RSVP Cryptographic Authentication
draft-ietf-rsvp-md5-00.txt

Wed Apr 12 17:18:31 PDT 1995

Fred Baker
Cisco Systems
fred@cisco.com

Status of this Memo

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet Drafts as reference material or to cite them other than as a "work in progress".

We expect that the important contents of this draft, perhaps in a modified form, will be included in the RSVP specification as an appendix at some point in the future. For the moment, it is published separately as a "talking paper," if for no good reason, then because it is easy for the author to do so.

The author notes that the paper derives directly from similar work done for OSPF and RIP Version II. This was done jointly by Ran Atkinson and Fred Baker. The base draft is also being modified by Dino Farinacci for IDMR. The use of this form of digest and hop-by-hop authentication is consistent with the expectations of the Security Area, and should be consistent with the IETF's Key Management Procedures when these are more fully defined.

1. Introduction

Growth in the Internet has made us aware of the need for improved authentication of routing information. This also applies to resource reservations; we are aware of a need for authentication of those who reserve bandwidth and timing resources, to counter "denial of reservable resources" attacks. RSVP provides a simple object to hold this authentication information, but the contents and use of the object are at this point undefined.

Without authentication, then only simple misconfigurations are detected. Simple passwords transmitted in the clear will further protect against the honest neighbor, but are useless in the general case. By simply capturing information on the wire - straightforward even in a remote environment - a hostile process can learn the password and overcome the network.

RSVP should use an authentication algorithm such as the algorithm specified in SNMP Version 2, augmented by a non-decreasing sequence number. MD5 is proposed as the standard authentication algorithm for RSVP, but the mechanism is intended to be algorithm-independent. While this mechanism is not unbreakable (no known mechanism is), it provides a greatly enhanced probability that a system being attacked will detect and ignore hostile messages. This is because we transmit the output of an authentication algorithm (e.g., MD5) rather than the secret Authentication Key. This output is a one-way function of a message and a secret Authentication Key. This Authentication Key is never sent over the network in the clear, thus providing protection against the passive attacks now commonplace in the Internet.

In this way, protection is afforded against forgery or message modification. It is possible to replay a message until the sequence number changes, but the sequence number makes replay in the long term less of an issue. The mechanism does not afford confidentiality, since messages stay in the clear; however, the mechanism is also exportable from most countries, which test a privacy algorithm would fail.

Other relevant rationales for the approach are that MD5 is used in SNMP Version 2, and is therefore present in routers already, as is some form of password management. A similar approach has been proposed for authentication in IP version 6 (IPv6).

2. Implementation Approach

Implementation requires three issues to be addressed:

(1) Definition of the appropriate object,

(2) Authentication procedures, and

(3) Management controls.

2.1. Object Format

The RSVP Message consists of a sequence of "objects," which are type-length-value encoded fields having specific purposes. The current draft defines two separate security concerns, and two separate objects to address them. The contents of both of these objects is undefined. They are, according to sections 2.5 and 4:

(1) Protecting RSVP Message Integrity

It may be necessary to ensure the integrity of RSVP messages against corruption or spoofing, hop by hop. RSVP messages have an optional integrity field that can be created and verified by neighboring RSVP nodes.

(2) Authenticating Reservation Requests

RSVP-mediated resource reservations may reserve network resources, providing special treatment for a particular set of users. Administrative mechanisms will be necessary to control who gets privileged service and to collect billing information. These mechanisms may require secure authentication of senders and/or receivers responsible for the reservation. RSVP messages may contain credential information to verify user identity.

