains the user or application credential.

4.3.3 Digital Signature

The DIGITAL_SIGNATURE attribute must be the last attribute in the attribute list and contains the digital signature of the AUTH_DATA policy element. The digital signature signs all data in the AUTH_DATA policy element up to the DIGITAL_SIGNATURE. The algorithm used to compute the digital signature depends on the authentication method specified by the CREDENTIAL SubType field.

A summary of DIGITAL_SIGNATURE attribute format is described below.



Length
> 4

A-Type
DIGITAL_SIGNATURE

SubType
No sub types for DIGITAL_SIGNATURE are currently defined. This field must be set to 0.

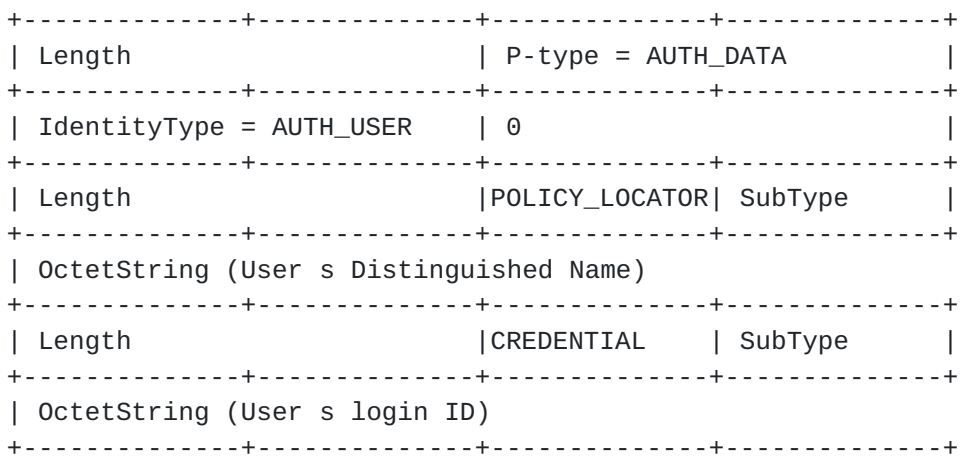
OctetString
    OctetString contains the digital signature of the AUTH_DATA.

5. Authentication Data Formats

Authentication attributes are grouped in a policy element to represent the identity credentials.

5.1 Simple User Authentication

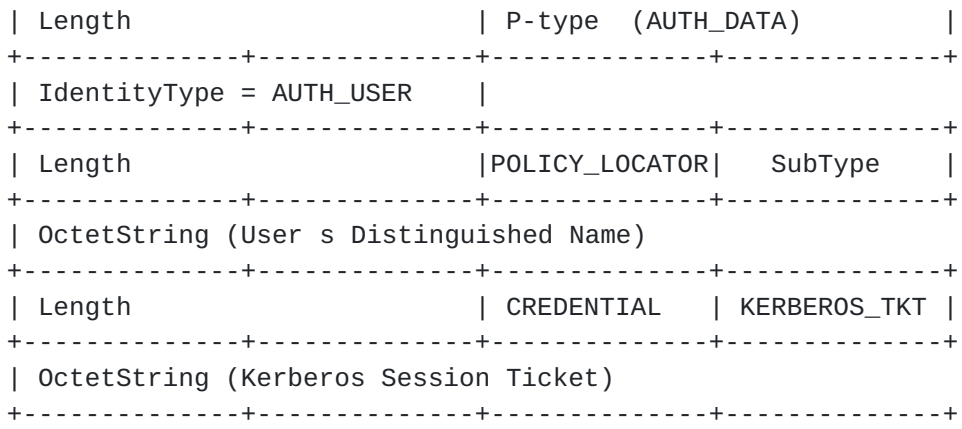
In simple user authentication method the user login ID (in plain ASCII or UNICODE text) is encoded as CREDENTIAL attribute. A summary of the simple user AUTH_DATA policy element is shown below.



5.2 Kerberos User Authentication

Kerberos [RFC 1510] authentication uses a trusted third party (the Kerberos Distribution Center KDC) to provide for authentication of the user to a network server. It is assumed that a KDC is present and both host and verifier of authentication information (router or policy server) implement Kerberos authentication.

A summary of the Kerberos AUTH_DATA policy element is shown below.

