TEAS Working Group
Internet-Draft
Obsoletes: 2747 3097 (if approved)
Intended status: Standards Track
Expires: 30 October 2026

R. Atkinson
J. Halpern
Consultant

28 April 2026

RSVP Cryptographic Authentication with HMAC-SHA2
draft-ietf-teas-rsvp-hmac-sha2-01

Abstract

This document specifies the use of the US NIST Secure Hash Standard in the Hashed Message Authentication Code (HMAC) mode with RSVP Cryptographic Authentication version 2. Along with draft-ietf-teas-rsvp-auth-v2, this document obsoletes RFC2747 and RFC3097.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents 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 "work in progress."

This Internet-Draft will expire on 30 October 2026.

Table of Contents

1. Introduction

This document specifies the use of the US NIST Secure Hash Standard in the Hashed Message Authentication Code (HMAC) mode with RSVP Cryptographic Authentication version 2.

Those secure hash algorithms are defined in the US NIST Secure Hash Standard (SHS) [NIST-SHS]. [NIST-HMAC] specifies multiple cryptographic hash functions, including SHA-256, SHA-384, and SHA-512. The Hashed Message Authentication Code (HMAC) authentication mode is defined in [NIST-HMAC] and [RFC2104].

While it is believed that [RFC2104] is mathematically identical to [NIST-HMAC] and it is also believed that algorithms in [RFC6234] are mathematically identical to [NIST-SHS], in the event of any confusion or discrepancies the NIST specifications are correct and are normative and canonical.

This addition to RSVP Cryptographic Authentication was driven by operator requests that they be able to use algorithms from the NIST Secure Hash Standard in the NIST HMAC mode for RSVP Cryptographic Authentication, instead of being forced to use the much older Keyed-MD5 algorithm and mode as originally defined for RSVP Cryptographic Authentication.[RFC2747][RFC3097] As of the date of publication, the Keyed MD5 construction is widely believed not to have sufficient cryptographic strength.[RFC6151]

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Background

All RSVP protocol exchanges can be authenticated using the revised mechanism defined in draft-ietf-teas-rsvp-auth-v2. That specification is carefully written to be independent both of the cryptographic algorithm and the cryptographic mode. This approach means that additional cryptographic modes and cryptographic algorithms can be defined in the future without needing to change the RSVP Authentication specification of draft-ietf-teas-rsvp-auth-v2.

This document specifies the use of NIST SHS algorithms with the HMAC mode for use with draft-ietf-teas-rsvp-auth-v2. In the future, other documents might be specified for other cryptographic algorithms with this or another cryptographic mode.

The combination of a cryptographic mode (e.g., HMAC) with a specific cryptographic algorithm (e.g., SHA256) is known as a "Cryptographic Transform" (e.g., HMAC-SHA256).

3. Cryptographic Authentication with NIST SHS in HMAC Mode

When using this authentication method, a shared secret Cryptographic Key, the cryptographic mode and algorithm (e.g., HMAC-SHA256), and its associated Key Identifier are configured in the router. For each RSVP protocol packet, that secret key is used to generate/verify a "message digest" that is placed in the Authentication Data field of the RSVP INTEGRITY object. The message digest is a one-way function of the RSVP protocol packet and the secret key. Since the secret key is never sent over the network in the clear, protection is provided against passive attacks [RFC1704].

This specification discusses the computation of the Authentication Data field of the RSVP INTEGRITY object when one of the NIST SHS family of algorithms is used in the Hashed Message Authentication Code (HMAC) mode.

With the additions in this document, the currently valid Cryptographic Transforms for use with RSVP Cryptographic Authentication include:

TRANSFORM NAME	SPECIFICATION(S)
Keyed-MD5	{{RFC2747}} and {{RFC3097}}
HMAC-SHA-256	(defined here)
HMAC-SHA-384	(defined here)
HMAC-SHA-512	(defined here)

Of the above, all implementations of this specification MUST support at least both:

HMAC-SHA-256
HMAC-SHA-512

and SHOULD (for backwards compatibility with existing implementations and deployments of RSVP Cryptographic Authentication) include support for:

Keyed-MD5

and MAY include support for:

HMAC-SHA-384

> NOTE WELL: This document deliberately does not specify a Cryptographic Transform using either SHA-1 or SHA-224 because those algorithms are considered to be too weak as of this document’s publication date.

An implementation of this specification MUST allow network operators to configure ANY RSVP Cryptographic Transform supported by that implementation for use with ANY given RSVP Security Association (and with its corresponding Key Identifier value) that is configured into that router.

3.1. Generating Cryptographic Authentication

First, following the procedure defined in draft-ietf-teas-rsvp-auth-v2, select the appropriate RSVP Security Association for use with this packet and set the Key Identifier field to the Key Identifier value of that RSVP Security Association.

Second, add an RSVP INTEGRITY object to the outgoing RSVP packet if one does not yet exist, taking care to size the Authentication Data field appropriately for the Cryptographic Algorithm specified in that RSVP Security Association. Using the appropriate RSVP Security Association, set the Flags field, set the AAL field to the appropriate value for the Cryptographic Algorithm that will be used, set the Key Identifier, and set the Sequence Number. The Authentication Data field is filled with the fixed value of "Apad", which is defined in the next section.