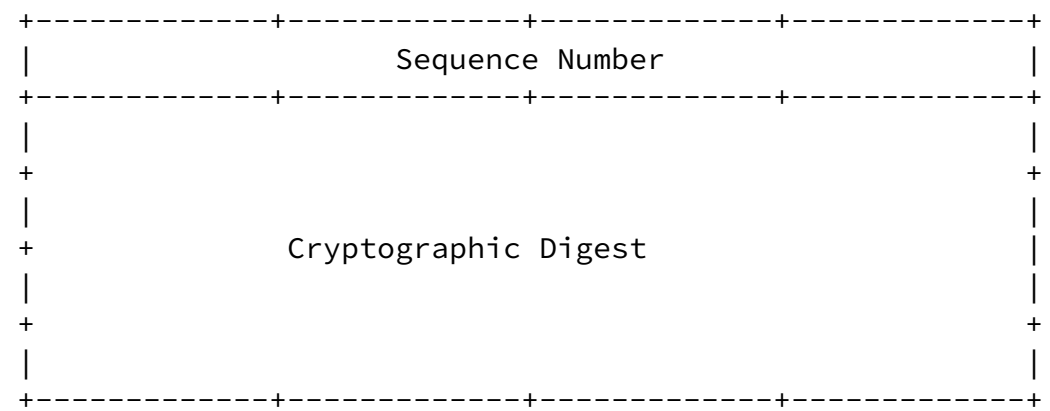
The author submits that, since reservations in RSVP are made hop by hop, and routers in transit networks are generally unaware of the individual users and data flows they support, end to end reservation authentication doesn't make sense. Rather, hop by hop authentication, with the assumption that authentication is transitive (the same assumption made by authenticated routing protocols) makes more sense. In addition, the effort that the sender of an RSVP message will put into its creation, if an MD5 digest is used, far exceeds the significant effort that goes into a UDP checksum; we propose that the checksums can be disabled if MD5 authentication is used, as the MD5 digest is a much stronger integrity

check.

In short, let's solve the several problems with a single solution. For the present, let us assume that it is the INTEGRITY OBJECT is made to serve the purpose.

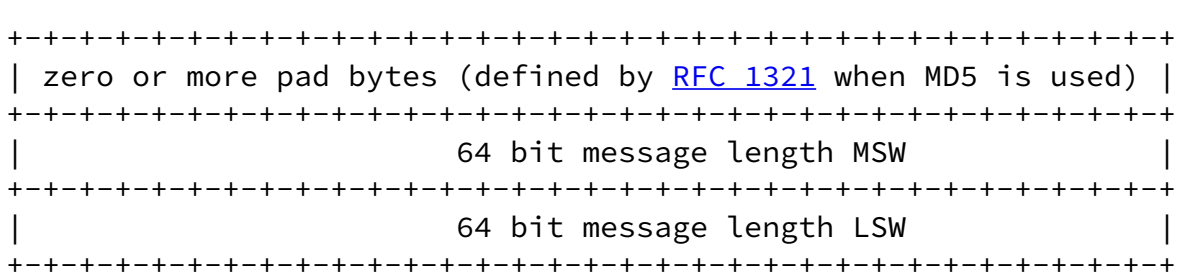
The INTEGRITY object is restructured as a "Keyed Message Digest" object. This consists in four fields:

LENGTH Length of the INTEGRITY object
CLASS INTEGRITY=13
TYPE 0 <= key number <= 255
VALUE
octets 0-3: An unsigned 32 bit non-decreasing sequence number.
octets 4- : the digest, which must be a multiple of four octets long.



2.2. Processing Algorithm

In memory, the following trailer is appended by the MD5 algorithm and treated as though it were part of the message.



2.2.1. Message Generation

The RSVP Packet is created as usual, with these exceptions:

(1) The RSVP checksum is not calculated, but is set to zero. (The UDP Checksum need not be calculated either, but this is often out of the UDP client's hands).

(2) The INTEGRITY object is inserted in its accustomed place, and it's location in the message remembered for later use.

(3) The current sequence number and authentication key are placed in the INTEGRITY object. If several messages are being created simultaneously (for example, in a periodic refresh generated by a router), the messages may all use the same sequence number. This is to assure that message reordering between RSVP peers does not cause authentication to fail.

The value used in the sequence number is arbitrary, but two suggestions are the time of the message's creation or a simple message counter.

The RSVP Authentication Key is selected by the sender based on the outgoing interface or neighbor. Multicast PATH messages will need to be keyed by interface, while unicast messages may by keyed by neighbor. As a simplifying assumption, we propose that immediately adjacent systems key by interface, and non-adjacent systems key by neighbor id.

Each key has a lifetime associated with it. No key is ever used outside its lifetime. If more than one key is currently alive, then the youngest key (the key whose lifetime most recently started) SHOULD be used. Since the key's algorithm is an attribute of the key, stored in the sender and receiver along with it, the Key ID effectively indicates which authentication algorithm is in use if the implementation supports more than one authentication algorithm.

(1) Trailing pad and length fields are added and the digest calculated using the indicated algorithm. When MD5 is the algorithm in use, these are calculated per RFC 1321.

