RSVP Authentication V2

draft-atkinson-teas-rsvp-auth-v2

Ran Atkinson, Tony Li

Problem Statement

- RSVP authentication V1 only supports HMAC-MD5
  - MD5 has been considered weak for 20+ years now
  - Recent attacks on MD5 have increased operational security concerns
  - Time for something stronger
- Generalize to support other algorithms and transforms
- Maintain backwards compatibility
- Other steps to reduce operational risks

Design Constraints

- Maintain backwards compatibility with existing implementations
- Maintain backwards compatibility with existing deployments
- Maintain existing multi-vendor interoperability

Implications of Design Constraints

- Avoid changes to on-the-wire RSVP Integrity object which might break legacy parsers
- Avoid changes to packet processing which break legacy implementations
- Avoid changes which break existing operational deployments

Changes (1)

- Base spec is now carefully modularized:
  - Independent of Crypto algorithm, Crypto mode, & Crypto key size.
  - Will use separate “Transform” docs for specific algorithms, modes, & key sizes, which is conceptually similar to how IPsec has done it for decades now.
  - Will not need to keep revising base spec for each new crypto transform.
- Authentication Data field in INTEGRITY object now may be longer than 16 octets.
  - Existing “Reserved” field has been repurposed to indicate additional length
- Discussion of Security Associations is now RSVP specific

Changes (2)

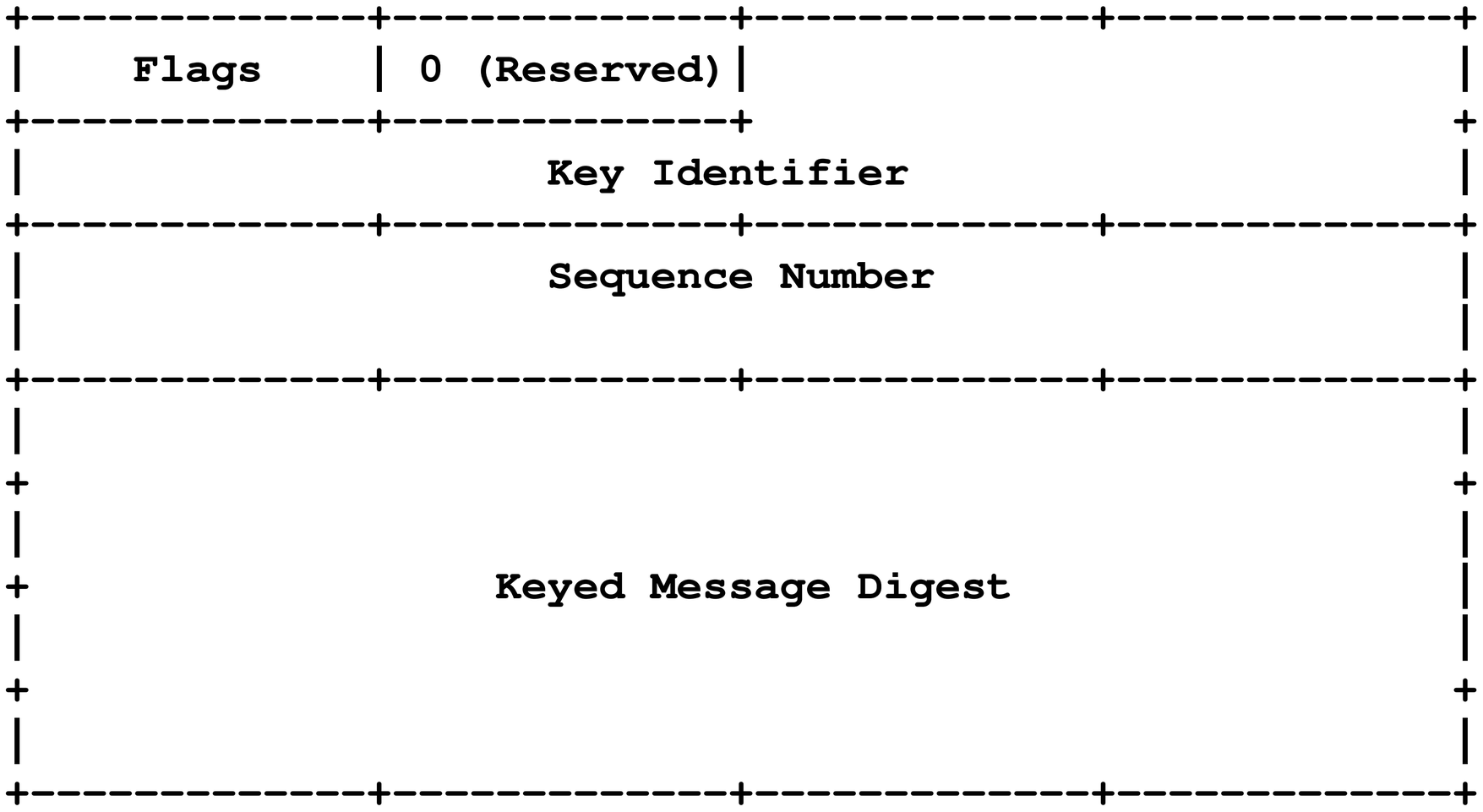
- Integrity handshake now MUST be implemented, which had been optional.
  - Specification is careful to maintain interoperability with legacy implementations which might not implement the Integrity handshake
- Key Management discussion has been updated
  - Specific mention of NetConf as a possible way to provision RSVP Security Associations, which documents existing widespread practice
  - Kerberos discussion was reduced as their appears to be little interest in extending Kerberos for RSVP. Nothing in the new document precludes someone from adding that capability separately, if it is desired

Requests

- Please review and comment
- Need separate documents covering various transforms
  - HMAC with SHA-256 & 256-bit key
  - HMAC with SHA-512 & 512-bit key
  - KMAC with SHA-3…
- Request WG adoption

Backup Slides

RFC 2747: Integrity Object



Key ID

- Q: Why is the Key ID field 48 bits rather than 64 bits ?
- A: Backwards compatibility and interoperability. The Key ID is part of the RSVP Integrity object. The prior specification chose 48 bits, which certainly is sufficiently large.

Keychain

- Q: Why does the specification not require a Keychain ?
- A: Nothing precludes implementations using a Keychain. Presence or absence of a Keychain is an implementation option which does not impact on-the-wire interoperability.

Hop-by-Hop vs. End-to-End Protection

- Q: Why does specification not require end-to-end use of the same RSVP Security Association ?
- A: Backwards compatibility.
  - The prior mechanism did not require operators to deploy this mechanism end-to-end.
  - Existing deployments of RSVP Authentication most commonly use hop-by-hop protections.
  - In fact, the authors are not aware of any end-to-end deployments of RSVP Authentication.

Adding cryptography increases attack risks

- This can be true, for example if one has a naive implementation.
  - Specification provides implementers with guidance on risks and on how to reduce risks through careful implementation.
- (D)DOS attacks are a risk with any useful protocol.
- RSVP deployments without cryptographic authentication have their own risks.
- Ultimately, operators will make choices about their RSVP-enabled network deployment.
- We are simply trying to provide more cryptographic agility, while maintaining multi-vendor interoperability and backwards compatibility.

Why not explicitly signal Key Rollover coming soon

- Backwards compatibility means strong desire to avoid changing the format of the RSVP Integrity object.
  - Changing the RSVP Integrity object in a way that breaks interoperability with existing implementations and existing deployments is undesirable.
- Separately, the Key ID is included in every RSVP Integrity object, so the receiver always knows which RSVP Security Association to use for the received RSVP packet.
  - Our approach fully supports out-of-order delivery of RSVP packets.