Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 19, 2008

M. Behringer
F. Le Faucheur
Cisco Systems Inc
November 16, 2007

Applicability of Keying Methods for RSVP Security
draft-behringer-tsvwg-rsvp-security-groupkeying-01.txt

Status of this Memo

Copyright Notice

Abstract

The Resource reSerVation Protocol (RSVP) allows hop-by-hop authentication of RSVP neighbors. This requires messages to be cryptographically signed using a shared secret between participating nodes. This document compares group keying for RSVP with per neighbor or per interface keying, and discusses the associated key provisioning methods as well as applicability and limitations of these approaches. Draft-weis-gdoi-for-rsvp specifies how the Group Domain of Interpretation (GDOI) can be used to distribute group keys

to RSVP nodes. The present document also discusses applicability of such group keying to RSVP encryption.

Table of Contents

1. Introduction and Problem Statement

The Resource reSerVation Protocol [RFC2205] allows hop-by-hop authentication of RSVP neighbors, as specified in [RFC2747]. In this mode, an integrity object is attached to each RSVP message to transmit a keyed message digest. This message digest allows the recipient to verify the authenticity of the RSVP node that sent the message, and to validate the integrity of the message. Through the inclusion of a sequence number in the scope of the digest, the digest also offers replay protection.

[RFC2747] does not dictate how the key for the integrity operation is derived. Currently, most implementations of RSVP use a statically configured key, per interface or per neighbor. However, to manually configure key per router pair across an entire network is operationally hard, especially for key changes. Effectively, many users of RSVP therefore resort to the same key throughout their network, and change it rarely if ever, because of the operational burden. [RFC3562] however recommends regular key changes, at least every 90 days.

[I-D.weis-gdoi-for-rsvp] provides an alternative solution, using GDOI ([RFC3547]) for key distribution. This allows dynamic key updates, valid for a complete set of RSVP speakers.

The present document describes the various keying methods and their applicability to different RSVP deployment environments, for both message integrity and encryption. It does not mandate any particular method, but is meant as a comparative guideline to understand where each RSVP keying method is best deployed, and its limitations. Furthermore, it discusses how RSVP hop by hop authentication is impacted in the presence of non-RSVP nodes, or subverted nodes, in the reservation path.

The document "RSVP Security Properties" ([RFC4230]) provides an overview of RSVP security, including RSVP Cryptographic Authentication [RFC2747], but does not discuss key management, nor the extensions that [I-D.weis-gdoi-for-rsvp] suggests. It states that "RFC 2205 assumes that security associations are already available.". The present document focuses specifically on key management with different key types, including GDOI derived keys, as specified in [I-D.weis-gdoi-for-rsvp]. Therefore this document complements [RFC4230].

2. The RSVP Trust Model

Many protocol security mechanisms used in networks require and use

per peer authentication. Each hop authenticates its neighbor with a shared key or certificate. This is also the model used for RSVP. Trust in this model is transitive. Each RSVP node trusts explicitely only its RSVP next hop peers, through the message digest contained in the INTEGRITY object. The next hop RSVP speaker in turn trusts its own peers and so on. See also the document ""RSVP security properties" [RFC4230] for more background.

The keys used for generating the RSVP messages can, in particular, be group keys (for example distributed via GDOI [RFC3547], as discussed in [I-D.weis-gdoi-for-rsvp]). The trust model is the same as for RSVP authentication. This is described in more detail in the section "Using GDOI For RSVP Encryption" in section 5.5.

The trust an RSVP node has to another RSVP node has an explicit and an implicit component. Explicitely the node trusts the other node to maintain the RSVP messages intact or confidential, depending on whether authentication or encryption (or both) is used. This means only that the message has not been altered or seen by another, non-trusted node. Implicitely each node trusts each other node with which it has a trust relationship established via the mechanisms here to adhere to the protocol specifications laid out by the various standards. Note that in any group keying scheme like GDOI a node trusts explicitely as well as implicitely all the other members of the group.

