

JSON Web Proofs Specifications

IETF 116 Yokohama, March 26, 2023

Jeremie Miller, Mike Jones, and David Waite

IETF History

JSON Web Proofs Specifications

IETF History

- Discussed during JSON Web Proofs (JWP) BoF at IETF 114 Philadelphia in July 2022
 - Existence proof of JSON-based container format and algorithms for Zero-Knowledge Proofs (ZKPs)
- Discussed more during virtual interim JWP BoF in October 2022
- Helped motivate reanimation of JOSE working group
 - Where we are today!

Overview of Specifications

JSON Web Proofs Specifications

What are they?

- Propose new container syntax, in the spirit of JOSE's JWS and JWE
- Goal is to support algorithms and cryptographic techniques for newer privacy-preserving applications such as "anonymous credentials" use cases and Zero-Knowledge Proofs (ZKPs)
- Native support for multiple payloads
- Enable transformations of a secured messages, both the payloads and integrity values, without compromising their integrity or verifiability

JSON Web Proofs Specifications

What are they?

Examples of capabilities algorithms may support include:

- Selectively disclose a subset of payloads to a verifier
- Multiple presentations of a container using uncorrelatable integrity values
- Prove a predicate without disclosing the payload values used for evaluation
- Proofs of Knowledge

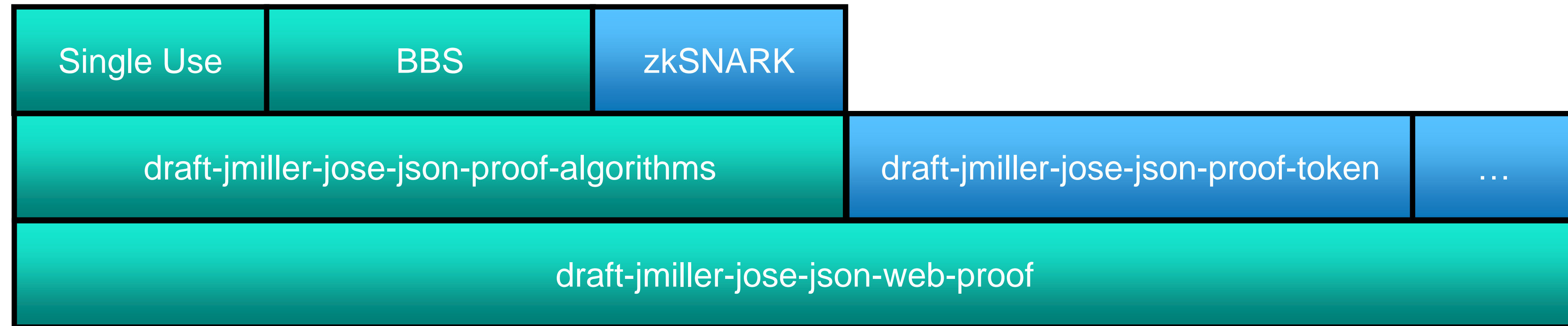
JSON Web Proofs Specifications

More History

- Early-Mid 2021 - Initial ideas circulated in the OpenID Connect SIOP community by Jeremie Miller and David Waite, incubated at the DIF in the Applied Crypto WG
- Late 2021 / Early 2022 - Initial -00 drafts with much guidance and input from Mike Jones
- Mid 2022 - Interest in use of JWP included in W3C Verifiable Credentials 2.0 Charter
- Mid 2022 - IETF BoFs where we proposed rechartering the JOSE WG to take forward the initial work
- Late 2022 - IETF JOSE WG reanimated
 - Many thanks to Roman Danyliw, Karen O'Donoghue, and John Bradley!

JSON Web Proofs Specifications

A guided tour



- JSON Web Proof – Analogous to JSON Web Signature (JWS) – RFC 7515
- JSON Proof Algorithms – Analogous to JSON Web Algorithms (JWA) – RFC 7518
- JSON Proof Token – Analogous to JSON Web Token (JWT) – RFC 7519

JWP Design Factors

Three Interrelated Privacy Features

Selective Disclosure, Unlinkability, and Proofs of Knowledge

- **Selective Disclosure** - revealing only a subset of that message while maintaining its verifiability
 - The message is separated into distinct disclosable payloads
 - The integrity-protected message can disclose subsets of the payloads
- **Unlinkability** - ensuring the integrity protection does not inherently enable correlation between verifiers when the same message is presented multiple times
 - Allows the presenter to generate new unique integrity protection values that still verify
- **Proofs of Knowledge** - keeping payloads private while still proving knowledge of or about them
 - Minimizes the information a verifier requires to only what is needed

KISS

Advanced crypto is already hard enough

- Strive to align with “*What would JOSE do?*”
- Strive to “Keep simple things simple”
- Core JWP draft is minimal container formatting only
- Explores techniques that are adoptable today (MAC-based)
- Support new signature types with necessary capabilities (BBS, PS Signatures)
- Remain flexible to support more advanced crypto as it evolves (DL-PoK, ZKPs, Mercurial, predicates, verifiable compute, etc.)

Comparison of JWP and JWS

Classic JSON Web Signature

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
.

Classic JSON Web Signature

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdBWliOilxMiM0NTY3ODkwliwibmFtZSI6IkpvaG4gR091dC50aWwuanVzIj09.
SfIKxwRJSMeKKE2OT4fwpMeJf36POk6yJV_adQssw5c

Protected Header

Payload

Signature

JSON Web Proof

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
.SI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2M
~JhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9e
y.SfIKxwRJSMeKKF2QT4fwpMeJf36POk
6yJV_adQssw5c

JSON Web Proof

The diagram illustrates the structure of a zk-SNARK transaction, which is composed of three main parts:

- Protected Header:** The top section, highlighted in red, containing the transaction ID and other metadata. Example text: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9`.
- Payloads:** The middle section, highlighted in green, containing the transaction data. Example text: `.eyJzdWliOiIxMjM0NTY3ODkwIiwibmFt`.
- Proof:** The bottom section, highlighted in blue, containing the zero-knowledge proof. Example text: `y.SfIKxwRJSMeKKF2QT4fwpMeJf36POk`.

The transaction is represented as a sequence of these three parts, with the Protected Header and Proof sections being highlighted in red and blue respectively, and the Payloads section being highlighted in green.

JSON Web Proof Presentation

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZ
IjoiSfIKxwRJSMeKKF2QT4fwpMeJf36PO
k6vJV_adQssw5c

Two Omitted Payloads

JWP Specification Links

- JSON Web Proof
 - <https://www.ietf.org/archive/id/draft-jmiller-jose-json-web-proof-01.html>
- JSON Proof Algorithms
 - <https://www.ietf.org/archive/id/draft-jmiller-jose-json-proof-algorithms-01.html>
- JSON Proof Token
 - <https://www.ietf.org/archive/id/draft-jmiller-jose-json-proof-token-01.html>

Next Steps

Working Group Adoption of JWP Specs?

- Could provide a basis for the reanimated working group to begin its new work
- A starting point from which we can iteratively improve and evolve
- As IETF working groups do!

The Question Before Us...