When any NIST SHS algorithm is used in HMAC mode with RSVP Cryptographic Authentication, the Authentication Data Length is equal to the normal hash output length (measured in bytes) for the specific NIST SHS algorithm in use.

Cryptographic Transform	AAL Field value
HMAC-SHA256	4
HMAC-SHA384	8
HMAC-SHA512	12

Table 1

Third, the Sequence Number of the RSVP INTEGRITY object is set following the procedures in draft-ietf-teas-rsvp-auth-v2.

Fourth, the authentication data is calculated, as described below.

3.2. Cryptographic Aspects

This describes the computation of the Authentication Data value when the HMAC cryptographic mode is combined with any NIST SHS algorithm is used with RSVP Cryptographic Authentication.

The value of Apad selected is arbitrary. The only goal was to pick a different value of Apad than is in use with other IETF routing or control protocols.

In the algorithm description below, the following nomenclature, which is consistent with [NIST-HMAC], is used:

H is the specific hashing algorithm (e.g., SHA-256).

K is the Authentication Key for the RSVP Security Association.

Ko is the cryptographic key used with the hash algorithm.

B is the block size of H, measured in octets rather than bits. Note well that B is the internal block size, not the hash size.

For SHA-256: B == 64
For SHA-384 and SHA-512: B == 128

L is the length of the hash, measured in octets rather than bits.

XOR is the exclusive-or operation.

Opad is the hexadecimal value 0x5c repeated B times.

Ipad is the hexadecimal value 0x36 repeated B times.

Apad is the hexadecimal value 0x7865FE3E repeated (L/4) times.

Implementation Notes:
This definition of Apad means that Apad is always the same length as the hash output.

The Authentication Data field length for SHA256 is 32 bytes, for SHA384 is 48 bytes, and for SHA512 is 64 bytes. As a side effect, RSVP packets containing the RSVP INTEGRITY object will be larger when hash functions with larger hash output sizes are used.

(1) PREPARATION OF KEY In this application, Ko is always L octets long.

If the Authentication Key (K) is L octets long, then Ko is equal to K. If the Authentication Key (K) is more than L octets long, then Ko is set to H(K). If the Authentication Key (K) is less than L octets long, then Ko is set to the Authentication Key (K) with zeros appended to the end of the Authentication Key (K), such that Ko is L octets long.

(2) FIRST-HASH First, the RSVP INTEGRITY object’s Authentication Data field is filled with the value Apad.

Then, a First-Hash, also known as the inner hash, is computed as follows:

First-Hash = H(Ko XOR Ipad || (RSVP Packet))

The definition of Apad (above) ensures the inner hash is always the same length as the hash output.

(3) SECOND-HASH Then a Second-Hash, also known as the outer hash, is computed as follows:

Second-Hash = H(Ko XOR Opad || First-Hash)

(4) RESULT The resulting Second-Hash becomes the Authentication Data that is sent in the Authentication Data field of the RSVP Integrity Object.

3.3. Message Verification

Message verification follows the procedure defined in draft-ietf-teas-rsvp-auth-v2, except that the cryptographic calculation of the message digest follows the procedure in Cryptographic Considerations above when any NIST SHS algorithm in the HMAC mode is in use. The KeyID of the received RSVP packet is used to help locate the correct RSVP Security Association to use.

Implementation Note: One must save the received digest value before calculating the expected digest value, so that after that calculation the received value can be compared with the expected value to determine whether to accept that RSVP packet.

3.4. Smooth Key Rollover

The Key Identifier field of the RSVP INTEGRITY object enables smooth key rollover in operational deployments. Based on the lifetime of the current RSVP Security Association, both sender and receiver will know when to switch to a different RSVP Security Association and will do so at the same time.

Both draft-ietf-teas-rsvp-auth-v2 and this document require that all implementations MUST support more than one RSVP Security Association at any given time, because this also is needed to enable smooth key rollover. Implementations SHOULD support many more than two concurrent RSVP Security Associations as that can greatly simplify operational security management. Additional discussion of this is in draft-ietf-teas-rsvp-auth-v2.

After changing the active RSVP Security Association, the RSVP sender will use the (different) Key Identifier value associated with the newly active RSVP Security Association. The receiver will use this new Key Identifier to select the appropriate (new) RSVP Security Association to use with the received RSVP packet whose INTEGRITY object contains the new Key Identifier value.

Additional discussion of smooth key rollover can be found in draft-ietf-teas-rsvp-auth-v2.

Because the Key Identifier field is present, the receiver does not need to (and implementations of this specification MUST NOT) try all configured RSVP Security Associations with any received RSVP packet. This requirement mitigates some of the risks of a Denial-of-Service (DoS) attack on the RSVP instance, but does not entirely prevent all conceivable DoS attacks. For example, an on-link adversary still could generate RSVP packets that are syntactically valid but that contain invalid Authentication Data, thereby forcing the receiver(s) to perform expensive cryptographic computations to discover that the packets are invalid.