The RSVP protocol can operate in the presence of a non-RSVP router in the path from the sender to the receiver. The non-RSVP hop will ignore the RSVP message and just pass it along. The next RSVP node can then process the RSVP message. For RSVP authentication to work in this case, the key used for computing the RSVP message digest needs to be shared by the two RSVP neighbors, even if they are not IP neighbors. However, in the presence of non-RSVP hops, while an RSVP node always know the next IP hop before forwarding an RSVP Message, it does not always know the RSVP next hop. Thus, the presence of non-RSVP hops impacts operation of RSVP authentication and may influence the keying approaches. This is further discussed in Section 5.3.

3. Key types for RSVP

3.1. Interface based keys

Most current RSVP authentication implementations support interface based RSVP keys. When the interface is point-to-point (and therefore an RSVP router only has a single RSVP neighbor on each interface), this is similar to neighbor based keys in the sense that a different

key is used for each neighbor. However, when the interface is multipoint, all RSVP speakers on a given subnet have to share the same key in this model, which makes it unsuitable for deployment scenarios where different trust groups share a subnet, for example Internet exchange points. In such a case, neighbor based keys are required.

3.2. Neighbor based keys

In this model, an RSVP key is bound to an interface plus a neighbor on that interface. It allows the distinction of different trust groups on a single subnet. (Assuming that layer-2 security is correctly implemented to prevent layer-2 attacks.)

3.3. Group keys

Here, all members of a group of RSVP nodes share the same key. This implies that a node uses the same key regardless of the next RSVP hop that will process the message (within the group of nodes sharing the particular key). It also implies that a node will use the same key on the receiving as on the sending side (when exchanging RSVP messages withn the group).

4. Key Provisioning Methods for RSVP

4.1. Static Key Provisioning

The simplest way to implement RSVP authentication is to use static, preconfigured keys. Static keying can be used with interface based keys, neighbor based keys or group keys.

However, such static key provisioning is expensive on the operational side, since no secure automated mechanism can be used, and initial provisioning as well as key updates require configuration. This method is therefore mostly useful for small deployments, where key changes can be carried out manually, or for deployments with automated configuration tools which support key changes.

Static key provisioning is therefore not an ideal model in a large network.

Often, the number of interconnection points across two domains where RSVP is allowed to transit is relatively small and well controlled. Also, the different domains may not be in a position to use an infrastructure trusted by both domains to update keys on both sides. Thus, manually configured keys may be applicable to inter-domain RSVP authentication.

Since it is not practical to carry out the key change at the exact same time on both sides, some grace period needs to be implemented during which an RSVP node will accept both the old and the new key. Otherwise, RSVP operation would suffer interruptions.

4.2. Per Neighbor Key Negotiation

To avoid the problem of manual key provisioning and updates in static key deployments, key negotiation between RSVP neighbors could be used. Key negotiation could be used to derive either interface or neighbor based keys. However, existing key negotiation protocols such as IKEv1[RFC2409] or IKEv2 [RFC4306] may not be appropriate in all environments because of the relative complexity of the protocols and related operations.

4.3. Dynamic Key Distribution using GDOI

[I-D.weis-gdoi-for-rsvp] describes a mechanism to distribute group keys to a group of RSVP speakers, using GDOI [RFC3547]. In this model, a key server authenticates each of the RSVP nodes independently, and then distributes a group key to the entire group.

5. Applicability of Various Keying Methods for RSVP

5.1. Per Neighbor or Per Interface Keys for Authentication

Per interface and per neighbor keys can be used within a single security domain. As mentioned above, per interface keys are only applicable when all the hosts reachable on the specific interface belong to the same security domain.

These key types can also be used between security domains, since they are specific to a particular interface or neighbor. Again, interface level keys can only be deployed safely when all the reachable neighbors on the interface belong to the same security domain.

As discussed in Section 5.3, per neighbor and per interface keys can not be used in the presence of non-RSVP hops.

5.2. Group Keys for Authentication

Group keys apply naturally to intra-domain RSVP authentication, since all RSVP nodes implicitely trust each other. Using group keys, they extend this trust to the group key server. This is represented in Figure 1.