(2) The digest is written over the RSVP Authentication Key in the INTEGRITY object. When MD5 is used, this digest will be 16 bytes long.

The trailing pad is not transmitted, as it is entirely predictable from the message length and algorithm in use.

2.2.2. Message Reception

When the message is received, the process is reversed:

(1) The RSVP Checksum is not calculated,

(2) The digest is set aside,

(3) The appropriate algorithm and key are determined from the INTEGRITY object's type field,

(4) The RSVP Authentication Key is written into the value field of the INTEGRITY object.

(5) Appropriate padding is added as needed, and

(6) A new digest calculated using the indicated algorithm.

If the calculated digest does not match the received digest, the message is discarded unprocessed. If the received sequence number is less than the last sequence number received, the message is also discarded. Ideally, the sending sequence number is stored in non-volatile memory, so that it survives resets. However, if a device has not validly spoken for some time and starts with a low sequence number, it would be advisable to accept its view of the world.

3. Management Procedures

3.1. Key Management Requirements

It is strongly desirable that a hypothetical security breach in one Internet protocol not automatically compromise other Internet protocols. The Authentication Key of this specification SHOULD NOT be stored using protocols or algorithms that have known flaws or fail to afford perfect forward secrecy.

Implementations MUST support the storage of more than one key at the same time, although it is recognized that only one key will normally be active on an interface. They MUST associate a specific lifetime (i.e., date/time first valid and date/time no longer valid) with each key, the key identifier, and MUST support manual key distribution (e.g., the privileged user manually typing in the key, key lifetime, and key identifier on the console). The lifetime may be infinite. If more than one algorithm is supported, then the implementation MUST require that the algorithm be specified for each key at the time the other key

information is entered. Keys that are out of date MAY be deleted at will by the implementation without requiring human intervention. Manual deletion of active keys SHOULD also be supported.

Note that there are four "times" that are important concerning a key:

+ The time the system starts accepting received packets signed with the key (KeyStartReceive).
+ The time the system starts signing packets with the key (KeyStartSign).
+ The time the system stops signing packets with the key, which is to say, the time it starts signing with the next key (KeyStopSign).
+ The time the system stops accepting received packets signed with the key (KeyStopReceive).

The times SHOULD be in the order listed, which is to say that none of these times occurs before the one mentioned before it. There needs to be some distance between starts and between stops to get a seamless transition. Each system sends with whichever key has the most recent "start" time, and makes its first attempt at validation of incoming traffic with the same key. If this validation fails and another (older) key is also active, the system should attempt to validate with any other active keys it may possess.

Note that the concept of a "key lifetime" does not require a hardware time of day clock or the use of NTP, although one or the other is advised; it merely requires that the earliest and latest times that the key is valid must be programmable in a way the system understands.

It is likely that the IETF will define a standard key management protocol. It is strongly desirable to use that key management protocol to distribute RSVP Authentication Keys among communicating RSVP implementations. Such a protocol would provide scalability and significantly reduce the human administrative burden. The Key ID can be used as a hook between RSVP and such a future protocol. Key management protocols have a long history of subtle flaws that are often discovered long after the protocol was first described in public. To avoid having to change all RSVP implementations should such a flaw be discovered, integrated key management protocol techniques were deliberately omitted from this specification.

3.2. Key Management Procedures

As with all security methods using keys, it is necessary to change the RSVP Authentication Key on a regular basis. To maintain stability

during such changes, implementations are required to simultaneously store and support the use of more than one RSVP Authentication Key on a given interface.

Each key will have its own Key Identifier, which is stored locally. The combination of the Key Identifier and the interface associated with the message uniquely identifies the Authentication Algorithm and RSVP Authentication Key in use.

As noted above in Section 2.2.1, the party creating the RSVP message will select a valid key from the set of valid keys for that interface. The receiver will use the Key Identifier and interface to determine which key to use for authentication of the received message. More than one key may be associated with an interface at the same time.

Hence it is possible to have fairly smooth RSVP Authentication Key rollovers without losing legitimate RSVP messages because the stored key is incorrect and without requiring people to change all the keys at once. To ensure a smooth rollover, each communicating RSVP system must be updated with the new key several minutes before they current key will expire and several minutes before the new key lifetime begins. The new key should have a lifetime that starts several minutes before the old key expires. This gives time for each system to learn of the new RSVP Authentication Key before that key will be used. It also ensures that the new key will begin being used and the current key will go out of use before the current key's lifetime expires. For the duration of the overlap in key lifetimes, a system may receive messages using either key and authenticate the message.

