JOSE Header Integrity Options

Design Goals

Need to integrity-protect some header parameters (e.g., digest algorithm for PSS [RFC6211])

Need compatibility with all major AEAD algorithms, notably GCM

State of the Art (-09)

Need to integrity-protect some header parameters (e.g., digest algorithm for PSS [RFC6211])

- **All header parameters are protected**

Need compatibility with all major AEAD algorithms, notably GCM

- **Current format causes GCM nonce re-use**

State of the Art (-09)

Need to integrity-protect some header parameters (e.g., digest algorithm for PSS [RFC6211])

All header parameters are protected

Need compatibility with all major AEAD algorithms, notably GCM

Current format causes GCM nonce re-use

BAD!

Proposals

1. Continue to protect everything, but combine multiple recipients' data together [JWE-10]

2. Only protect what really needs to be protected (omitting per-recipient fields)

Proposals

1. Continue to protect everything, but combine multiple recipients' data together [JWE-10]

    Prevents incrementally adding receipients

2. Only protect what really needs to be protected (omitting per-recipient fields)

    Requires sender to choose fields

Neither one needs to significantly change the single-recipient case

Beauty contest!

Parameters

- Encrypt a 32-byte payload with GCM
- Two recipients, one wrapped with AES-KW, the other with RSA-OAEP
- For argument, "enc" needs protection

```
Overall:
    "enc":"A128GCM"
    "initialization_vector":"7k5QJi1p97jM-a2uQG7yhg"
    "ciphertext":"NDt1oc7joRuKF3ZzUglaJtPFCrQFB_25pg5EvNSC74E"
    "authentication_tag":"l26KVB14Z9pJ7__e2tHvWg"
Recipient 1:
    "alg":"A128KW"
    "kid":"1"
    "encrypted_key":"x6hcRL82gzIti0j_WROBADqm9OuW7_XW"
Recipient 2:
    "alg":"RSA-OAEP"
    "kid": "2"
    "encrypted_key":"dPF-nRkxmuNfzPsPIB14rEfzSiFSn1l1O4JLVI7b6R
                     -Sz3aU1qBvdalleqx55mafVgmvSEyo5uo_lH6JQEHCjA"
```

Current (-09)

```
// header1 = base64({"alg":"A128KW","enc":"A128GCM","kid":"1"})
// header2 = base64({"alg":"RSA-OAEP","enc":"A128GCM","kid":"2"})

{"recipients":[
  {"header":"eyJhbGciOiJBMTI4S1ciLCJlbmMiOiJBMTI4R0NNIiwia2lkIjoiMSJ9Cg",
   "encrypted_key":"x6hcRL82gzIti0j_WROBADqm9OuW7_XW"},
  {"header":"eyJhbGciOiJSU0EtT0FFUCIsImVuYyI6IkExMjhHQ00iLCJraWQiOiIyIn0K",
   "encrypted_key":"dPF-nRkxmuNfzPsPIB14rEfzSiFSn1l1O4JLVI7b6R
                    -Sz3aU1qBvdalleqx55mafVgmvSEyo5uo_lH6JQEHCjA"}],
  "initialization_vector":"7k5QJi1p97jM-a2uQG7yhg",
  "ciphertext":"NDt1oc7joRuKF3ZzUglaJtPFCrQFB_25pg5EvNSC74E",
  "authentication_tag":"l26KVB14Z9pJ7__e2tHvWg"
}
```

Two AAD values, one IV

BAD!