4. Security Considerations

This document enhances the security of the RSVP signaling protocol by specifying support for the algorithms defined in the NIST Secure Hash Standard (SHS) using the Hashed Message Authentication Code (HMAC) mode.

The value Apad is used here primarily for consistency with IETF specifications for HMAC-SHA authentication of RIPv2 SHA [RFC4822] and IS-IS SHA [RFC5310]. The value of Apad chosen is arbitrary, other than being different from the value used with RIPv2, OSPFv2, or IS-IS authentication.

The quality of the security provided by the Cryptographic Authentication option depends completely on the strength of the cryptographic algorithm and cryptographic mode in use, the strength of the key being used, and the correct implementation of the authentication mechanism in all communicating RSVP implementations. Accordingly, the use of high assurance development methods is recommended. Security of a deployment of RSVP Authentication also requires that all parties maintain the secrecy of the shared secret key. [RFC4086] and [NIST-ENTROPY] provide guidance on methods for generating cryptographically random bits.

Because the RSVP protocol contains information that need not be kept confidential, privacy is not a requirement. This mechanism significantly increases the work an adversary would need to undertake to inject false information into the RSVP protocol deployment, while remaining practical to deploy.

If the RSVP Security Association is not rekeyed frequently enough, then this mechanism is vulnerable to a replay attack by any on-link node. An on-link node could record a legitimate RSVP packet sent on the link, then replay that packet at the next time the recorded RSVP packet's sequence number is valid.

To prevent this replay attack, operators ought to rekey the RSVP session prior to the sequence number repeating. For additional discussion, please read draft-ietf-teas-rsvp-auth-v2.

The replay attack also might be prevented by using link-encryption.[IEEE-802.1AE-2018]

Operators also should note that an upper-layer authentication mechanism, such as this specification, cannot prevent an attack on the lower layers. Operators should consider deployment of link-layer encryption, such as [IEEE-802.1AE-2018], to protect not only the link-layer but also as additional protection for the upper-layers.

5. IANA Considerations

IANA is requested to add the following entries to the RSVP Cryptographic Transforms registry created in draft-ietf-teas-rsvp-auth-v2:

TRANSFORM	SPECIFICATION	IMPLEMENTATION STATUS
HMAC-256	(this document)	MUST implement
HMAC-384	(this document)	MAY implement
HMAC-512	(this document)	MUST implement

6. Acknowledgements

The authors would like to thank TBD for review of this document.

TBD (in alphabetical order by last name) provided feedback on earlier versions of this document. That feedback has greatly improved both the technical content and the readability of the current document.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[NIST-SHS] (US) National Institute of Standards and Technology (NIST), "Secure Hash Standard (SHS)", August 2015, <https://csrc.nist.gov/pubs/fips/180-4/upd1/final>. Federal Information Processing Standard 180-4

[NIST-HMAC] (US) National Institute of Standards and Technology (NIST), "The Keyed-Hash Message Authentication Code (HMAC)", July 2008, <https://csrc.nist.gov/pubs/fips/198-1/final>. Federal Information Processing Standard 198-1

7.2. Informative References

[RFC1704] Haller, N. and R. Atkinson, "On Internet Authentication", RFC 1704, DOI 10.17487/RFC1704, October 1994, <https://www.rfc-editor.org/info/rfc1704>.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <https://www.rfc-editor.org/info/rfc2104>.

[RFC2747] Baker, F., Lindell, B., and M. Talwar, "RSVP Cryptographic Authentication", RFC 2747, DOI 10.17487/RFC2747, January 2000, <https://www.rfc-editor.org/info/rfc2747>.

[RFC3097] Braden, R. and L. Zhang, "RSVP Cryptographic Authentication -- Updated Message Type Value", RFC 3097, DOI 10.17487/RFC3097, April 2001, <https://www.rfc-editor.org/info/rfc3097>.

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <https://www.rfc-editor.org/info/rfc4086>.

[RFC4822] Atkinson, R. and M. Fanto, "RIPv2 Cryptographic Authentication", RFC 4822, DOI 10.17487/RFC4822, February 2007, <https://www.rfc-editor.org/info/rfc4822>.

[RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, DOI 10.17487/RFC5310, February 2009, <https://www.rfc-editor.org/info/rfc5310>.

[RFC6151] Turner, S. and L. Chen, "Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms", RFC 6151, DOI 10.17487/RFC6151, March 2011, <https://www.rfc-editor.org/info/rfc6151>.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

[IEEE-802.1AE-2018] Institute of Electrical and Electronics Engineers (IEEE), "IEEE Standard for Local and Metropolitan Area Networks - Media Access Control (MAC) Security", December 2018, <https://standards.ieee.org/IEEE/802.1AE/7154/>. IEEE Standard 802.1AE

[NIST-ENTROPY] Turan, M., "Recommendation for the Entropy Sources Used for Random Bit Generation", January 2018, <https://csrc.nist.gov/pubs/sp/800/90/b/final>. Special Publication 800-90B

Authors' Addresses

Ran Atkinson
Consultant
United States of America
Email: rja.lists@gmail.com

Joel Halpern
Consultant
United States of America
Email: jmh@joelhalpern.org