Key storage SHOULD persist across a system restart, warm or cold, to avoid operational issues. Key lifetime is an obvious issue, to be solved by the implementation. Obvious solutions include the use of the Network Time Protocol, hardware time of day clocks, or waiting some time before emitting the first message to determine what key other systems are signing with. The matter is left for the implementor.

3.3. Pathological Cases

Two pathological cases exist which must be handled, which are failures of the network manager. Both of these should be exceedingly rare.

During key switch-over, devices may exist which have not yet been successfully configured with the new key. Therefore, systems MAY implement (and would be well advised to implement) an algorithm that detects the set of keys being used by its neighbors, and transmits its

messages using both the new and old keys until all the neighbors are using the new key or the lifetime of the old key expires. Under normal circumstances, this elevated transmission rate will exist for a single refresh interval.

If the last key associated with an interface expires, it is unacceptable to revert to an unauthenticated condition, and not advisable to disrupt current reservations. Therefore, the system should send a "last authentication key expiration" notification to the network manager and treat the key as having an infinite lifetime until the lifetime is extended, the key is deleted by network management, or a new key is configured.

4. Conformance Requirements

To conform to this specification, an implementation MUST support all of its aspects. The MD5 authentication algorithm defined in RFC-1321 MUST be implemented by all conforming implementations. A conforming implementation MAY also support other authentication algorithms such as NIST's Secure Hash Algorithm (SHA). Manual key distribution as described above MUST be supported by all conforming implementations. All implementations MUST support the smooth key rollover described under "Key Change Procedures."

The user documentation provided with the implementation MUST contain clear instructions on how to ensure that smooth key rollover occurs.

Implementations SHOULD support a standard key management protocol for secure distribution of RSVP Authentication Keys once such a key management protocol is standardized by the IETF.

5. Acknowledgments

This work was done by the RSVP Working Group, of which Bob Braden is the Chair. Ran Atkinson co-authored the original papers describing it, written for OSPF and RIP-II. The author gladly acknowledges significant inputs from each of these sources.

6. References

[1] Braden et al, "Resource ReSerVation Protocol (RSVP) -- Version 1

Functional Specification", March 24. 1995 John,

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

[3] S. Bellovin, "Security Problems in the TCP/IP Protocol Suite", ACM Computer Communications Review, Volume 19, Number 2, pp.32-48, April 1989.

[4] N. Haller, R. Atkinson, "Internet Authentication Guidelines", RFC-XXXX (already submitted to RFC Editor), September 1994.

[5] N. Haller, R. Atkinson, "On Internet Authentication", Request for Comments 1704, DDN Network Information Center,

7. Security Considerations

This entire memo describes and specifies an authentication mechanism for RSVP that is believed to be secure against active and passive attacks. Passive attacks are clearly widespread in the Internet at present. Protection against active attacks is also needed even though such attacks are not currently widespread.

Users need to understand that the quality of the security provided by this mechanism depends completely on the strength of the implemented authentication algorithms, the strength of the key being used, and the correct implementation of the security mechanism in all communicating RSVP implementations. This mechanism also depends on the RSVP Authentication Key being kept confidential by all parties. If any of these incorrect or insufficiently secure, then no real security will be provided to the users of this mechanism.

Specifically with respect to the use of SNMP, compromise of SNMP security has the necessary result that the various RSVP configuration parameters (e.g. routing table, RSVP Authentication Key) manageable through SNMP could be compromised as well. Changing Authentication Keys using non-encrypted SNMP is no more secure than sending passwords in the clear.

Confidentiality is not provided by this mechanism. Work is underway within the IETF to specify a standard mechanism for IP-layer encryption. That mechanism might be used to provide confidentiality for RSVP in the future. Protection against traffic analysis is also not provided. Mechanisms such as bulk link encryption might be used when protection against traffic analysis is required.

8. Author's Address

Fred Baker
Cisco Systems
519 Lado Drive
Santa Barbara, California 93111
Phone: (805) 681-0115
Email: fred@cisco.com

Table of Contents