Proposal #1: Everyone Together (-10)

```
// header1 = base64({"alg":"A128KW","enc":"A128GCM","kid":"1"})
// header2 = base64({"alg":"RSA-OAEP","enc":"A128GCM","kid":"2"})

header1~header2
.key1~key2
.initialization_vector.ciphertext.authentication_tag
```

```
eyJhbGciOiJBMTI4S1ciLCJlbmMiOiJBMTI4R0NNIiwia2lk
IjoiMSJ9Cg~eyJhbGciOiJSU0EtT0FFUCIsImVuYyI6IkExM
jhHQ00iLCJraWQiOiIyIn0K.x6hcRL82gzIti0j_WROBADqm9
OuW7_XW~dPF-nRkxmuNfzPsPIB14rEfzSiFSn1l1O4JLVI7
b6R-Sz3aU1qBvdalleqx55mafVgmvSEyo5uo_lH6JQEHCjA.
7k5QJi1p97jM-a2uQG7yhg.NDt1oc7joRuKF3ZzUglaJtPFCrQF
B_25pg5EvNSC74E.l26KVB14Z9pJ7__e2tHvWg
```

All header parameters and encrypted keys protected (some twice!)

Aside: CMS

Only protect what needs protecting

```
AuthEnvelopedData ::= SEQUENCE {
  version CMSVersion,
  originatorInfo [0] IMPLICIT OriginatorInfo OPTIONAL,
  recipientInfos RecipientInfos,
  authEncryptedContentInfo EncryptedContentInfo,
  authAttrs [1] IMPLICIT AuthAttributes OPTIONAL,
  mac MessageAuthenticationCode,
  unauthAttrs [2] IMPLICIT UnauthAttributes OPTIONAL }

SignerInfo ::= SEQUENCE {
  version CMSVersion,
  sid SignerIdentifier,
  digestAlgorithm DigestAlgorithmIdentifier,
  signedAttrs [0] IMPLICIT SignedAttributes OPTIONAL,
  signatureAlgorithm SignatureAlgorithmIdentifier,
  signature SignatureValue,
  unsignedAttrs [1] IMPLICIT UnsignedAttributes OPTIONAL }
```

Proposal #2: Only what's needed

```
// auth = base64({"enc":"A128GCM"})

{ "authenticated_attributes":" eyJlbmMiOiJBMTI4R0NNIn0K",
  "recipients":[
    { "alg":"A128KW",
      "kid":"1",
      "encrypted_key":"x6hcRL82gzIti0j_WROBADqm9OuW7_XW" },
    { "alg":"RSA-OAEP",
      "kid":"2",
      "encrypted_key":"dPF-nRkxmuNfzPsPIB14rEfzSiFSn1l1O4JLVI7b6R
                       -Sz3aU1qBvdalleqx55mafVgmvSEyo5uo_lH6JQEHCjA" }
  ],
  "initialization_vector":"7k5QJi1p97jM-a2uQG7yhg",
  "ciphertext":"NDt1oc7joRuKF3ZzUglaJtPFCrQFB_25pg5EvNSC74E",
  "authentication_tag":"l26KVB14Z9pJ7__e2tHvWg"
}
```

Only protect what needs protecting

No repetition
Everything else JSON

Single-Recipient Case

```
// Ignore the second recipient
// header = auth = base64({"alg":"A128KW","enc":"A128GCM","kid":"1"})

// Proposal #1 - No tildes
eyJhbGciOiJBMTI4S1ciLCJlbmMiOiJBMTI4R0NNIiwia2lkIjoiMSJ9Cg
.x6hcRL82gzIti0j_WROBADqm9OuW7_XW
.7k5QJi1p97jM-a2uQG7yhg.NDt1oc7joRuKF3ZzUglaJtPFCrQF
B_25pg5EvNSC74E.l26KVB14Z9pJ7__e2tHvWg

// Proposal #2 - Auth everything
{ "authenticated_attributes":" eyJhbGciOiJBMTI4S1ciLCJlbmMiOiJBMTI
                               4R0NNIiwia2lkIjoiMSJ9Cg",
  "encrypted_key":"x6hcRL82gzIti0j_WROBADqm9OuW7_XW",
  "initialization_vector":"7k5QJi1p97jM-a2uQG7yhg",
  "ciphertext":"NDt1oc7joRuKF3ZzUglaJtPFCrQFB_25pg5EvNSC74E",
  "authentication_tag":"l26KVB14Z9pJ7__e2tHvWg"
}
```

Only difference is whether encrypted key is protected

Discuss!