Internet Draft
File: <draft-ietf-rap-user-identity-00.txt>
Expiration: September 13, 1998

Satyendra Yadav
Intel
Ramesh Pabbati
Microsoft
Peter Ford
Microsoft
Shai Herzog
IPHighway

User Identity Representation for RSVP

March 13, 1998

Status of this Memo

1. Abstract

This document describes the representation of user 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 requests (PATH or RESV) in a secure manner. We describe the encoding of user identity as RSVP policy element. 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. 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.

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.

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 identifying the user making RSVP requests and make authorization decisions such as policy based admission control based on the identity of the user requesting resources from the network.

We describe the user 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 the user based on admission policies and identity information presented in the AUTH_DATA before admitting the RSVP massage.

We also describe a mechanism to solve the key distribution problem between hosts and the first hop router for implementing [MD5]. We define a new C-type for INTEGRITY [MD5] object to indicate that the key used to generate the message digest for INTEGRITY object is in the AUTH_DATA policy element. Policy server will return this key

after authenticating the user. First hop router then verifies the integrity check of the RSVP message.

2.1 Operation

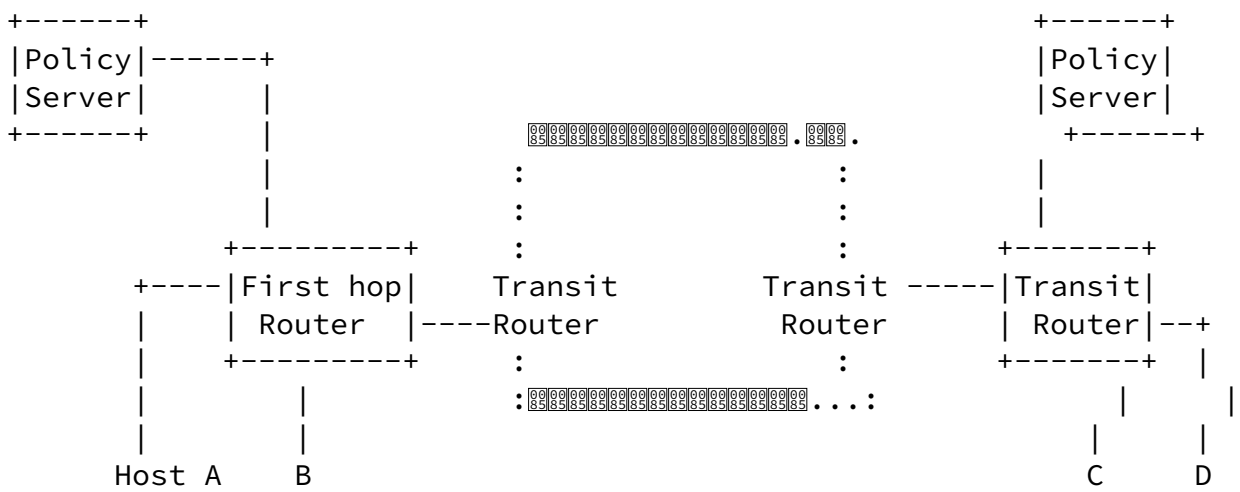


Figure 1: User authentication using AUTH_DATA and INTEGRITY object

Host systems generate AUTH_DATA policy element. User key may be used to compute the messages digest of the INTEGRITY object. The First hop router send the policy data object to the policy server. Policy server authenticates the user and returns the user key to the first hop router. First hop router then verifies the integrity object to check the integrity of the RSVP message. After a user has been authenticated, first hop router installs the RSVP reservation and forwards the new policy element returned by the policy server.

