jose
Internet-Draft
Intended status: Standards Track
Expires: 8 May 2026

M. Jones
Self-Issued Consulting
D. Waite
J. Miller
Ping Identity

4 November 2025

JSON Proof Token and CBOR Proof Token
draft-ietf-jose-json-proof-token-12

Abstract

JSON Proof Token (JPT) is a compact, URL-safe, privacy-preserving representation of claims to be transferred between three parties. The claims in a JPT are encoded as base64url-encoded JSON objects that are used as the payloads of a JSON Web Proof (JWP) structure, enabling them to be digitally signed and selectively disclosed. JPTs also support reusability and unlinkability when using Zero-Knowledge Proofs (ZKPs).

A CBOR-based representation of JPTs is also defined, called a CBOR Proof Token (CPT). It has the same properties as JPTs, but uses the JSON Web Proof (JWP) CBOR Serialization, rather than the JSON-based JWP Compact Serialization.

Table of Contents

1. Introduction

JSON Proof Token (JPT) is a compact claims representation format intended to be used in the same ways as a JSON Web Token (JWT) [RFC7519], but with additional support for selective disclosure and unlinkability. JPTs encode claim values to be transmitted as payloads of a JSON Web Proof (JWP) [I-D.ietf-jose-json-web-proof]. JPTs are always represented using the JWP Compact Serialization. The corresponding claim names are not transmitted in the payloads and are stored in a separate structure that can be externalized and shared across multiple JPTs.

Likewise, CBOR Proof Token (CPT) is a similar compact claims representation format intended to be used in the same ways as a CBOR Web Token (CWT) [RFC8392], but with the same support for selective disclosure and unlinkability. CPTs are represented using the JWP CBOR Serialization. The corresponding claim names are not transmitted in the payloads and are stored in a separate structure that can be externalized and shared across multiple CPTs.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Background

JWP defines a container binding together an integrity-protected Header, one or more payload slots, and a cryptographic proof. It does not define how claims are organized into payloads and what formats they are in. JPTs are intended to be as close to a JWT as possible, while also supporting the selective disclosure and unlinkability of JWPs. Likewise, CPTs are intended to be as close to a CWT as possible, while also supporting the selective disclosure and unlinkability of JWPs.

4. Design Considerations

The rationale behind the design for JSON Proof Tokens and CBOR Proof Tokens is important when considering how they are structured. These sections detail the underlying reasoning informing their design.

4.1. Unlinkability

Supporting unlinkability is perhaps the most challenging design constraint for JPTs and CPTs. Even the smallest oversight can introduce a subtle vector for relying parties to collude and correlate one or more subjects across their usage.

The principal tools to prevent this are data minimization and uniformity. The data included SHOULD be minimized to remove potential correlation points. The data SHOULD contain only values that are able to be selectively disclosed with consent or transformed by the proof algorithm when presented.

Any other data that is repeated across multiple JPTs or CPTs is externalized so that it is uniform across every issuance. This includes preventing the usage of optional Headers, dynamic mapping of claims to payloads, changes to how many payloads are included, and the ordering of the payloads.

4.2. Selective Disclosure

While JWPs provide the underling structure for easily supporting selective disclosure, JPTs and CPTs must go a step further to ensure that holders can effectively provide choice and consent on exactly what is being disclosed. Software using JWPs or CPTs MUST know the mappings from payloads to claims. All disclosed payloads MUST be mapped to claims and made accessible to the application. Holders SHOULD understand the semantics of all potentially disclosed claims to the extent needed to decide whether to disclose them. JPTs and CPTs SHOULD NOT contain claims that are intended only for a specific verifier.

4.3. Familiarity

JPTs are intended to be as close to a JWT as possible in order to provide the simplest transition for any JWT-based system to add support for JPTs. The same is true for CPTs and CWTs.

Although there are some stark differences in the lifecycle of a JPT, from the application’s perspective, the interface to a JPT can be made fairly similar: a JSON object containing a mix of required and optional claims with well-understood values. Likewise, A CPT is a CBOR object containing a mix of required and optional claims with well-understood values.

The most significant divergence required by JPTs and CPTs is that of supporting values that may be disclosed or may instead only be a proof about the value. Applications are required to interact with

the JPT or CPT on a payload-by-payload basis instead of just verifying a JWT or CWT and then being able to interact with the JSON or CBOR body directly.

