Encrypted Payloads in SUIT

draft-ietf-suit-firmware-encryption

Status

- Three draft updates since last IETF meeting.
  - Ken Takayama and David Brown joined as co-authors
  - Had dependency on COSE-HPKE since some time (which also keeps changing)
  - New dependency on draft-ietf-cose-aes-ctr-and-cbc
- Title changed from “Software Encryption with SUIT Manifests” to “Encrypted Payloads in SUIT Manifests”
- Text improvements throughout the document.
- Approach for using encryption changed → next slide

Old Approach

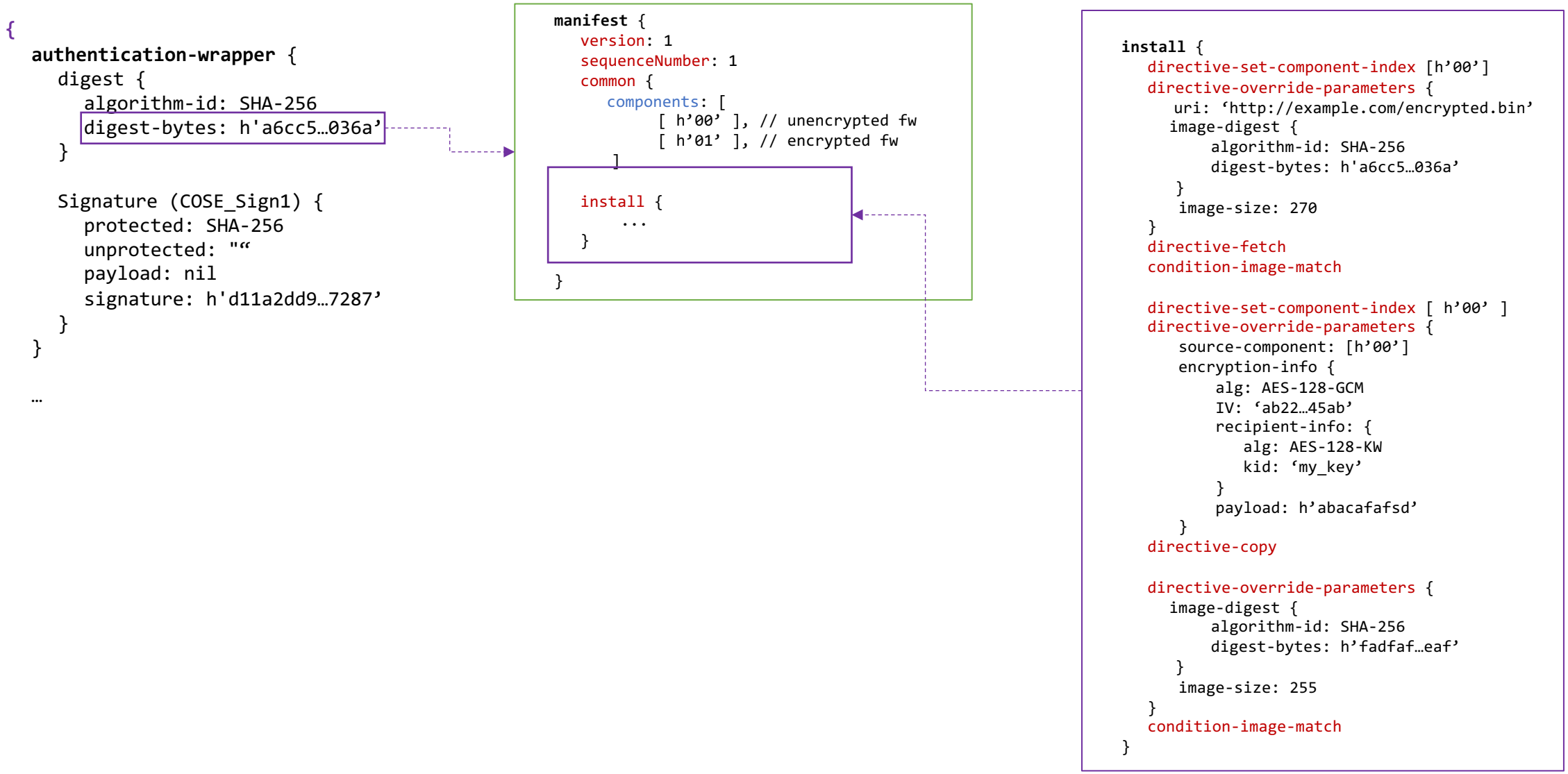
- Encryption information was in the SUIT Envelope and therefore not covered by a signature/MAC.
- Allowed distribution system to add encryption information after the author signed the manifest.
- Without integrated payloads the trust domain functionality had to be used since the URL in the manifest couldn’t be changed.