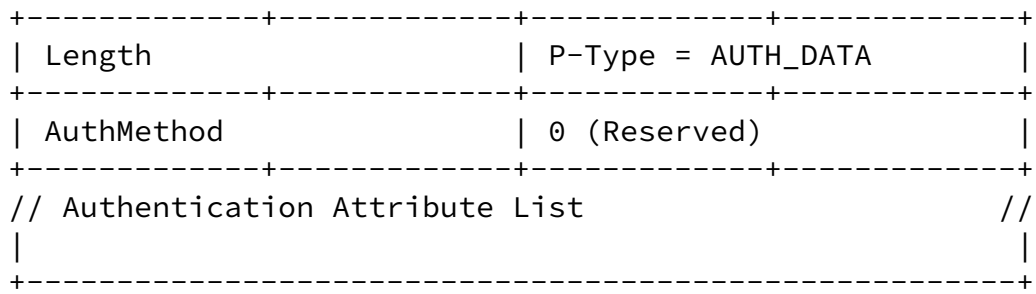
3. Policy Element for User Authentication

3.1 Policy Data Object Format

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

3.1 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.



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. Value to be defined.

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

1 AUTH_SIMPLE Simple authentication scheme with no user

credentials

2 AUTH_KERBEROS Kerberos V5 authentication protocol

3 AUTH_PKEY_DC Public-Key authentication using Digital Certificate

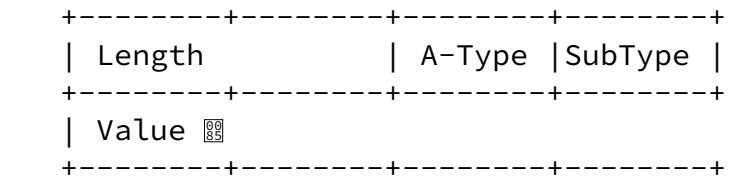
Reserved
Must be set to 0.

Authentication Attribute List

Authentication attributes contain information specific to authentication methods. The policy element provides the mechanism for grouping a collection of authentication attributes. For example, AUTH_DATA may contain a canonical user identifier (e.g. user id) and an authenticator for that user (e.g. kerberos ticket).

3.1.1 Authentication Attribute List

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 USER_ID User's network (such as NT Domain) login ID.

2 USER_DN User's Distinguished Name (DN) such as X.500 DN described in [RFC 1779].

3 USER_CRED User's credentials, such as Kerberos ticket, or digital certificate.

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

Value:
The A-Type, SubType and Length fields determine the format of the Value field.

The format of the value field is one of following data types.

OctetString 0-65351 octets long

TimeStamp 32 bit value, seconds since 00:00:00 GMT January 1 1970

3.1.2 User ID

USER_ID attribute indicates the user's network login ID (e.g. NT Domain\username). A summary of the USER_ID attribute format is shown below.

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


Length
> 4

A-Type
USER_ID

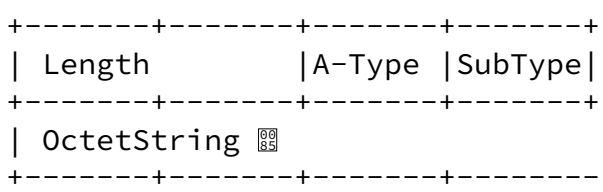
SubType
Following sub types for USER_ID are defined.

1	ASCII	OctetString contains the user ID as a ASCII string [ASCII].
2	UNICODE	OctetString contains the user ID as an UNICODE string [UNICODE].
3	ASCII_ENCRYPT	Octet OctetString contains encrypted ASCII user ID. The user key (kerberos session key or private key) is used for encryption.
4	UNICODE_ENCRYPT	Octet OctetString contains encrypted UNICODE user ID. The user key (kerberos session key or private key) is used for encryption.

OctetString
The OctetString field contains the user ID.

3.1.3 User Distinguish Name

USER_DN indicates the string representation of the user's Distinguished Name (DN).



Length
> 4

A-Type
USER_DN

SubType
Following sub types for USER_DN are defined.

1	X500	OctetString contains the user's X.500 DN as described in the RFC 1779.

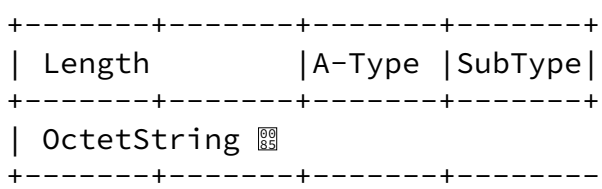
2 X500_ENCRYPT OctetString contains the user's encrypted X.500 DN. The user key (kerberos session key or private key) is used for encryption.

