Encrypted Payloads in SUIT Manifests

H. Tschofenig, R. Housley, B. Moran, D. Brown, K. Takayama
(Special thanks to L. Lundblade for his review comments)

IETF#117

IETF®

Overview

- Encryption of firmware, software and any other payload.
- Relies on IETF SUIT manifest for conveying meta-data.
- Efficient encryption of payloads for
    1. Single sender – single recipient,
    2. Single sender – multiple recipients.
- Two content key distribution mechanisms supported:
    1. AES-KW
    2. Ephemeral-static Diffie-Hellman
- Flexible deployment modes:
    - Author signing and encrypting payloads (requires author to know the recipients)
    - Author delegates encryption to another party (utilizes SUIT multiple trust domains)

Draft Update

- Improved wording for better readability
- Removed left-over functionality regarding CEK Verification
  - Changed integration in the SUIT manifest prior to IETF#116
- Examples updated based on prototyping with t_cose (see https://github.com/laurencelundblade/t_cose/tree/dev)
- Use a two-layer structure for ES-DH (instead of three-layer structure)

- Updated context information structure
  - Now has a dependency on draft-isobe-cose-key-thumbprint for use in context information structure.

Context Information Structure from draft-ietf-suit-firmware-encryption-13

Hash of Public Key of Sender

Hash of Public Key of Recipient

Algorithm Info

Context

```
PartyInfoSender = (
    identity : bstr,
    nonce : nil,
    other : bstr .size 0
)

PartyInfoRecipient = (
    identity : bstr,
    nonce : nil,
    other : bstr .size 0
)

COSE_KDF_Context = [
    AlgorithmID : int,
    PartyUInfo : [ PartyInfoSender ],
    PartyVInfo : [ PartyInfoRecipient ],
    SuppPubInfo : [
        keyDataLength : uint,
        protected : bstr .cbor recipient_header_pr_map,
        other: bstr "SUIT Payload Encryption"
    ],
    SuppPrivInfo : bstr .size 0
]
```

Context Information Structure Examples

- KeyMint - ProtectedData.aidl - Android Code
  - Uses public keys of sender and recipients. Additionally encodes the strings “client” and “server”
  - Algorithm information
- DIDComm - DIDComm Messaging Specification v2.1
  - Uses tag of the AEAD cipher as input to the context info structure.
- LAMPS
  - Uses algorithm information and (optionally) user supplied data.
  - Sender and recipient identity remains empty.

Proposed Context Information Structure

NULL

NULL

Algorithm Info
(AES-KW)

Protocol
Context

```
PartyInfoSender = (
    identity : bstr,
    nonce : nil,
    other : bstr .size 0
)

PartyInfoRecipient = (
    identity : bstr,
    nonce : nil,
    other : bstr .size 0
)

COSE_KDF_Context = [
    AlgorithmID : int,
    PartyUInfo : [ PartyInfoSender ],
    PartyVInfo : [ PartyInfoRecipient ],
    SuppPubInfo : [
        keyDataLength : uint,
        protected : bstr .cbor recipient_header_pr_map,
        other: bstr "SUIT Payload Encryption"
    ],
    SuppPrivInfo : bstr .size 0
]
```

ECDH-ES
+ A128KW

Two Layer ES-DH Structure

Content Encryption Layer

Recipient

```
96(
  [
   / protected / h'a10101' / {
       \ alg \ 1:1 \ AES-GCM-128 \
     } / ,
   / unprotected / {
     / iv / 5:h'26682306D4FB28CA01B43B80'
     },
   / encrypted firmware /
    h'F21AC5881CD6FC45754C65790F806C81A57
      B8D96C1988233BF40F670172405B5F107FD',
    [
       / protected / h'A101381C' / {
           \ alg \ 1:-29 \ ECDH-ES + A128KW \
         } / ,
         h'A101381C',
       / unprotected / {
             / ephemeral / -1: {
                   / kty / 1:2,
                   / crv / -1:1,
                   / x / -2:h'415A8ED270C4B1F10B0A2D42B28EE602
                                8CE25D74552CB4291A4069A2E989B0F6',
                   / y / -3:h'CCC9AAF60514B9420C80619A4FF068BC
                                1D77625BA8C90200882F7D5B73659E76'
                 },
                 / kid / 4:'kid-1'
        },
        / ciphertext - CEK encrypted with KEK /
        h'B37CCD582696E5E62E5D93A555E9072687D6170B122322EE'
    ]
  ]
)
```

