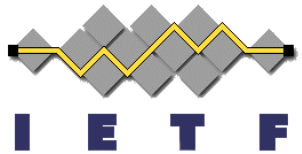
JWS [RFC 7515] signing without base64url encoding the payload

draft-jones-jose-jws-signing-input-options-00

Mike Jones

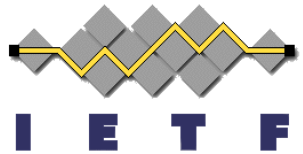
IETF 93
Prague
July 2015

JSON Web Signature (JWS) [RFC 7515] Representation



- Defines signing/MACing payloads using JSON-based data structures
- Base64url encodes the payload and other fields
  - Encoding uses characters 0-9, A-Z, a-z, -, _
  - Makes fields URL-safe
  - Prevents changes to payload during transmission
  - Results in a 33% payload size expansion
- Compact Serialization representation:
  - BASE64URL(headers).BASE64URL(payload).BASE64URL(signature)
- An equivalent JSON Serialization representation also defined

Detached Payloads



- Appendix F of JWS describes using detached content
  - Integrity protecting a payload not included in the JWS
- Applications can use this to sign content that is transmitted without modification
  - For instance, signing an HTTP body
- Calculation identical to normal payloads – involving base64url encoding the payload

New draft defines option to not encode the payload

- draft-jones-jose-jws-signing-input-options
- Current syntax (subject to change):
  - “b64”: false
- Always safe for detached payloads
- Also safe for included payloads that happen to be URL-safe and don’t include “.”

Also defines option to not integrity protect headers

- Normal signature input includes header parameters
- In some circumstances, it's OK and more convenient to not integrity protect them
  - More convenient for huge payloads
    - Since you don't have to allocate memory to concatenate the headers and payload as an input to the signature/MAC function
- Current syntax (subject to change):
  - "sph": false

Draft Status

- Currently an individual draft
- JOSE chairs currently considering whether to adopt as a JOSE working group draft
- See the JOSE mailing list thread
  - [jose] way forward for two remaining drafts