OctetString
The OctetString field contains the user's DN.

3.1.4 User Credentials

USER_CRED indicates the credentials of the user to be authenticated. This attribute is not used for simple authentication mechanism. For kerberos authentication method the USER_CRED object contains the kerberos session ticket. For public key based authentication this field contains the user certificate.

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



Length
> 4

A-Type
USER_CRED

SubType
Following sub types for USER_CRED are defined.

1 KERBEROS_TKT OctetString contains kerberos ticket.

2 X509_V3_CERT OctetString contains X.509v3 digital certificate [X.509].

3 PGP_CERT OctetString contains PGP digital certificate.

OctetString
The OctetString contains the user credentials.

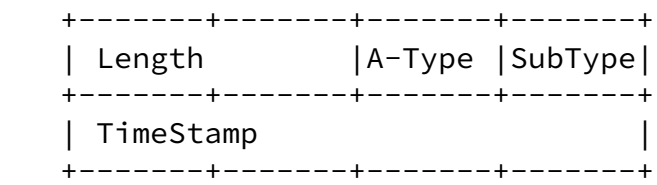
3.1.5 Time Stamp

This attribute indicates the time when the authentication policy element was created. This attribute is used to prevent the replay of authentication information. Policy server will reject a message if

it contains a TIME_STAMP that does not fall in the time window

(current time - delta, current time + delta). The delta has to be carefully chosen to allow for time inconsistencies between systems and message delivery time. This is not a big problem if Network Time Protocol (NTP) is used to synchronize clocks.

A summary of TIME_STAMP attribute format is described below.



Length
8

A-Type
Value of 8 for TIME_STAMP

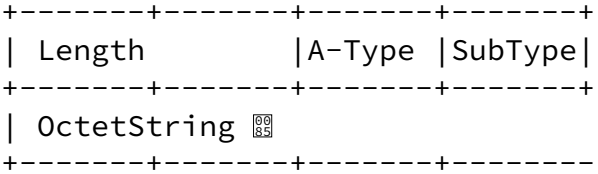
SubType
No sub types for TIME_STAMP are currently defined. This field must be set to 0.

TimeStamp
Time when authentication policy element was created.

3.1.6 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 AuthMethod 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.

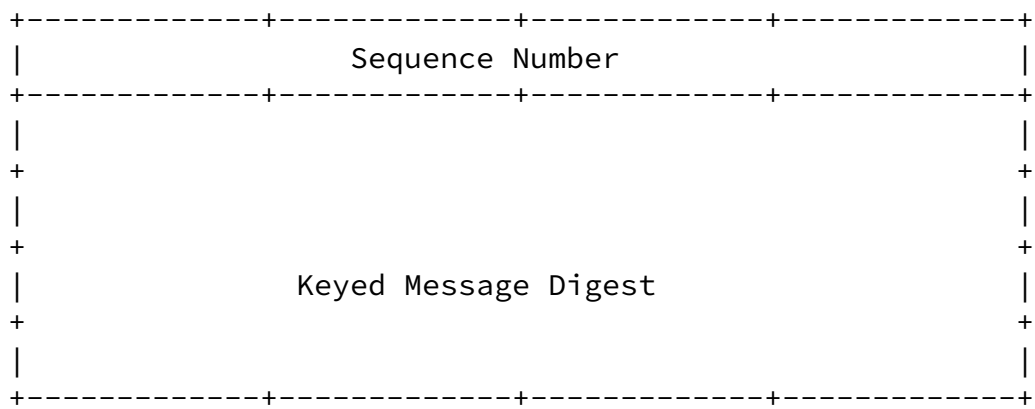
OctetString
    OctetString contains the digital signature of the AUTH_DATA.