Complete Example

```
/ SUIT_Envelope_Tagged = / 107({
  / authentication-wrapper / 2: << [
    / digest: / << [
      / algorithm-id: / -16 / SHA-256 /,
      / digest-bytes: / h'8c757f37c9af51cec00b9…7b5baae8aaee33'
    ] >>,
    / signatures: / << 18([
      / protected: / << {
        / alg / 1: -7 / ES256 /
      } >>,
      / unprotected: / {
      },
      / payload: / null,
      / signature: / h'bda73be94f7a442c895e514e…714e714e4fe'
    ]) >>
  ] >>,
 / manifest(verified) / 3: << {
    / manifest-version / 1: 1,
    / manifest-sequence-number / 2: 1,
    / common / 3: << {
      / components / 2: [
        ['decrypted-firmware']
      ]
    } >>,
/ install(verified) / 17: << [
      / directive-set-component-index / 12, 0,
      / directive-override-parameters / 20, {
        / content / 18: h'67becf8b…8a2',
        / encryption-info / 19: << 96([
          / protected: / << {
            / alg / 1: 1 / A128GCM /
          } >>,
          / unprotected: / {
            / IV / 5: h'046f37fb2d1a75402cb4452be877d1e4'
          },
          / payload: / null,
          / recipients: / [
            [
              / protected: / << {
                / alg / 1: -29 / ECDH-ES + A128KW /
              } >>,
              / unprotected: / {
                / ephemeral key / -1: {
                  / kty / 1: 2 / EC2 /,
                  / crv / -1: 1 / P-256 /,
                  / x / -2: / h'8802d5c68a1..2b66e7381edd7b',
                  / y / -3: / h'1b1b1ef6c…02705394d',
                },
                / kid / 4: 'kid-3'
              },
              / CEK: / h'aab40270..9dd7b'
            ]
          ]
        ])>>
      },
      / directive-write / 18, 15
    ] >>
  } >>
})
```


IETF

Complete Example

```
/ SUIT_Envelope_Tagged = / 107({
  / authentication-wrapper / 2: << [
    / digest: / << [
      / algorithm-id: / -16 / SHA-256 /,
      / digest-bytes: / h'8c757f37c9af51cec00b9…7b5baae8aaee33'
    ] >>,
    / signatures: / << 18([
      / protected: / << {
        / alg / 1: -7 / ES256 /
      } >>,
      / unprotected: / {
      },
      / payload: / null,
      / signature: / h'bda73be94f7a442c895e514e…714e714e4fe'
    ]) >>
  ] >>,
  / manifest(verified) / 3: << {
    / manifest-version / 1: 1,
    / manifest-sequence-number / 2: 1,
    / common / 3: << {
      / components / 2: [
        ['decrypted-firmware']
      ]
    } >>,
```

Digital Signature of the hash of the manifest

```
    / install(verified) / 17: << [
        / directive-set-component-index / 12, 0,
        / directive-override-parameters / 20, {
          / content / 18: h'67becf8b…8a2',
          / encryption-info / 19: << 96([
            / protected: / << {
              / alg / 1: 1 / A128GCM /
            } >>,
            / unprotected: / {
              / IV / 5: h'046f37fb2d1a75402cb4452be877d1e4'
            },
            / payload: / null,
            / recipients: / [
              [
                / protected: / << {
                  / alg / 1: -29 / ECDH-ES + A128KW /
                } >>,
                / unprotected: / {
                  / ephemeral key / -1: {
                    / kty / 1: 2 / EC2 /,
                    / crv / -1: 1 / P-256 /,
                    / x / -2: / h'8802d5c68a1..2b66e7381edd7b',
                    / y / -3: / h'1b1b1ef6c…02705394d',
                  },
                  / kid / 4: 'kid-3'
                },
                / CEK: / h'aab40270..9dd7b'
              ]
            ]
          ])>>
        },
        / directive-write / 18, 15
      ] >>
    } >>
  })
```


IETF

Complete Example

```
/ SUIT_Envelope_Tagged = / 107({
  / authentication-wrapper / 2: << [
    / digest: / << [
      / algorithm-id: / -16 / SHA-256 /,
      / digest-bytes: / h'8c757f37c9af51cec00b9…7b5baae8aaee33'
    ] >>,
    / signatures: / << 18([
      / protected: / << {
        / alg / 1: -7 / ES256 /
      } >>,
      / unprotected: / {
      },
      / payload: / null,
      / signature: / h'bda73be94f7a442c895e514e…714e714e4fe'
    ]) >>
  ] >>,
  / manifest(verified) / 3: << {
    / manifest-version / 1: 1,
    / manifest-sequence-number / 2: 1,
    / common / 3: << {
      / components / 2: [
        ['decrypted-firmware']
      ]
    } >>,
```

Encryption info with ephemeral-static DH

```
/ install(verified) / 17: << [
    / directive-set-component-index / 12, 0,
    / directive-override-parameters / 20, {
      / content / 18: h'67becf8b…8a2',
      / encryption-info / 19: << 96([
        / protected: / << {
          / alg / 1: 1 / A128GCM /
        } >>,
        / unprotected: / {
          / IV / 5: h'046f37fb2d1a75402cb4452be877d1e4'
        },
        / payload: / null,
        / recipients: / [
          [
            / protected: / << {
              / alg / 1: -29 / ECDH-ES + A128KW /
            } >>,
            / unprotected: / {
              / ephemeral key / -1: {
                / kty / 1: 2 / EC2 /,
                / crv / -1: 1 / P-256 /,
                / x / -2: / h'8802d5c68a1..2b66e7381edd7b',
                / y / -3: / h'1b1b1ef6c…02705394d',
              },
              / kid / 4: 'kid-3'
            },
            / CEK: / h'aab40270..9dd7b'
          ]
        ]
      ])>>
    },
    / directive-write / 18, 15
  ] >>
} >>
})
```

Encrypted content

Write Content to component

IETF

Next Steps

- Update context info structure (again)
- Update ES-DH example based on latest t_cose implementation containing the context information structure.
- Add complete example describing SUIT manifest
  - Thanks to hackathon code progress with https://github.com/kentakayama/libcsuit
- Then, ready for publication.