Diameter End-to-End Security: Keyed Message Digests, Digital Signatures, and Encryption

draft-korhonen-dime-e2e-security-02

Jouni Korhonen

IEFT #95

Overview

- Overview
- Background
- Changes from -01 to -02
- Strawman solutions proposal

Background

- Charter:
  - Dec 2012 - Submit 'problem statement and requirements for Diameter end-to-end security framework' to the IESG for consideration as an Informational RFC -> **done'ish.**
  - Maintaining and/or progressing, along the standards track, the Diameter Base protocol and Diameter Applications. This includes extensions to Diameter Base protocol that can be considered as enhanced features or bug fixes -> **end to end security falls in this category.**

- Resurrecting old work in this area:
  - draft-korhonen-dime-e2e-security-02

Changes from -01 to -02

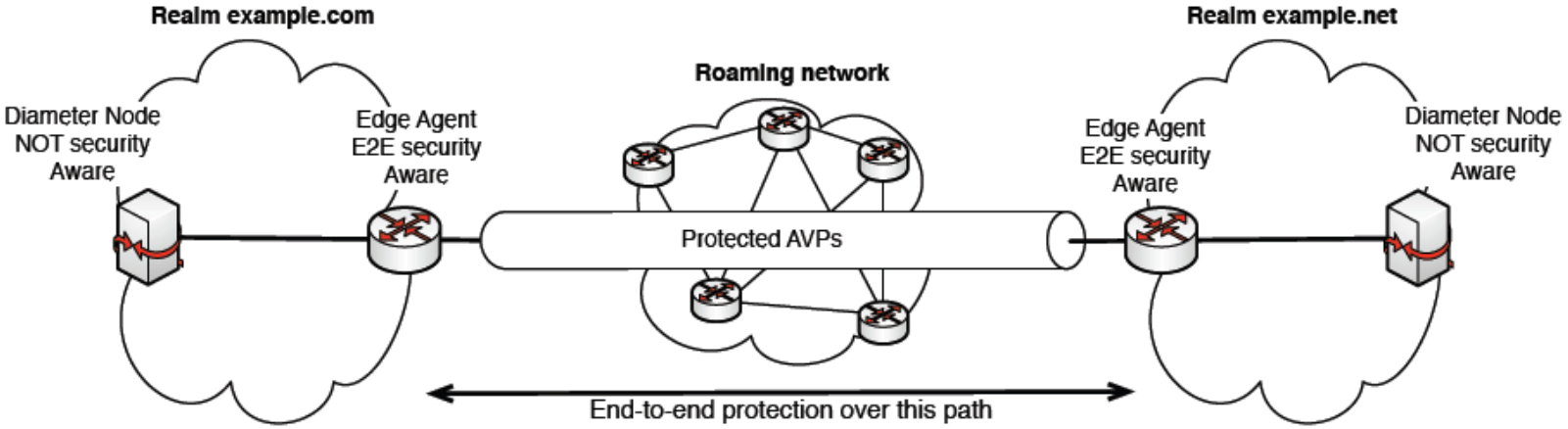
- Changes since IETF 85.. erm none really :)