4. INTEGRITY object

The INTEGRITY object is carried in the RSVP message to verify the integrity of the message on hop-by-hop basis. A detail description of the format of INTEGRITY object can be found in "RSVP Cryptographic Authentication" [MD5].

In this section, we describe a new type (C-Type =3) of INTEGRITY object to indicate that the key used to compute the message digest is in the AUTH_DATA policy element.

Policy Keyed Message Digest, INTEGRITY Object: Class = 4, C-Type = 3



The host system uses the user key to generate the message digest of the INTEGRITY object. The user key is transmitted in the AUTH_DATA

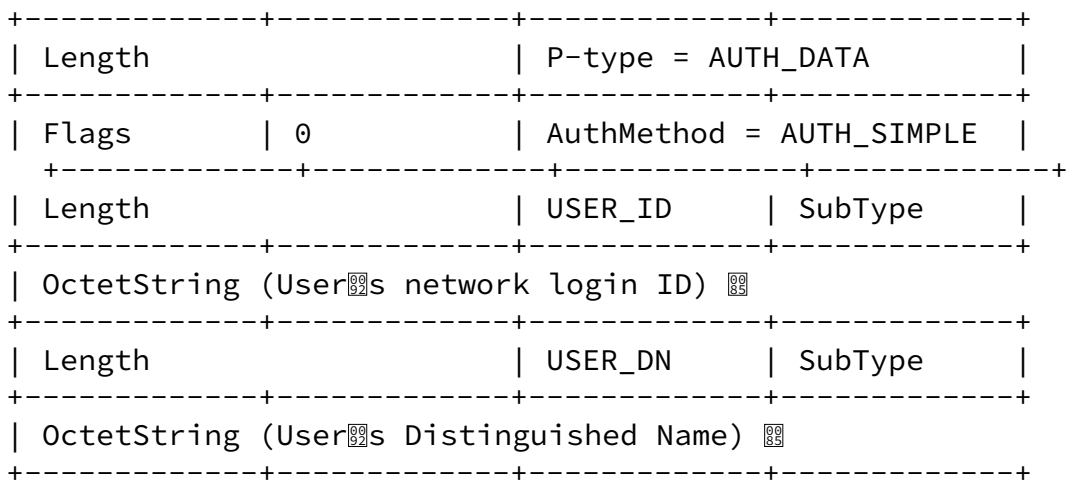
policy element. The message digest is computed according to the rules presented in section 5, "Message Processing Rules".

5. Authentication Data Formats

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

5.1 Simple Authentication

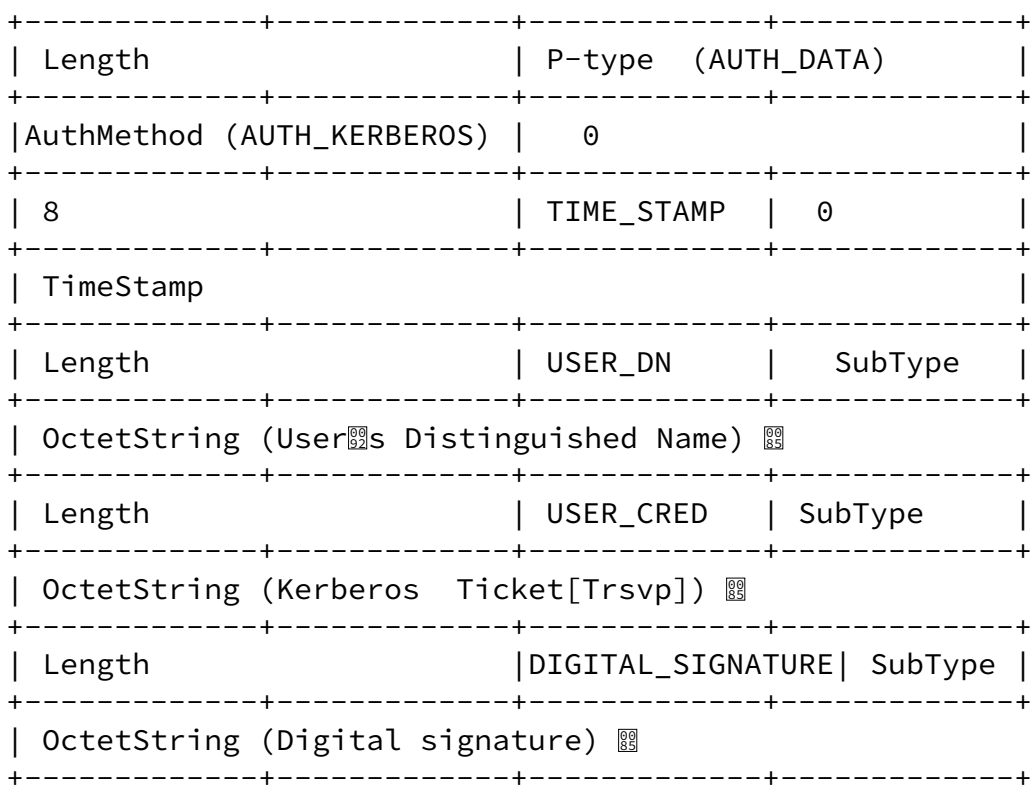
The AUTH_SIMPLE method simply encodes the user identification such as a logon id. The USER_DN attribute is optional. Simple authentication does not encode any credentials.