4.4. Proofs

To generate a variety of efficient ZKPs of knowledge, range, membership, or other predicates, it is essential that each individual payload is only a single claim value. This greatly simplifies the task of linking a derived proof of a given claim to the specific payload that was also signed by the issuer. While JPTs and CPTs support claims that have complex object or array compound values, they also allow for simple claim values such as strings, numbers, and booleans that can be used directly in generating predicate proofs.

5. Claim Names

It is RECOMMENDED that the claim names used with JPTs come from those in the IANA JSON Web Token Claims Registry [IANA.JWT] established by [RFC7519], when those fit the application's needs. Likewise, it is RECOMMENDED that the claim names used with CPTs come from those in the IANA CBOR Web Token Claims Registry [IANA.CWT] established by [RFC8392], when those fit the application's needs.

6. Claims Header Parameter

A JSON Proof Token or CBOR Proof Token assigns each payload a claim name. Payloads MUST each have a negotiated and understood claim name within the application context. The simplest solution to establish payload claim names is as an ordered array that aligns with the ordering of payload slots. This claims array can be conveniently included in the Claims Header Parameter.

The claims Header Parameter is an array listing the Claim Names corresponding to the JWP payload slots, in the same order as the payload slots. Each array value is a Claim Name, as defined in [RFC7519] or [RFC8392]. Use of this Header Parameter is OPTIONAL.

All JPT payloads that are claim values MUST be the base64url encoding of the UTF-8 representation of a JSON value. That said, predicate proofs derived from payload values are not represented as claims; they are contained in the presentation proof using algorithm-specific representations.

All CPT payloads that are claim values MUST be a CBOR value. Likewise, CPT predicate proofs derived from payload values are not represented as claims; they are contained in the presentation proof using algorithm-specific representations.

The following is an example Issuer Header that includes a claims property:

```
{
  "kid": "HjfcpyjuZQ-O8Ye2hQnNbT9RbbnrobptdnExR0DUjU8",
  "alg": "BBS",
  "claims": [
    "iat",
    "exp",
    "family_name",
    "given_name",
    "email",
    "address",
    "age_over_21"
  ]
}
```

In this example, the "iat" and "exp" would be JSON-formatted numbers, "family_name", "given_name" and "email" would be JSON strings (in quotes), "address" would be a JSON object, and "age_over_21" would be either true or false.

7. Claims ID ("cid") Header Parameter

A Claims ID ("cid") value can be used as an identifier for a set of claim names without explicitly listing them. Its use is similar to the Key ID ("kid") Header Parameter.

The structure of the cid value is unspecified. For JPTs, its value MUST be a case-sensitive string. For CPTs, its value MUST be a binary string. Use of this Header Parameter is OPTIONAL.

The cid can be used similarly to a kid in order to ensure that it is possible to externally resolve and then verify that the correct list of claim names is being used when processing the payloads containing the claim values.

If there is an associated JWK containing the signing key information, the claims key is also registered there as a convenient location for the claim names. Likewise, if there is an associated COSE_Key containing the signing key information, the claims key is also registered there as a convenient location for the claim names.

When the claims array is transferred as a property in the Issuer Header, any variations of that array between JWP will be visible to the verifier, and can leak information about the subject or provide an additional vector for linkability. Given the privacy design considerations around linkability, it is RECOMMENDED that the claims are defined externally to an individual JPT or CPT and either referenced or known by the application context.

The following is an example Header that includes a cid:

```
{
  "kid": "HjfcpyjuZQ-O8Ye2hQnNbT9RbbnrobptdnExR0DUjU8",
  "alg": "BBS",
  "cid": "guA8PAI14Gkn4273f1rR606yMbRMFg4y"
}
```

8. Presented Claims and Proofs

Each claim in the issued form of the JPT or CPT results in one of three things in the presented form of the JPT or CPT:

1. A disclosed JSON or CBOR value.
2. An indicator that the value was not disclosed.
3. An algorithm-specific proof method.

8.1. Disclosed

A disclosed payload of a JPT is represented as a UTF8-encoded octet string representing a valid JSON value. A disclosed payload of a CPT is represented as a CBOR value.

8.2. Undisclosed

The placeholder indicating that a payload was not disclosed is represented as described in Section 6 (Serializations).

9. Example JPT and CPT

See the examples in Appendix A.1 of [I-D.ietf-jose-json-proof-algorithms].

10. Security Considerations

* Header Minimization

11. IANA Considerations

11.1. JSON Web Proof Header Parameters Registration

This section registers the following Header Parameter in the IANA "JSON Web Proof Header Parameters" registry established by [I-D.ietf-jose-json-web-proof].