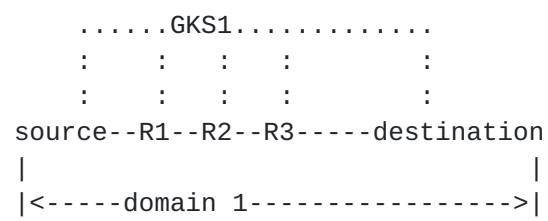


Figure 1: Group Key Server within a single security domain

A single group key cannot normally be used to cover multiple security domains however, because by definition the different domains do not trust each other and would not be willing to trust the same group key server. For a single group key to be used in several security domains, there is a need for a single group key server, which is trusted by both sides. While this is theoretically possible, in practice it is unlikely that there is a single such entity trusted by both domains. Figure 2 illustrates this setup.

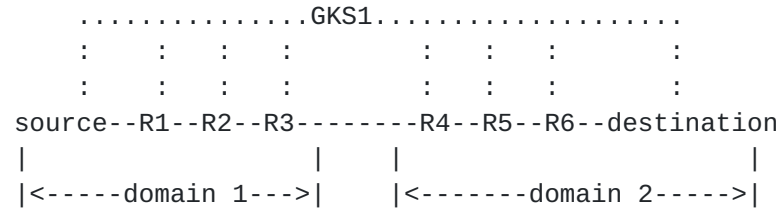


Figure 2: A Single Group Key Server across security domains

A more practical approach for RSVP operation across security domains, is to use a separate group key server for each security domain, and to use per interface or per peer authentication between the two domains. Figure 3 shows this set-up.

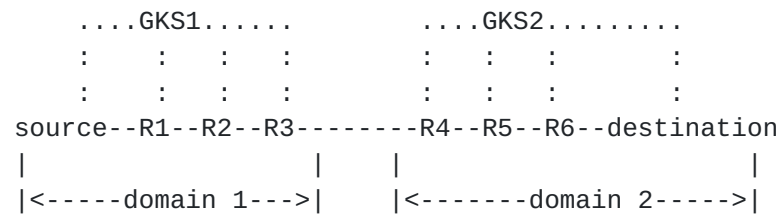
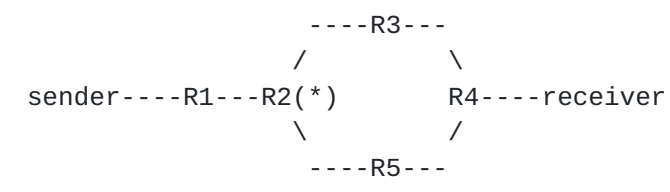


Figure 3: A group Key Server per security domain

5.3. Non-RSVP Hops

In the presence of a non-RSVP router in the path from the sender to the receiver, regular RSVP keeps working. The non-RSVP node ignores the RSVP message, and passes it on transparently to the next node. Figure 4 illustrates this scenario. R2 in this picture does not participate in RSVP, the other nodes do. In this case, R2 will pass

on any RSVP messages unchanged, and will ignore them.



(*) Non-RSVP hop

Figure 4: A non-RSVP Node in the path

However, this creates an additional challenge for RSVP authentication. In the presence of a non-RSVP hop, with some RSVP messages such as a Path message, an RSVP router does not know the RSVP next hop for that message at the time of forwarding it. In fact, part of the role of a Path message is precisely to discover the RSVP next hop (and to dynamically re-discover it when it changes, say because of a routing change). For example, in Figure 4, R1 knows that the next IP hop for a Path message addresed to the receiver is R2, but it does necessarily not know if the RSVP next hop is R3 or R5.

This means that per interface and per neighbor keys cannot easily be used in the presence of non-RSVP routers on the path between senders and receivers.

By contrast, group keying will naturally work in the presence of non-RSVP routers. Referring back to Figure 4, with group keying, R1 would use the group key to sign a Path message addressed to the receiver and forwards it to R2. Being a non-RSVP node, R2 and will ignore and forward the Path message to R3 or R5 depending on the current shortest path as determined by routing. Whether it is R3 or R5, the RSVP router that receives the Path message will be able to authenticate it successfully with the group key.

5.4. Subverted RSVP Nodes

A subverted node is defined here as an untrusted node, for example because an intruder has gained control over it. Since RSVP authentication is hop-by-hop and not end-to-end, a subverted node in the path breaks the chain of trust. This is to a large extent independent of the type of keying used.

