JWS Multiple Payload Option

draft-waite-jws-multi-payload

David Waite, Ping Identity
IETF 117, San Francisco
July 25, 2023
Background

- JSON Web Proofs describes a multiple payload format using tilde ("~") delimiters between payloads
- This allows applications to represent their messages as multiple independent data items
- JSON Proof Algorithms can operate on those items individually (e.g., selective disclosure)
- Applications will still need to describe the semantics of those multi-payload messages and how to act on them
What is JWS Multi-Payload Option?

- Adds the JWP tilde-delimited multi-payload format as a JWS extension
- Defines a new representation of JSON and compact serializations, and of signing input
- Describes multi-payload-aware signature algorithms, operating on the header and payload(s) directly, rather than the JWS Signing Input
- Describes interactions with other JWS features (e.g., detached payloads, unencoded payloads)
- Allows greater reuse in data formats/semantics between JWS and JWP
Example

```json
{
    "alg": "HS256",
    "mp": true,
    "crit": ["mp"]
}
```

IyBKU09OIFdlYiBQcm9vZnMKClNwZWNpZmlj ~
YXRpb24gd29yayBmb3IgSlNPTiBXZWigUHJvb2ZzCgpUaGUgSlNPTiBXZWigUHJvb2ZzIGVmZm9ydCB ~
haW1zIHR

IyBKU09OIFdlYiBQcm9vZnMKClNwZWNpZmljYXRpb24g
How Does it Work?
Syntax

• Defines:
  • Multi-payload compact serialization using tilde delimiters
  • JSON serialization using "payloads" key
  • JWS Signing Input based on protected header and payloads
Multi-Payload Aware Algorithms

• Existing algorithms operate on JWS Signing Input as octets
  • Can support multiple payloads using new signing rule

• Multi-payload-aware algorithms take protected header and payloads as inputs directly (each as octets)

• Proof algorithms would separately become multi-payload-aware algorithms
  • With further capabilities around controlling information release
Interactions with other JWS options

• Defines a new "mp" protected header to indicate multi-payload format
  • Similar to "b64": false, defines requirements on the payload and JWS signing inputs
• Multi-payload algorithms mandate "mp": true and can omit it
  • JWS implementations fail due to unknown algorithms if they don't support "mp"
• Backward-compatibility options described leveraging "b64":false for existing algorithms
  • An existing JWS library without support for “mp” will pass the payloads through as input for application processing
• Detached payloads dictate the application detaches _all_ payloads, no mixed mode
Motivation
Bridging JWS and JWP/JPA

• Today JWP and JPA leverage the concepts and pieces:
  • JWK
  • Protected header names
• JWS Multiple Payloads would blend things together
• Proofs become multi-payload-aware algorithms with additional capabilities
  • E.g., holder can derive new proofs to control information release
• JWP describes how to present these derived forms
Breaking the Dependency Graph

- Keep the flexibility for applications to use arbitrary payload data formats
- JSON Proof Tokens today describe a multi-payload alternative to JWTs
- When applications use multiple-payload data
  - We don't want to require them to use proof algorithms like BBS or zkSnarks if they do not need them
New JSON Web Proofs Structure*
Proposed Structure (1/3)

- JWS Multi-Payload Option describes:
  - Multi-payload form
  - Multi-payload aware algorithms

- JSON Web Proofs describes:
  - Roles and capabilities
  - Issuance and presentation serializations
  - Application use for issuance and for presentation
Proposed Structure (2/3)

• JSON Proof Algorithms describes:
  • Additional requirements for a multi-payload-aware algorithm to be a proof
  • Common properties a proof algorithm may have
  • Security and privacy considerations around algorithms with particular properties
  • Initial proof algorithm entries (e.g., BBS) and registration process for future algorithms
Proposed Structure (3/3)

- JSON Proof Tokens describes:
  - Multi-payload token format that works with JWS or JWP
  - Similarities with JWT (and where it needed to deviate)
  - Any additional security/privacy considerations around use with JWS, or JWP algorithms with particular properties
Next Steps

• Reviews by working group members
• Possible working group adoption after reviews