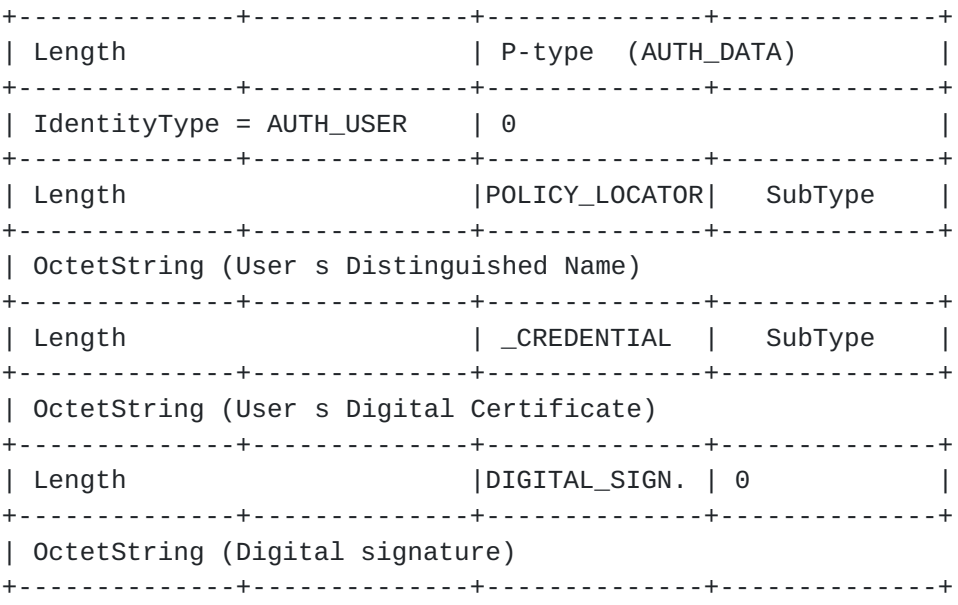


5.2.1. Operational Setting using Kerberos Identities

An RSVP enabled host is configured to construct and insert AUTH_DATA policy element into RSVP messages that designate use of the Kerberos authentication method (KERBEROS_TKT). Upon RSVP session initialization, the user application contacts the KDC to obtain a Kerberos ticket for the next network node or its policy server. A router when generating a RSVP message contacts the KDC to obtain a Kerberos ticket for the next hop network node or its policy server. The identity of the policy server or next network hop can be statically configured, learned via DHCP or maintained in a directory service. The Kerberos ticket is sent to the next network node (which may be a router or host) in a RSVP message. The router may be a policy node or may use a policy server. The KDC is used to validate the ticket and authentication the user sending RSVP message.

5.3 Public Key based User Authentication

In public key based user authentication method digital certificate is encoded as user credentials. The digital signature is used for authenticating the user. A summary of the public key user AUTH_DATA policy element is shown below.



5.3.1. Operational Setting for public key based authentication

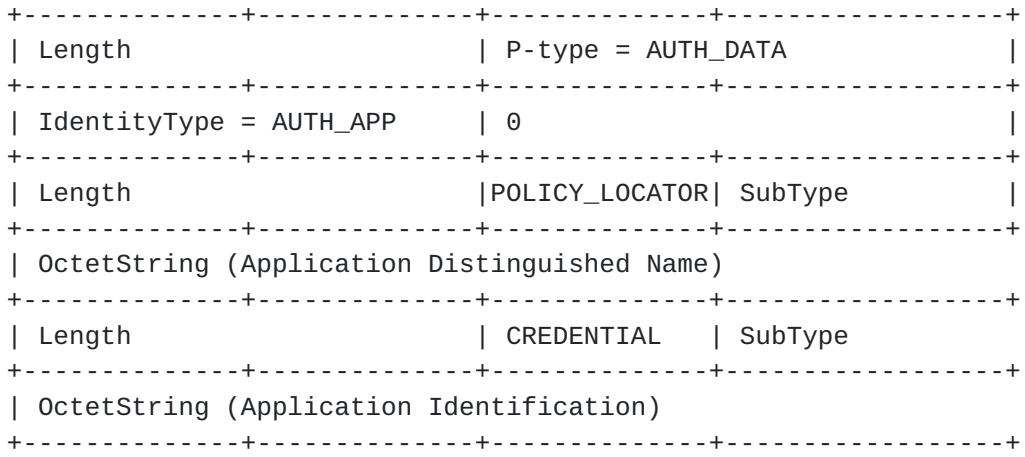
Public key based authentication assumes following:

- RSVP service requestors have a pair of keys (private key and public key).
- Private key is secured with the user.
- Public keys are stored in digital certificates and a trusted party, certificate authority (CA) issues these digital certificates.
- The verifier (policy server or router) has the ability to verify the digital certificate.

RSVP requestor uses its private key to generate DIGITAL_SIGNATURE. User Authenticators (router, policy server) use the user s public key (stored in the digital certificate) to verify the signature and authenticate the user.