For interface or per-neighbor keying, the subverted node can now introduce fake messages to its neighbors. This can be used in a variety of ways, for example by changing the receiver address in the Path message, or by generating fake Path messages. This allows path states to be created on every RSVP router along any arbitrary path

through the RSVP domain. That in itself could result in a form of Denial of Service by allowing exhaustion of some router resources (e.g. memory). The subverted node could also generate fake Resv messages upstream corresponding to valid Path states. In doing so, the subverted node can reserve excessive amounts of bandwidth thereby possibly performing a denial of service attack.

Group keying allows the additional abuse of sending fake RSVP messages to any node in the RSVP domain, not just adjacent RSVP nodes. However, in practice this can be achieved to a large extent also with per neighbor or interface keys, as discussed above. Therefore the impact of subverted nodes on the path is comparable, independently whether per-interface, per-neighbor or group keys are used.

5.5. RSVP Encryption

The keying material can also be used to encrypt the RSVP messages using IPsec [RFC2401], instead of, or in addition to authenticating them. The same considerations apply for this case as discussed above for the authentication case. Group keys are applicable only within a trusted domain, but they allow operation through non-RSVP speakers without further configuration. Per interface or per neighbor keys work also inter-domain, but do not operate in the presence of a non-RSVP router.

The existing GDOI standard as described in [RFC3547] contains all relevant policy options to allow for RSVP encryption, and no extensions are necessary. An example GDOI policy would be to encrypt all packets of the RSVP protocol itself (IP protocol 46). A router implementing GDOI is therefore automatically able to encrypt RSVP.

[Editor's note: Applicability of tunnel vs transport mode still need to be discussed.]

5.6. RSVP Notify Messages

[RFC3473] introduces the Notify message and allows such Notify messages to be sent in a non-hop-by-hop fashion. As discussed in the Security Considerations section of [RFC3473], this can interfere with RSVP's hop-by-hop integrity and authentication model. [RFC3473] describes how standard IPsec based integrity and authentication can be used to protect Notify messages. We observe that, alternatively, in some environments, group keying may allow use of regular RSVP authentication ([RFC2747]) for protection of non-hop-by-hop Notify messages. For example, this may be applicable to controlled environments where nodes invoking notification requests are known to belong to the same key group as nodes generating Notify messages.

6. End Host Considerations

Unless RSVP Proxy entities ([I-D.ietf-tsvwg-rsvp-proxy-approaches] are used, RSVP signaling is controlled by end systems and not routers. As discussed in [RFC4230], RSVP allows both user-based security and host-based security. User-based authentication aims at "providing policy based admission control mechanism based on user identities or application." 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 user's host and transmitted inside the RSVP message. This way, a user may authenticate to the Policy Decision Point (or directly to the first hop router). Host-based security relies on the same mechanisms as between routers (i.e. INTEGRITY object) as specified in [RFC2747]. For host-based security, interface or neighbor based keys may be used, however, key management with pre-shared keys can be difficult in a large scale deployment, as described in section 4. In principle an end host can also be part of a group key scheme, such as GDOI. If the end systems are part of the same zone of trust as the network itself, group keying can be extended to include the end systems. If the end systems and the network are in different zones of trust, group keying cannot be used.

7. Applicability to Other Architectures and Protocols

While, so far, this document only discusses RSVP security assuming the traditional RSVP model as defined by [RFC2205] and [RFC2747], the analysis is also applicable to other RSVP deployment models as well as to similar protocols:

- Aggregation of RSVP for IPv4 and IPv6 Reservations [RFC3175]: This scheme defines aggregation of individual RSVP reservations, and discusses use of RSVP authentication for the signaling messages. Group keying is applicable to this scheme, particularly when automatic Deaggregator discovery is used, since in that case, the Aggregator does not know ahead of time which Deaggregator will intercept the initial end-to-end RSVP Path message.
- Generic Aggregate Resource ReSerVation Protocol (RSVP) Reservations [RFC4860]: This document also discusses aggregation of individual RSVP reservations. Here again, group keying applies and is mentioned in the Security Considerations section.
- Aggregation of Resource ReSerVation Protocol (RSVP) Reservations over MPLS TE/DS-TE Tunnels [RFC4804]([RFC4804]): This scheme also defines a form of aggregation of RSVP reservation but this time over MPLS TE Tunnels. Similarly, group keying may be used in such an environment.