Strawman solutions proposal

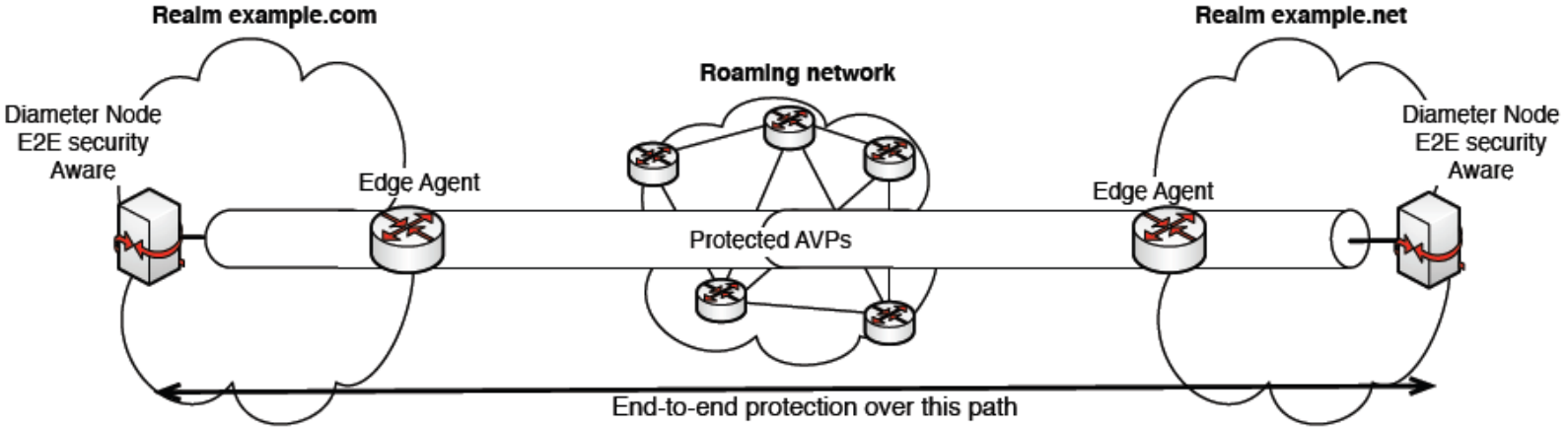
- In scope:
  - AVP integrity and confidentiality protection.
- Out of scope:
  - Authentication & authorization of end points.
  - Key management.

Two main deployment cases

Site-to-site



End-to-end



Protecting AVPs

- Two new AVPs are defined for protecting other AVPs:
  - Signed-Data (octet string) for integrity protection of one or more AVPs.
  - Encrypted-Data (octet string) for confidentiality protection of one or more AVPs.

- Original proposal selected JSON-based approach:
  - JSON Web signature (JWS) for integrity protection.
  - JSON Web Encryption (JWE) for confidentiality protection.

- **New thinking:** what about CBOR/COSE instead of Diameterified use of JSON??

Signed-Data AVP

- The AVP carries JSON Web Signature (JWS) of one or more of AVPs. Each protected AVP is hashed and the hash is included into the JWS payload.

- Hashed AVPs are linked to “originals” using their AVP Code. If there are multiple instances of the same AVP, you hash them all and do one by one verification -> allows for rearranging AVPs and detection of addition/removal/modification of AVPs.

- Both JWS Payload and signature use the same hash algorithm of the cryptographic algorithm indicated in the JWS Header.

- Can be included into **existing Diameter applications**.

Encrypted-Data AVP

- The AVP carries JSON Web Encryption (JWE) data structure and the JWE Payload embeds of one or more protected AVPs.

- Cannot be used with existing Diameter applications since encrypted AVPs are embedded inside the Encrypted-Data AVP(s).

Error Handling

- Transient failures:
  - DIAMETER_KEY_UNKNOWN – A Signed-Data or an Encrypted-Data AVP is received that was generated using a key that cannot be found in the key store. To recover a new end-to-end key establishment procedure may need to be invoked.
  - DIAMETER_HEADER_NAME_ERROR (TBD12 – This error code is returned when a Header Parameter Name is not understood in the JWSHeader AVP or in the JWE-Header AVP.

- Permanent failures:
  - DIAMETER_DECRYPTION_ERROR – This error code is returned when an Encrypted-Data AVP is received and the decryption fails for an unknown reason.
  - DIAMETER_SIGNATURE_ERROR – This error code is returned when a Signed-Data AVP is received and the verification fails for an unknown reason.

Anyway..

- For now this is just a resurrection of an old draft.
- What folks like the overall ‘framework’? Could it serve as a starting point for end to end security solution for Diameter (after some ‘minor’ tweaking)?
- I would welcome discussion and improvement proposal on this draft.