5.4 Simple Application Authentication

The application authentication method encodes the application identification such as an executable filename as plain ASCII or UNICODE text.



6. Operation

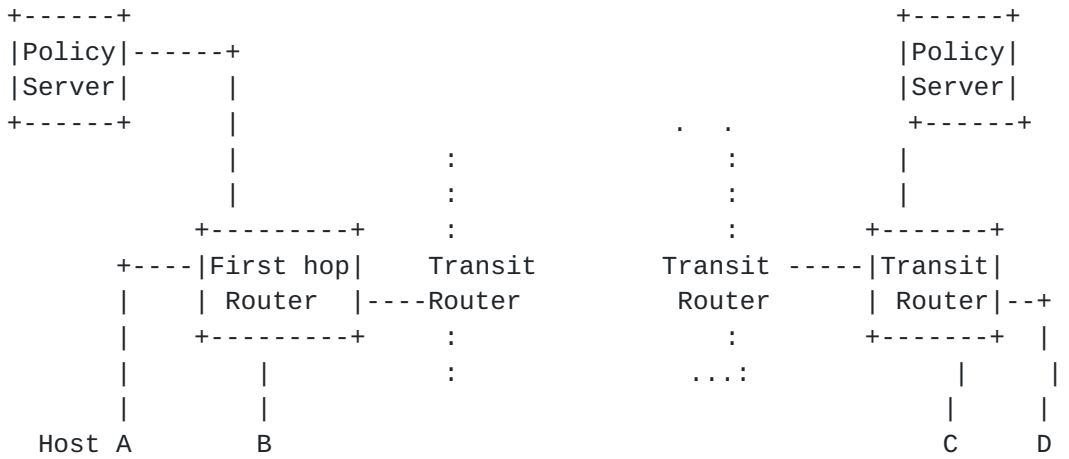


Figure 1: User and Application Authentication using AUTH_DATA object

Hosts and network nodes generate AUTH DATA policy elements, contents of which are depends on the identity type used and the authentication method used. But generally contains authentication

credentials (Kerberos ticket or digital certificate) and policy locators (which can be the X.500 Distinguished Name of the user or network node or application names). Host systems generate AUTH_DATA policy element containing the authentication identity when making the RSVP request.

Network nodes generate user AUTH_DATA policy element using the following rules

1. For unicast sessions the user policy locator is the copied from the previous hop. The authentication credentials are for the current network node identity.

2. For multicast messages the user policy locator is for the current network node identity. The authentication credentials are for the current network node.

Network nodes generate application AUTH_DATA policy element using the following rules:

1. For unicast sessions the application AUTH_DATA is the copied from the previous hop.

2. For multicast messages the application AUTH_DATA is either the first application AUTH_DATA in the message or chosen by the policy server.

7. Message Processing Rules

7.1 Message Generation (RSVP Host)

An RSVP message is created as specified in [RFC2205] with following modifications.

1. An authentication policy element, AUTH_DATA, is created and the IdentityType field is modified to indicate the authentication identity type being used.

   - A DN is inserted as POLICY_LOCATOR attribute.

   - Credentials such as Kerberos ticket or digital certificate

are inserted as the CREDENTIAL attribute.

2. A POLICY_DATA object is inserted in the RSVP message in appropriate place with AUTH_DATA as one of the policy elements. If INTEGRITY object is not computed for the RSVP message then an INTEGRITY object MUST be computed for this POLICY_DATA object, as described in the [POL_EXT], and MUST be inserted as an option.

7.2 Message Reception (Router/Subnet Bandwidth Manager (SBM))

RSVP message is processed as specified in [RFC2205] with following modifications.

1. If router/SBM is not policy aware then it SHOULD send the RSVP message to the policy server and wait for response.

2. Reject the message if the response from the policy server is negative.

3. Continue processing the RSVP message.

7.3 Authentication (Router/SBM/Policy server)

1. Retrieve the AUTH_DATA policy element.

2. Check the IdentityType field and return an error if the identity type is not supported.

3. Verify credential

   - Simple authentication: e.g. Get user ID and validate it, or get executable name and validate it.

   - Kerberos: Send the Kerberos ticket to the KDC to obtain the session key. Using the session key authenticate the user.

   - Public Key: Validate the certificate that it was issued by a trusted Certificate Authority (CA) and authenticate the user or application by verifying the digital signature.