New Approach

- Encryption info is part of the manifest.
- Requires distribution system to create a second manifest with a dependency on the manifest created by the author.
  - Requires two manifest in this case but simplifies the security story.
  - See example at the end of the deck
- Firmware encryption draft focuses only on the case where the author signs and encrypt.
  - Leaves the dependency case to draft-ietf-suit-trust-domains.

Example with author signing manifest and encrypting firmware

Encrypted firmware image is available at
http://example.com/encrypted.bin


{
authentication-wrapper {
digest {
algorithm-id: SHA-256
digest-bytes: h'a6cc5…036a'
}
Signature (COSE_Sign1) {
protected: SHA-256
unprotected: ""
payload: nil
signature: h'd11a2dd9…7287'
}
}
…
manifest {
version: 1
sequenceNumber: 1
common {
components: [
[h'00'], // unencrypted fw
[h'01'], // encrypted fw
]
install {
...
}
}
install {
directive-set-component-index [h'00']
directive-override-parameters {
uri: 'http://example.com/encrypted.bin'
image-digest {
algorithm-id: SHA-256
digest-bytes: h'a6cc5…036a'
}
image-size: 270
}
directive-fetch
condition-image-match
directive-set-component-index [h'00']
directive-override-parameters {
source-component: [h'00']
encryption-info {
alg: AES-128-GCM
IV: 'ab22…45ab'
recipient-info: {
alg: AES-128-KW
kid: 'my_key'
}
payload: h'abacafafsd'
}
directive-copy
directive-override-parameters {
image-digest {
algorithm-id: SHA-256
digest-bytes: h'fadfaf…eaf'
}
image-size: 255
}
condition-image-match
}

Example with author signing manifest and distribution system encrypting firmware

(for inclusion in the trust domains draft)

Outline of the idea

- Distribution System's manifest contains
  - fetch Author's manifest from "https://ds.example.com/manifest_a.suit"
  - fetch encrypted firmware from "https://ds.example.com/encrypted-firmware.bin"
  - decrypt it with parameter-encryption-info <= in firmware encryption draft
  - triggers Author's manifest with process-dependency <= in trust domains draft
- Author's manifest contains
  - only validate the decrypted firmware

In detail

Distribution System's Manifest

```
/ SUIT_Envelope = / {
  suit-authentication-wrapper: SUIT_Authentication / by Distribution System /,
  / inside the signature /
  suit-manifest : << {
    suit-common: << {
      suit-components: [
        [h'00'] / to be decrypted firmware /,
        [h'01'] / encrypted firmware /,
      ]
    } >>,
    suit-dependency-resolution: << [
      suit-directive-set-dependency-index 0,
      suit-directive-override-parameters {
        suit-parameter-uri: "https://ds.example.com/manifest_a.suit"
      },
      suit-directive-fetch
    ] >>,
    suit-install: << [
      / loads encrypted firmware /
      suit-directive-set-component-index 1 / [h'01'] /,
      suit-directive-override-parameters {
        suit-parameter-uri: "https://ds.example.com/encrypted-firmware.bin"
      },
      suit-directive-fetch,

      / decrypts it /
      suit-directive-set-component-index 0 / [h'00'] /,
      suit-directive-override-parameters {
        suit-parameter-source-component: 1 / [h'01'] /
        suit-parameter-encryption-info: SUIT_Encryption_Info / how to decrypt it /
      },
      suit-directive-copy,

      / request to parse Author's manifest /
      suit-directive-set-dependency-index 0 / Author's Manifest /,
      suit-directive-process-dependency
    ] >>
  } >>
}
```

triggers

Depends

Author's Manifest

```
/ SUIT_Envelope = / {
  suit-authentication-wrapper: SUIT_Authentication / by Author /,
  / inside the signature /
  suit-manifest : << {
    suit-common: << {
      suit-components: [
        [h'00'] / decrypted firmware /
      ]
    } >>,
```

```
    suit-install: << [
      / verify that the decrypted firmware is intact /
      suit-directive-set-component-index 0 / [h'00'] /,
      suit-directive-override-parameters {
        suit-parameter-image-digest: <<digest of raw firmware image [h'00']>>,
        suit-parameter-image-size: <<size of raw firmware image [h'00']>>
      },
      suit-condition-image-match
    ] >>
  } >>
}
```