5.2 Kerberos 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. The USER_DN attribute is optional. TIME_STAMP and DIGITAL_SIGNATURE attribute may be omitted when INTEGRITY object is being inserted.

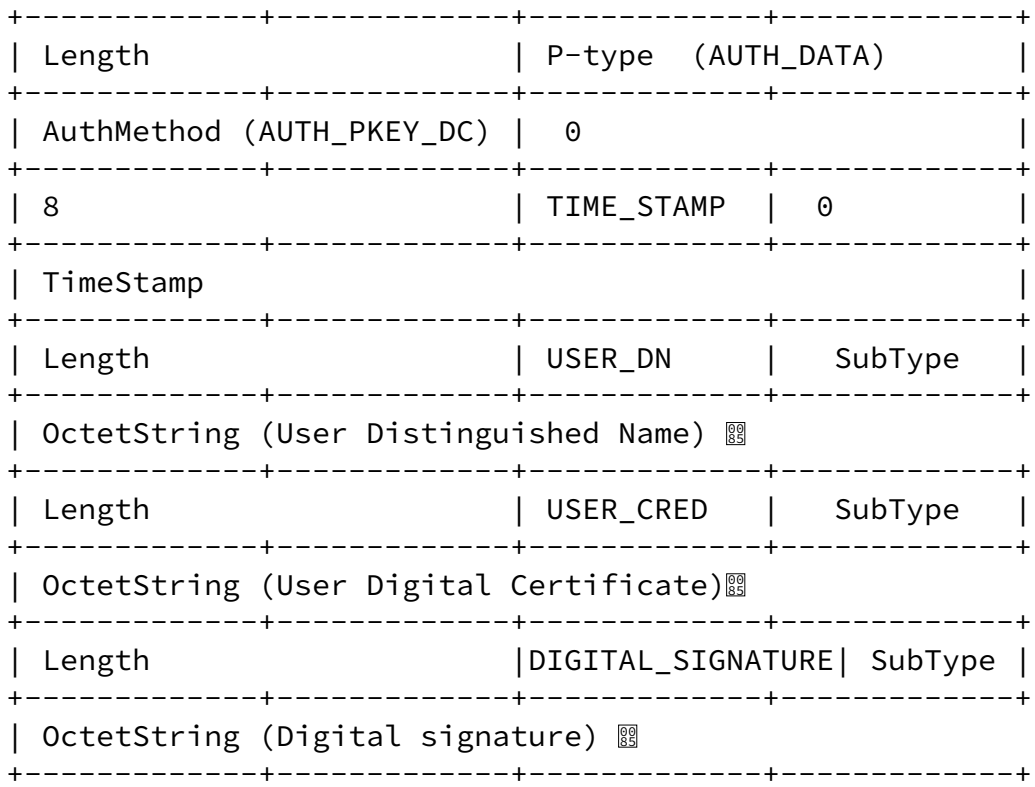


5.2.1. Operational Setting using Kerberos Identities

An RSVP enabled host is configured to construct and insert RSVP policy elements into RSVP messages that designate use of the kerberos authentication method (AUTH_KERBEROS). Upon RSVP session initialization, the user application contacts the KDC to obtain a kerberos ticket (Trsvp) for the policy server configured in the host. The identity of the policy server can be statically configured, learned via DHCP or maintained in a directory service. Trsvp is encrypted using a shared secret key that is known by the policy server and the KDC. Host will send the Trsvp to the first hop router in a RSVP message. Router in turn sends the Trsvp to the policy server. Policy server decrypts the Tsrvp, interprets the ticket and identifies the user requesting RSVP reservation. The Kerberos ticket contains the session key (Sk), which policy server sends to the router for verification of the INTEGRITY (C-Type = 3) object. The policy server may use the session key to decrypt the user authentication attributes such as USER_ID or USER_DN.

5.3 Public Key based Authentication

A summary of the public key AUTH_DATA policy element is shown below. The USER_DN attribute is optional. TIME_STAMP and DIGITAL_SIGNATURE attribute may be omitted when INTEGRITY object is being inserted.



5.3.1. Operational Setting for public key based authentication

Public key based authentication assumes following:

- Users making RSVP requests 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 (the policy server) has the ability to verify the digital certificate.

RSVP host uses user's private key to encrypt USER_ID and USER_DN and generate DIGITAL_SIGNATURE. Policy server uses the user's public key (stored in the digital certificate) to authenticate user and decrypt user information.