11.1.1. Registry Contents

11.1.1.1. "claims" (Claims) Header Parameter

* Header Parameter Name: Claims
* Header Parameter JSON Label: claims
* Header Parameter CBOR Label: 10
* Header Parameter Usage Location(s): Issued
* Change Controller: IETF
* Specification Document(s): Section 6 of this specification

11.1.1.2. "cid" (Claims ID) Header Parameter

* Header Parameter Name: Claims ID
* Header Parameter JSON Label: cid
* Header Parameter CBOR Label: 11
* Header Parameter Usage Location(s): Issued
* Change Controller: IETF
* Specification Document(s): Section 7 of this specification

11.2. JSON Web Key Parameters Registry

This section registers the following JWK parameter in the IANA "JSON Web Key Parameters" registry [IANA.JOSE] established by [RFC7517].

11.2.1. Registry Contents

* Parameter Name: claims
* Parameter Description: Array of claim names
* Used with "kty" Value(s): *
* Parameter Information Class: Public
* Change Controller: IESG
* Specification Document(s): Section 7 of this specification

11.3. COSE Key Common Parameters Registry

This section registers the following COSE_Key parameter in the IANA "COSE Key Common Parameters" registry [IANA.COSE] established by [RFC8152].

11.3.1. Registry Contents

* Name: claims
* Label: TBD (requested assignment 6)
* CBOR Type: array
* Value Registry: CBOR Web Token Claims
* Description: Array of claim names
* Reference: Section 7 of this specification

11.4. Media Types Registry

This section registers the following media type [RFC2046] in the IANA "Media Types" registry [IANA.MediaTypes] in the manner described in [RFC6838].

11.4.1. application/jpt

The media type for a JSON Proof Token (JPT) is application/jpt.

* Type name: application
* Subtype name: jpt
* Required parameters: n/a
* Optional parameters: n/a
* Encoding considerations: 8bit; JPT values are encoded as a series of base64url-encoded values (some of which may be the empty string) separated by period ('.') characters.
* Security considerations: See Section 10 of this specification
* Interoperability considerations: n/a
* Published specification: This specification
* Applications that use this media type: Applications releasing claims with zero-knowledge proofs
* Additional information:
  - Magic number(s): n/a
  - File extension(s): n/a
  - Macintosh file type code(s): n/a
* Person & email address to contact for further information: Michael B. Jones, michael_b_jones@hotmail.com
* Intended usage: COMMON
* Restrictions on usage: none
* Author: Michael B. Jones, michael_b_jones@hotmail.com
* Change controller: IETF
* Provisional registration: No

11.4.2. application/cpt

The media type for a CBOR Proof Token (CPT) is application/cpt.

* Type name: application

* Subtype name: cpt
* Required parameters: n/a
* Optional parameters: n/a
* Encoding considerations: 8bit; CPT values are encoded as CBOR
* Security considerations: See Section 10 of this specification
* Interoperability considerations: n/a
* Published specification: This specification
* Applications that use this media type: Applications releasing claims with zero-knowledge proofs
* Additional information:
  - Magic number(s): n/a
  - File extension(s): n/a
  - Macintosh file type code(s): n/a
* Person & email address to contact for further information: Michael B. Jones, michael_b_jones@hotmail.com
* Intended usage: COMMON
* Restrictions on usage: none
* Author: Michael B. Jones, michael_b_jones@hotmail.com
* Change controller: IETF
* Provisional registration: No

11.5. Structured Syntax Suffix Registry

This section registers the following entries in the IANA "Structured Syntax Suffix" registry [IANA.StructuredSuffix] in the manner described in [RFC6838].

11.5.1. +jpt

* Name: JSON Proof Token (JPT)
* +suffix: +jpt
* References: This specification
* Encoding considerations: 8bit; JPT values are encoded as a series of base64url-encoded values (some of which may be the empty string) separated by period ('.') characters.
* Interoperability considerations: n/a
* Fragment identifier considerations: n/a
* Security considerations: See Section 10 of this specification
* Contact: Michael B. Jones, michael_b_jones@hotmail.com
* Author/Change controller: IETF

11.5.2. +cpt

* Name: CBOR Proof Token (CPT)
* +suffix: +cpt
* References: This specification
* Encoding considerations: 8bit; CPT values are encoded as CBOR
* Interoperability considerations: n/a

* Fragment identifier considerations: n/a
* Security considerations: See Section 10 of this specification
* Contact: Michael B. Jones, michael_b_jones@hotmail.com
* Author/Change controller: IETF

11.6. CoAP Content-Formats Registry

This section registers the following CoAP Content-Formats value in the [IANA.CoAP.Formats] registry.

