Firmware Encryption with SUIT Manifests

draft-ietf-suit-firmware-encryption

Hannes Tschofenig, Russ Housley, Brendan Moran

Status

- After the virtual interim meeting in May, the draft became a WG item in July.
  - -00 is a resubmission of the individual draft (delta editorial issues)
  - -01 includes an architectural description and further editorial fixes.
    - Implementation of AES-KW for COSE_Encrypt is available.
    - HPKE implementation is also available but has not been integrated into COSE_Encrypt yet.
- Missing:
  - Detailed examples.
  - A description about manifest encryption (currently the focus is on firmware encryption).
  - Should we describe challenges of maintaining confidentiality on the device itself?

Document Management Question

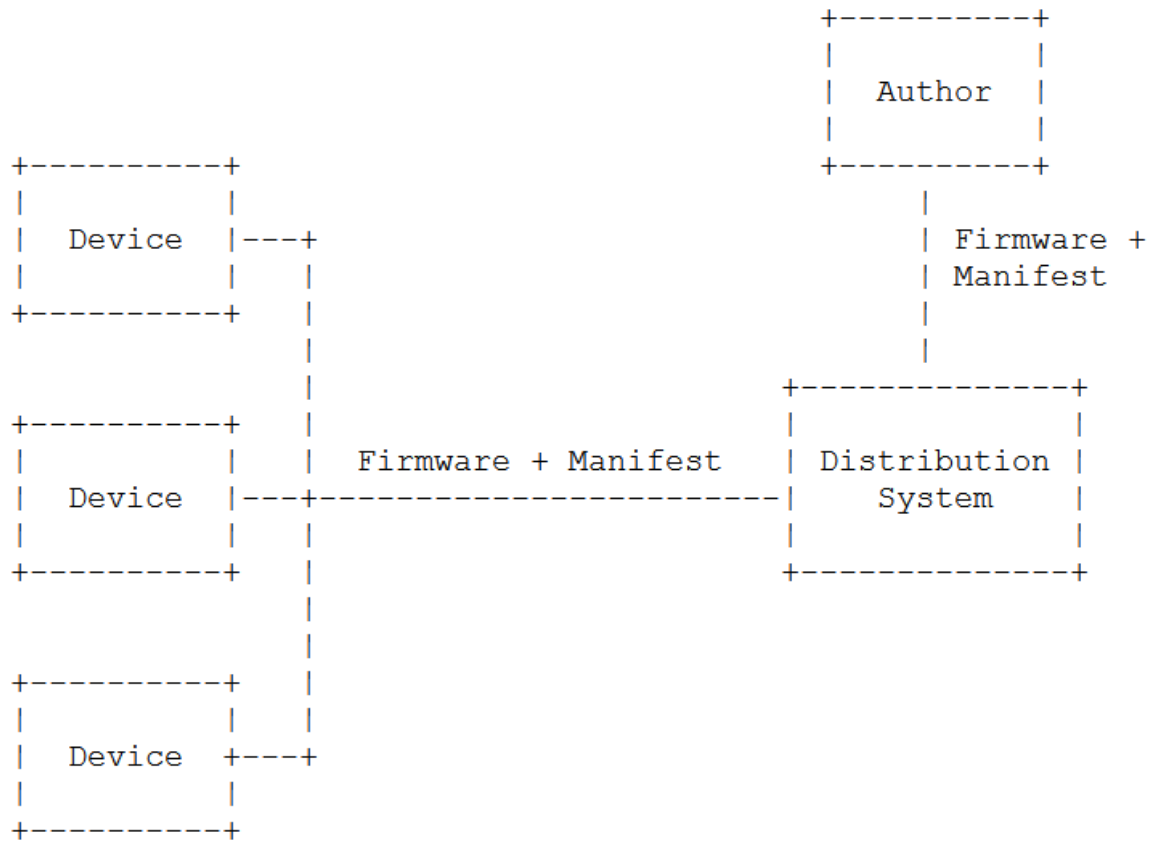
- Should the HPKE functionality become generically usable for COSE?

Architectural Considerations

1. A firmware author may not know all recipients (devices) at time of manifest creation.
2. A distribution system may wish to reduce / increase the numer of recipients a firmware is sent to.

- Conclusion: Recipient list MUST be mutable when the manifest is created.

  => Recipient list cannot be contained in manifest



Mutable Recipient List Threats

Device Suppression in Distribution

- Class: Elevation of Privilege, Repudiation
- Description: Attacker prunes a target device from the recipient list
- Attacker capabilities:
  - On-Path Attacker modifies recipient list in transit OR
  - Attacker compromises CDN that is hosting recipient list
- Results: impossible to determine at point of distribution whether device is unauthorized or under attack

Device energy/flash exhaustion

- Class: Denial of Service
- Description: Attacker alters—rather than deleting cryptographic material for a recipient.
- Attacker capabilities:
  - On-Path Attacker modifies recipient list in transit, OR
  - Attacker compromises CDN that is hosting recipient list
- Results: Recipient does not know that the CEK is incorrect, so it decrypts payload with incorrect key, expending energy and flash cycles until full payload is stored, then fails.

Mutable Recipient List Mitigation: Device Suppression

Idea: Each entity that modifies the recipient list MUST sign the new list.

The old list and signature are discarded.
Each distributor that receives a recipient list MUST verify the signature of the recipient list against the trust anchors of entities authorized to make those modifications.
The signature MAY be delivered to the Recipient, but this is not required.

Encoding proposal: Element added to SUIT_Envelope containing:

{ tstr => COSE_Sign } ; COSE_Sign in detached mode, tstr equal to envelope key of referenced COSE_Encrypt.

Result: Modifications to recipient list can be detected by intermediaries that are not SUIT processors.

HPKE

- Based on current design, the ephemeral ECDHE public key is not authenticated when COSE_Encrypt is in the envelope.
  - Remember: The authentication by the firmware author is the digital signature applied to the manifest.
- Challenge: Envelope is not protected by itself. Hence, a distribution system or any other entity along the path to the device could replaces ephemeral key and CEK ciphertext.

Solution #1:

- Author creates ephemeral public key and places it into the manifest
- Disadvantages:
  - Same ephemeral key used for all recipients.
  - Inflates recipient list by 8 bytes/recipient

Solution #2:

- Author trusts the distribution system.
- Channel security used between the device and the distribution system.

Mutable Recipient List Mitigation: Device energy/flash exhaustion

- Idea: The manifest contains a digest of the CEK. Decrypted CEK verified against digest before decrypting the firmware image.
- Advantages:
  - Any number of ephemeral keys can be used for one CEK
  - AEAD not required; smaller per-recipient data
- Disadvantage:
  - Increases Manifest size by 1 SUIT_Digest
  - Security analysis requirement (!)
- Design variation: Instead of digest of CEK, one could use encryption of a dummy value or of derived data.