6. Message Processing Rules

6.1 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 AuthMethod field is modified to indicate the authentication method being used.

   - User credential (kerberos ticket or digital certificate) is inserted as USER_CRED attribute.

   - Optionally user DN is inserted as USER_DN attribute.

   - If INTEGRITY object is not being inserted then AUTH_DATA must contain the TIME_STAMP and DIGITAL_CERTIFICATE attributes.

2. A POLICY_DATA object is inserted in the RSVP message in appropriate place with AUTH_DATA as one of the policy elements.

3. Optionally, an INTEGRITY object is created and inserted in RSVP message as specified in [MD5] with following modifications:

   - User key (session key for kerberos, private key for pubic key based authentication) may be used to generate keyed message digest for INTEGRITY object (C-type = 3).

   - AUTH_DATA is included in the computation of message digest.

6.2 Message Reception (Router/SBM)

RSVP message is processed as specified in [RFC2205] and [MD5] with

following modifications.

1. If INTEGRITY object is present and its C-Type is 3 then postpone the verification of the INTEGRITY object until a response from policy server is received. Otherwise verify the INTEGRITY object using shared key and jump to step 4.

2. Send the policy object to the policy server and wait for response.

3. Reject the message if the response from the policy server is negative. For C-Type 3 INTEGRITY object, verify the message integrity using the key returned by the policy server. The router for later use (e.g., on refreshes) may cache this key.

4. Continue processing the RSVP message.

6.3 User Authentication (Policy server)

1. Retrieve the AUTH_DATA policy element.

2. Check the AuthMethod field and return an error if the authentication method is not supported.

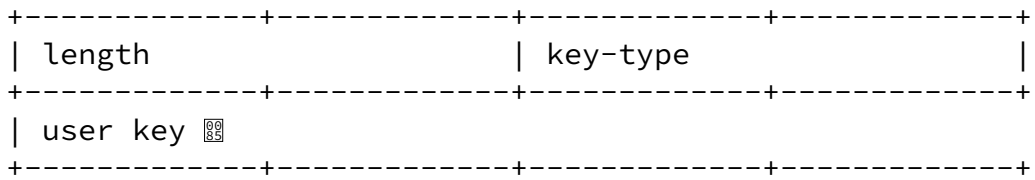
3. Verify USER_CRED

   - Public Key: Validate the user certificate that it was issued by a trusted Certificate Authority (CA) and identify the user.