- Pre-Congestion Notification (PCN): [I-D.ietf-pcn-architecture] defines an architecture for flow admission and termination based on aggregated pre-congestion information. One deployment model for this architecture is based on IntServ over DiffServ: the DiffServ region is PCN-enabled, RSVP signalling is used end-to-end but the PCN-domain is a single RSVP hop, i.e. only the PCN-boundary-nodes process RSVP messages. In this scenario, RSVP authentication may be required among PCN-boundary-nodes and the considerations about keying approaches discussed earlier in this document apply. In particular, group keying may facilitate operations since the ingress PCN-boundary-node does not necessarily know ahead of time which Egress PCN-boundary-node will intercept and process the initial end-to-end Path message. Note that from the viewpoint of securing end-to-end RSVP, there are a lot of similarities in scenarios involving RSVP Aggregation over aggregate RSVP reservations ([RFC3175], [RFC4860]), RSVP Aggregation over MPLS-TE tunnels ([RFC4804]), and RSVP (Aggregation) over PCN ingress-egress aggregates.

8. Security Considerations

This entire document discusses RSVP security.

9. Acknowledgements

The authors would like to thank everybody who provided feedback on this document. Specific thanks to Bob Briscoe, Hannes Tschofenig and Brian Weis.

10. Changes to Previous Version

The following changes were made in version 01:

- New section "Applicability to Other Architectures and Protocols": Goal is to clarify the scope of this document: The idea presented here is also applicable to other architectures (PCN[I-D.ietf-pcn-architecture], RFC3175 and RFC4860, etc.
- Clarified the scope of this document versus RFC4230 (in the introduction, last paragraph).
- Added a section on "End Host Considerations".
- Expanded section 5.5 (RSVP Encryption) to clarify that GDOI contains all necessary mechanisms to do RSVP encrpytion.
- Tried to clarify the "trust to do what?" question raised by Bob Briscoe in a mail on 26 Jul 2007. See the section on trust model.

- Lots of small editorial changes (references, typos, figures, etc).
- Added an Acknowledgements section.

11. Informative References

[I-D.ietf-pcn-architecture]
Eardley, P., "Pre-Congestion Notification Architecture", October 2007.

[I-D.ietf-tsvwg-rsvp-proxy-approaches]
Faucheur, F., "RSVP Proxy Approaches", draft-ietf-tsvwg-rsvp-proxy-approaches-02 (work in progress), September 2007.

[I-D.weis-gdoi-for-rsvp]
Weis, B., "Group Domain of Interpretation (GDOI) support for RSVP", July 2007.

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

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

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

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

[RFC3175] Baker, F., Iturralde, C., Le Faucheur, F., and B. Davie, "Aggregation of RSVP for IPv4 and IPv6 Reservations", RFC 3175, September 2001.

[RFC3473] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.

[RFC3547] Baugher, M., Weis, B., Hardjono, T., and H. Harney, "The Group Domain of Interpretation", RFC 3547, July 2003.

[RFC3562] Leech, M., "Key Management Considerations for the TCP MD5 Signature Option", RFC 3562, July 2003.

[RFC4230] Tschofenig, H. and R. Graveman, "RSVP Security Properties", RFC 4230, December 2005.

[RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[RFC4804] Le Faucheur, F., "Aggregation of Resource ReSerVation Protocol (RSVP) Reservations over MPLS TE/DS-TE Tunnels", RFC 4804, February 2007.

[RFC4860] Le Faucheur, F., Davie, B., Bose, P., Christou, C., and M. Davenport, "Generic Aggregate Resource ReSerVation Protocol (RSVP) Reservations", RFC 4860, May 2007.

Authors' Addresses

Michael H. Behringer
Cisco Systems Inc
Village d'Entreprises Green Side
400, Avenue Roumanille, Batiment T 3
Biot - Sophia Antipolis 06410
France

Email: mbehring@cisco.com
URI: http://www.cisco.com

Francois Le Faucheur
Cisco Systems Inc
Village d'Entreprises Green Side
400, Avenue Roumanille, Batiment T 3
Biot - Sophia Antipolis 06410
France

Email: flefauch@cisco.com
URI: http://www.cisco.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).