8. Security Considerations

The purpose of this draft is to describe a mechanism to authenticate

RSVP requests based on user identity in a secure manner. RSVP INTEGRITY object [MD5] is used to protect the policy object containing user identity information from security (replay) attacks. Combining the AUTH_DATA policy element and the INTEGRITY object results in a secure access control that enforces authentication based on both the identity of the user and the identity of the originating node.

Simple authentication does not contain credential that can be securely authenticated and is inherently less secured.

The Kerberos authentication mechanism is reasonably well secured.

User authentication using a public key certificate is known to provide the strongest security.

9. Error Signaling

If Policy server fails to verify the AUTH_DATA policy element then it must indicate to the first hop router the Error Code = 02 (Policy control failure). The policy server may specify error value field. These typically include:

- Authentication method not supported

- Authentication failure
- Required attribute (specify) missing. For example CREDENTIAL attribute missing.
- Unknown attribute (specify) type.
- Unknown attribute (specify) sub type.

10. Appendix A

Internet-Draft RSVP Cryptography Authentication [MD5] describes a mechanism to authenticate RSVP messages based on the interface of a node on which the RSVP messages are sent. A single key is used to authenticate RSVP requests made by all users for an interface of the node. The security of such a setup has following limitations:

- Authentication and RSVP admission control on the basis of

node credentials alone is less secure as a node can impersonate all of its users even when they are not logged in.

- It is not possible to identify the user making the RSVP request.

Mechanism described in this draft can be used to solve the key distribution problem between hosts and routers for implementing [MD5]. We call this key the integrity key and if a network node is using authentication information, it may obtain this integrity key from a key distribution center by using the authentication information. A network node can verify the Integrity object of the RSVP message by using the authentication credentials to obtain the key from a key distribution center. The integrity key is used to compute the messages digest in the INTEGRITY object. The next hop router may either be a policy node; a policy server client or it may obtain the integrity key from a key distribution center using the authentication credentials from the first AUTH_DATA policy element. The key management APIs described in [MD5] needs to use the whole message to obtain the necessary information to obtain the key for integrity verification from a key distribution center.
The key to generate integrity object for PATH and RESV tear should use the same key that was used to generate integrity objects for the PATH or RESV message.

Using the identity policy elements to find the integrity key does not work for the last hop for multicast RSVP. This requires authenticated multicast joins to allow the control of who can send RSVP messages for a multicast group. The certificate authority can issue secret keys for use between the requestor and the verifier. This secret key is used for the integrity key.

If the policy server or another key server supplies an integrity key then check the integrity object. The router or host for later use (e.g., on refreshes) may cache this key. If a policy server obtains the integrity key on behalf of the router it can send the key using cops [COPS-RSVP].

11. Acknowledgments

We would like to thank Raj Yavatkar, Bob Linden and many others for their valuable comments on this draft.

12. References

[ASCII] Coded Character Set -- 7-Bit American Standard Code for Information Interchange, ANSI X3.4-1986.

[MD5] Baker, F., et.al. "RSVP Cryptographic Authentication." Internet-Draft, draft-ietf-rsvp-md5-07.txt, November 1998.

[POL-EXT] Herzog, S., "RSVP Extensions for Policy Control." Internet-Draft, draft-ietf-rap-policy-ext-00.txt, April 1998

[RFC 1510] The Kerberos Network Authentication Service (V5). Kohl J., Neuman, C. RFC 1510.

[RFC 1704] On Internet Authentication. Haller, N, Atkinson, R., RFC 1704.

[RFC 1779] A String Representation of Distinguished Names. S. Kille. RFC 1779

[RFC 2205] Braden, R., et. al., "Resource ReSerVation Protocol (RSVP) Version 1 Functional Specification." RFC 2205.

[RFC 2209] Braden, R., Zhang, L., "Resource ReSerVation Protocol (RSVP) - Version 1 Message Processing Rules." RFC 2209.

[UNICODE] The Unicode Consortium, "The Unicode Standard, Version 2.0", Addison-Wesley, Reading, MA, 1996.

[X.509] ITU-T (formerly CCITT) Information technology Open Systems Interconnection The Directory: Authentication Framework Recommendation X.509 ISO/IEC 9594-8

13. Author Information

Satyendra Yadav
Intel, JF3-206
2111 NE 25th Avenue
Hillsboro, OR 97124
Satyendra.Yadav@intel.com

Ramesh Pabbati
Microsoft
1 Microsoft Way
Redmond, WA 98054
rameshpa@microsoft.com

Tim Moore
Microsoft
1 Microsoft Way
Redmond, WA 98054
timmoore@microsoft.com

Shai Herzog
IPHighway
2055 Gateway Pl., Suite 400
San Jose, CA 95110
herzog@iphighway.com