JSON Web Proof

Mike Jones, David Waite

IETF 124, Montreal
November 5, 2025

Document Updates Since Madrid

-11

- Editorial changes
- Updated some text on header usage
- Use consistent terminology (Issuer Header, Presentation Header), eliminating “Protected Header”.

-12

- Reflect IANA early review
- Update to new BLS key representation (compressed/ZCash style)

Editorial: CBOR Examples

There is an appendix with a single CBOR example today

Instead, we would like parity with JSON (as much as possible)

The task is to create a set of semantically similar JSON and CBOR examples to support the algorithm definitions, and to rely on them for examples in rest of the document.

The hope is that this exercise will uncover issues in the spec (it already has)

In progress: Registry Changes

While working to unify JSON and CBOR examples, we encountered an issue:

While JWP creates its own header registry and algorithms registry, JWK and CWK do not know about them.

1. Evaluating the impact of leveraging “JSON Web Signature and Encryption Algorithms” and “COSE Algorithms” registries.
2. Alternatively, could change how we restrict key usage to not have a conflicting parameter

In Progress: JPT Subclaim Support

JSON Proof Tokens (JPT) constrains *JWP* payloads down to claims, targeting the same problem domains as SD-JWT and SD-CWT with newer algorithms.

The expectation is that additional work will map JPT to application domains, similar to *SD-JWT-based Verifiable Credentials*

Problem Statement

JPT and SD-JWT use different approaches.

- SD-JWT works as a preprocessor that re-composes a JSON document based on a subset of hash-identified disclosures in the JSON structure
- JPT provides keyed values, which are disclosed per the underlying presentation algorithm

Despite trade-offs of the different approaches, we do not want to constrain the ability to solve application domains with JPT

Feedback from Digital Credential Prototyping

Two independent prototypes using JPT for digital credentials had a common feedback: mapping payloads to entire claims was too limiting.

1. An `address` claim may lead to disclosing a full location, when a verifier only needs a `country` or `postal_code`
2. A full array of `affiliations` or `nationalities` might be disclosed when otherwise just one relevant value was appropriate
3. A `birthdate` may need to be disclosed when all that is really needed is a predicate - that a person is over a certain age

Status today

Since JPT today only provides for release of entire claims, these subsets and predicates must be codified into new claims to reduce information disclosed:

- `address_postal_code`
- `nationality_1`
- `age_over_25`

SD-JWT provides facilities for the first two without creating (and registering) new claims, so there is motivation to provide some level of additional affordances to JPT.

Proposal

Define a syntax to express that a payload holds a subset of a claim (a subclaim) as a JSON value or CBOR data item.

Ideally the syntax would:

1. Have the same semantics for both JPT and CPT (with JPT utilizing a subset)
2. Would be simple enough to avoid concerns about third party expressions
3. Avoid corner cases which are unexpectedly computationally/memory expensive
4. Align with work already within the IETF
5. Align with work in application domains we envision JPT being applied to

Evaluation of existing work

Spec	**IETF Status**	**Remote-safe (Q)**	**Simple (Q)**	**Cheap (Q)**	**CBOR-ready**
JSON Pointer	RFC				
JSON Path	RFC				
SD-JWT VC (metadata path)	WG I-D			[1]	
OID4VP (DCQL query)				[1]	
CBOR Pointer	Ind I-D				

** See slide notes*

Current Proposal

Largely based on CBOR Pointer

- No negative indexing of arrays
- No indexing into serialized CBOR

Comparison to DQCL and SD-JWT VC syntax (for JSON)

- Removed the one query-based addition (null-indexed arrays)

Example of Usage (modified from SD-JWT VC)

```
{
    "given_name": "John",
    "family_name": "Doe",
    "email": "johndoe@example.com",
    "phone_number": "+1-202-555-0101",
    "address": {
      "street_address": "123 Main St",
      "locality": "Anytown",
      "region": "Anystate",
      "country": "US"
    },
    "birthdate": "1940-01-01",
    "is_over_18": true
  }
```

Example of JPT Usage (today)

"given_name"	"John"
"family_name"	"Doe"
"email"	"johndoe@example.com"
"phone_number"	"+1-202-555-0101"

"address"	{
	   "street_address": "123 Main St",
	   "locality": "Anytown",
	   "region": "Anystate",
	   "country": "US"
	}

"birthdate"	"1940-01-01"
"is_over_18"	true

Subclaim Proposal

"given_name"	"John"
"family_name"	"Doe"
"email"	"johndoe@example.com"
"phone_number"	"+1-202-555-0101"
"address"	{ "street_address": "123 Main St", "locality": "Anytown", "region": "Anystate", "country": "US" }
["address","country"]	**"US"**
"birthdate"	"1940-01-01"
"is_over_18"	true

Open Questions

1. Does this seem like the proper approach?

2. How important would it be to reference a syntax defined elsewhere, vs. creating another derivative?

3. Do we define how to construct a document from subclaims?
    - Somewhat easy for JSON, harder for CBOR