11.6.1. "CPT" CoAP Content-Format

The CoAP Content-Format for a CBOR Proof Token (CPT) is as follows.

* Content Type: application/cpt
* ID: TBD (requested assignment 20)
* Reference: Section 11.4.2 of this specification

12. References

12.1. Normative References

[I-D.ietf-jose-json-web-proof]
Waite, D., Jones, M. B., and J. Miller, "JSON Web Proof", Work in Progress, Internet-Draft, draft-ietf-jose-json-web-proof-latest, <https://datatracker.ietf.org/doc/html/draft-ietf-jose-json-web-proof>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8392] Jones, M., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "CBOR Web Token (CWT)", RFC 8392, DOI 10.17487/RFC8392, May 2018, <https://www.rfc-editor.org/info/rfc8392>.

12.2. Informative References

[I-D.ietf-jose-json-proof-algorithms]
Jones, M. B., Waite, D., and J. Miller, "JSON Proof Algorithms", Work in Progress, Internet-Draft, draft-ietf-jose-json-proof-algorithms-latest, <https://datatracker.ietf.org/doc/html/draft-ietf-jose-json-proof-algorithms>.

[IANA.COSE]
IANA, "CBOR Object Signing and Encryption", <https://www.iana.org/assignments/cose>.

[IANA.CWT] IANA, "CBOR Web Token", <https://www.iana.org/assignments/cwt>.

[IANA.CoAP.Formats]
IANA, "CoAP Content-Formats", <https://www.iana.org/assignments/core-parameters/core-parameters.xhtml#content-formats>.

[IANA.JOSE]
IANA, "JSON Object Signing and Encryption", <https://www.iana.org/assignments/jose>.

[IANA.JWT] IANA, "JSON Web Token", <https://www.iana.org/assignments/jwt>.

[IANA.MediaTypes]
IANA, "Media Types", <https://www.iana.org/assignments/media-types>.

[RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, DOI 10.17487/RFC2046, November 1996, <https://www.rfc-editor.org/info/rfc2046>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/info/rfc6838>.

[RFC7517] Jones, M., "JSON Web Key (JWK)", RFC 7517, DOI 10.17487/RFC7517, May 2015, <https://www.rfc-editor.org/info/rfc7517>.

[RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", RFC 8152, DOI 10.17487/RFC8152, July 2017, <https://www.rfc-editor.org/info/rfc8152>.

Appendix A. Acknowledgements

This work was incubated in the DIF Applied Cryptography Working Group (https://identity.foundation/working-groups/crypto.html).

We would like to thank Brent Zundel for his valuable contributions to this specification.

Appendix B. Document History

[[To be removed from the final specification]]

-12

* IANA Considerations section changes from IANA Early Review

-11

* Change Issuer Protected Header to Protected Header
* Remove JWP qualifiers on Header and Protected Header

-10

* Registered +jpt and +cpt structured syntax suffixes.
* Clarify mapping of the claims array to payload data using "payload slot" nomenclature
* Move proof methods text to JWP.

-09

* No changes

-08

* Defined CBOR Proof Token (CPT).
* Registered application/jpt and application/cpt media types and CPT CoAP Content-Format.
* Made some additional references normative.

-07

* Changing primary editor
* Move claims definition from JWP, to live beside cid
* Update cid registry entry to assign CBOR label

-06

* Update reference to new repository home

* Fixed #99: Discussed issued and presented forms of JPTs.

-05

* Define and register Claims ID JWP Header Parameter.

-04

* Refactoring figures and examples to be built from a common set across all three documents

-03

* Improvements resulting from a full proofreading.
* Added examples of JSON object and JSON boolean claims.

-02

* Update example to use the current BBS algorithm

-01

* Correct cross-references within group.

-00

* Created initial working group draft based on draft-jmiller-jose-json-proof-token-01

Authors' Addresses

Michael B. Jones
Self-Issued Consulting
Email: michael_b_jones@hotmail.com
URI: https://self-issued.info/

David Waite
Ping Identity
Email: dwaite+jwp@pingidentity.com

Jeremie Miller
Ping Identity
Email: jmiller@pingidentity.com