Key Managed JSON Web Signature (KMJWS)

Mike Jones

IETF 92 – Dallas

March 24, 2015

Background and Motivation

- Wrote draft-jones-jose-key-managed-json-web-signature to:
  - Write down a straightforward way to do it
    - Reusing features already present in JWE and JWS
  - Satisfy future requests for this functionality, should they occur
  - Provide input to CBOR JOSE binding discussions
    - Including what we might do the same and differently
- Wrote it now because the WG may be closed

KMJWS Structure

- A KMJWS (yes, it’s a terrible name) contains:
  - JOSE Header
  - Encrypted Key
  - Payload
  - MAC
- Compact Serialization has 4 parts
- JSON Serialization uses JWS and JWE names
  - Different recipients use different MAC keys

KMJWS JOSE Header

- Example:

```
{"alg":"RSA-OAEP","mac":"HS256"}
```

- “alg” values from JWE key management algs
- “mac” values from JWS MAC algs

A Takeaway from This Exercise

- The structure could be more uniform between JWS, JWE, & KMJWS objects if we didn’t overload the “alg” header parameter name
- For instance, use these alternative header parameters, when applicable:
  - “int” – Integrity algorithm (MAC or signature)
  - “enc” – Content encryption algorithm
  - “kma” – Key management algorithm
- The current JWS and JWE direct encryption cases wouldn’t include “kma”
- *Possibly relevant when designing CBOR encoding*

Still not clear there’s demand for this

- JOSE Issue #2 (No key management for MAC) was closed as *won’t fix*
- Jim Schaad wrote when closing it:

  “The working group has already considered this and has determined that it will not be addressed. Until a request for the feature comes in from a group such as the WebCrypto group it will not be re-considered.”
- Not presupposing that there’s demand now

Next Steps and Conclusions

- I plan to revise the individual draft to address comments by Jim Schaad
- It’s up to us whether to take this further or not

- Even if just as though experiment for the possible CBOR binding, I hope people find this exercise useful