   - Kerberos: Verify the session ticket and identify the user.

   - DIGITAL_SIGNATURE is verified if present.

4. If user authentication was successful then send the user key (kerberos session key or public key) to the router. The format of the key returned is described below.



length
    length of the key object.

key-type
1 KERBEROS kerberos session key
2 PUBLIC_KEY User's public key

user-key
User key present in the AUTH_DATA policy element.

7. Security Considerations

The purpose of this draft is to describe a mechanism to authenticate RSVP requests based on user identity in a secure manner. [MD5] specifies how an RSVP message can be digitally signed and checked by each RSVP service node including an initial signature that can be applied by the originating host. We propose extensions to INTEGRITY and POLICY_DATA object to provide a strong 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 user credential and is inherently less secure. This method is inherently insecure.

The kerberos authentication mechanism is reasonably well secured.

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

8. 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

- User Authentication failure

- Required attribute (specify) missing. For example USER_CRED attribute missing.

- Unknown attribute (specify) type.

- Unknown attribute (specify) sub type.

9. References

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

[LPM] Herzog, S., "Local Policy Modules (LPM): Policy Control for RSVP." Internet-Draft, draft-ietf-rsvp-policy-lpm-1.ps, November 1996.

[MD5] Baker, F., "RSVP Cryptographic Authentication." Internet-Draft, draft-ietf-rsvp-md5-05.txt, May 1997.

[POL-EXT] Herzog, S., "RSVP Extensions for Policy Control." Internet-Draft, draft-ietf-rap-policy-ext-04.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

10. Glossary

Authentication
: The process of determining the identity (usually the name) of the other party in some communication exchange.

Certificate
: The public key of a particular principal, together with some other information relating to the names of the principal and the certifying authority, rendered unforgeable by enciphering with the private key of the certification authority that issued it.

Certification Authority (CA)
: An authority trusted by one or more principals to create and assign certificates.

Credentials
: Information required by principals in order to for them to authenticate. Credentials may contain information used to initiate the authentication process, information used to respond to an authentication request (verifier information), and cached information useful in improving performance.

Digital Signature
: A value computed from a block of data and a key that could only be computed by someone knowing the key. A digital signature computed with a secret key can only be verified by someone knowing that secret key. A digital signature computed with a private key can be verified by anyone knowing the corresponding public key.

Private key
: Cryptographic key used in asymmetric (public key) cryptography to decrypt and/or sign messages. In asymmetric cryptography, knowing the encryption key is independent of knowing the decryption key. The decryption (or signing) private key cannot be derived from the encrypting (or verifying) public key.

Public key
: Cryptographic key used in asymmetric cryptography to encrypt messages and/or verify signatures.

Secret key
: Cryptographic key used in symmetric cryptography to encrypt, sign, decrypt and verify messages. In symmetric cryptography, knowledge of the decryption key implies knowledge of the encryption key, and vice-versa.

Sign
: A process which takes a piece of data and a key and produces a digital signature which can only be calculated by someone with the key. The holder of a corresponding key can verify the signature.

Strong authentication
: Authentication by means of cryptographically derived authentication tokens and credentials. The actual working definition is closer to that of "zero knowledge" proof: authentication so as to not reveal any information usable by either the verifier, or by an eavesdropping third party, to further their potential ability to impersonate the claimant.

Ticket
: A data structure certifying an authenticating (public) key by virtue of being signed by a user principal using their (long term) private key. The ticket also includes the UID of the principal.

Trusted authority
: The public key, name and UID of a certification authority trusted in some context to certify the public keys of other principals.

Verify
: To cryptographically process a piece of data and a digital signature to determine that the holder of a particular key signed the data.

11. Acknowledgements

We would like to thank Raj Yavatkar and Scott Hahn for their valuable comments on this draft.

12. Author Information

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

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

Peter Ford
Microsoft
1 Microsoft Way
Redmond, WA 98054
Peterf@microsoft.com

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