JWS, JWE, JWK, and JWA

John Bradley

IETF 86

March 13, 2013

JOSE Status by Specification

- JWS
  - No Significant changes since March 2011
  - Over a dozen implementations, with several more since IETF 85
- JWE
  - No significant changes post the one agreed at IETF84.
    - Last significant change was using only algorithms with integrity after IETF 84
  - At least 6 known implementations
- JWK
  - Semantically very stable
    - A few syntax changes were made before IETF 85
  - Also over a dozen known implementations
- JWA
  - Open issues largely closed after IETF 84
  - Used in JWS, JWE, JWK implementations

Primary Remaining Open Issue

- Criticality of understanding header fields
- No consensus as a result of the poll
- The chairs got some of us together Monday
- Together we arrived at a proposed resolution
- Following slides describe proposed resolution
  - Has five parts

Criticality Resolution Part 1 of 5

- Change the language
  - "Additional members MAY be present in the JWK. If present, they MUST be understood by implementations using them."
- to
  - "Additional members MAY be present in the JWK. If not understood by implementations encountering them, they MUST be ignored."
- Make the same change for JWK Set as well

Criticality Resolution Part 2 of 5

- Characterize all existing JWS and JWE header fields as either must be understood or may be ignored:
  - "alg", "enc", and "zip" must be understood
  - "kid", "x5u", "x5c", "x5t", "jwk", "jku", "typ", and "cty" can be ignored because while not using them may result in the inability to process some signatures or encrypted content, this will not result in a security violation - just degraded functionality
  - "epk", "apu", "apv", "epu", and "epv" must be understood and used when "alg" or "enc" values requiring them are used, and otherwise may be ignored

Criticality Resolution Part 3 of 5

- Define new "crit" (critical) header field that lists which additional fields not defined in the base specs must be understood and acted upon when present. For instance, an expiration-time field could be marked as must-be-understood-and-acted-upon:

```
{"alg":"ES256",
 "crit":["exp"],
 "exp":1363284000
}
```

Criticality Resolution Part 4 of 5

- All additional header fields not defined in the base specifications and not contained in the "crit" list MUST be ignored if not understood

Criticality Resolution Part 5 of 5

- Define a new "asd" (application-specific data) header field whose value is a JSON structure whose contents are opaque to and ignored by JWS and JWE implementations but for which its contents MUST be provided to applications using JWS or JWE, provided that the signature/MAC validation or decryption operation succeeds
- The intended use of this field is to have a standard place to provide application-specific metadata about the payload or plaintext
- *Note that this part is independent of the other 4*

Other Key Remaining Issue (#3)

- Currently AES-CBC+HMAC-SHA encryption uses Concat KDF with a CMK
  - Some have objected to its use, and complexity
- Alternative is to use key that is the concatenation of the AES, HMAC keys
  - 384 bits for A128CBC+HS256
  - 768 bits for A256CBC+HS512
- Which does the WG want to do?

New Issues Filed - Issue #2

- No key management for MAC
  - This is a duplicate of the issue "Add other than pre-shared MAC key", which was closed in the October 24, 2012 consensus call

New Issues Filed - Issue #4

- Impossible to separate wrapped key from encrypted data
  - This seems to not be true, as the direct encryption mode enables this separation
  - This issue should be closed accordingly

New Issues Filed - Issue #5

- Unclear instructions for key management
  - Fix will be non-normative clarifications

New Issues Filed - Issue #6

- Unclear requirements levels on fields
- Most of the fields in JWE and JWS are listed as OPTIONAL, even though they are REQUIRED in some cases
  - The resolution to the "must understand" issue will also address this

New Issues Filed - Issue #7

- Algorithm identifiers/parameters incompatible with WebCrypto
  - They have different purposes, so this isn't a problem in practice
  - Also, WebCrypto could use some of the JWA identifiers where they make sense
    - This is their responsibility - not ours
  - We should close this issue, especially since it is largely a duplicate of the issue "Support Multiple types for algorithms", which was closed in the October 22, 2012 consensus call

New Issues Filed - Issue #8

- Direct mode for encryption needs security analysis
  - We can do this analysis
  - (Note that we already have a consensus call result to include direct encryption)

New Issues Filed - Issue #9

- Add "spi" (Security Parameters Index) field
  - Several people have requested that this be a separate ID
  - We should re-evaluate after there's a complete ID whether to merge this functionality into the existing specs
  - Since is separable functionality that could remain in its own draft, this shouldn't delay WGLC

New Issues Filed - Issue #10

- There should be no MTI algorithms in JWA. It should be up to applications to define required algorithms.
  - The indication from the IESG is that we won't get past them without MTI algs

New Issues Filed - Issue #11

- Whether to change the JWE encoding to use the binary encoding/decoding rules for the Initialization Vector and Integrity Value specified in RFC 5116
  - No existing crypto libraries surveyed do this
    - This change would require extra work by all implementations
  - JWE already specifies a simple means of representing these values
  - Other systems, including CMS don't do this
  - JWE supports variable length values for these fields, whereas RFC 5116 is less flexible
  - No practical benefit to change
  - Therefore we should reject this issue now

New Issues Filed - Issue #12

- Remove x5c from JWE
  - It duplicates equivalent functionality available x5u & kid
- Alternatively it needs to be the certificate used to encrypt(The recipient)
  - The chain is pointless and doesn't need to be validated.

Conclusions

- The specs are mature and implemented
  - They are already in production use
- Most open issues have been closed
- After applying the "must understand" resolution and deciding what key format to use with AES CBC, we'll be ready for WGLC

Backup Slides

Poll Results on Header Criticality

- FIRST POLL: Should all header fields be critical for implementations to understand?
  - 19 Yes, 12 No (61% Yes, 39% No)
- SECOND POLL: Should the result of the first poll be "YES", should text like the following be added?
  - 25 Yes, 6 No (81% Yes, 19% No)
- THIRD POLL: Should the result of the first poll be "NO", which syntax would you prefer for designating the header fields that may be ignored if not understood?
  - 20 A, 3 B, 6 C, 2 No opinion (65% A, 10% B, 19% C, 6% No opinion)