Workgroup: Web Authorization Protocol
Internet-Draft:
draft-fett-selective-disclosure-jwt-00
Published: 1 June 2022
Intended Status: Standards Track
Expires: 3 December 2022
Authors: D. Fett K. Yasuda
yes.com Microsoft

Selective Disclosure JWT (SD-JWT)

Abstract

This document specifies conventions for creating JSON Web Token (JWT) documents that support selective disclosure of JWT claim values.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 3 December 2022.

Copyright Notice

Table of Contents

1. Introduction
  1.1. Conventions and Terminology
2. Terms and Definitions
  2.1. Selective Disclosure JWT (SD-JWT)
  2.2. SD-JWT Salt/Value Container (SVC)
  2.3. SD-JWT Release (SD-JWT-R)
  2.4. Holder binding
  2.5. issuer
  2.6. holder
  2.7. verifier
3. Concepts
  3.1. Creating an SD-JWT
  3.2. Creating an SD-JWT Release
  3.3. Verifying an SD-JWWT Release
4. Data Formats
  4.1. SD-JWT Format
    4.1.1. Payload
    4.1.2. Example 1 - Flat SD-JWT
    4.1.3. Example 2 - Structured SD-JWT
  4.2. SD-JWT Salt/Value Container (SVC)
    4.2.1. Payload
    4.2.2. Example 1 - SVC for a Flat SD-JWT
    4.2.3. Example 2 - SVC for a Structured SD-JWT
  4.3. SD-JWT and SVC Combined Format
  4.4. SD-JWT-R Format
  4.5. Presentation Format
5. Verification
6. Security Considerations
  6.1. Holder Binding
7. Privacy Considerations
  7.1. Claim Names
  7.2. Unlinkability
8. Acknowledgements
9. IANA Considerations
10. Normative References
11. Informative References
Appendix A. Additional Examples
  A.1. Example 3 - Complex Structured SD-JWT
  A.2. Example 4 - W3C VC
Appendix B. Document History
Authors' Addresses

1. Introduction

The JSON-based claims in a signed JSON Web Token (JWT) [RFC7519] document are secured against modification using JSON Web Signature (JWS) [RFC7515] digital signatures. A consumer of a signed JWT

document that has checked the document's signature can safely assume that the contents of the document have not been modified. However, anyone receiving an unencrypted JWT can read all of the claims and likewise, anyone with the decryption key receiving an encrypted JWT can also read all of the claims.

This document describes a format for JWTs that support selective disclosure (SD-JWT), enabling sharing only a subset of the claims included in the original JWT instead of releasing all the claims to every verifier. This document also defines a format for so-called SD-JWT Releases (SD-JWT-R).

One of the common use cases of a signed JWT is representing a user's identity created by an issuer. In such a use case, there has been no privacy-related concerns with existing JOSE signature schemes, because when a signed JWT is one-time use, it contains only JWT claims that the user has consented in real time to release to the verifier. However, when a signed JWT is intended to be multi-use, the ability to selectively disclose a subset of the claims depending on the verifier becomes crucial to ensure minimum disclosure and prevent verifier from obtaining claims irrelevant for the use case at hand.

One example of such a multi-use JWT is a verifiable credential, or a tamper-evident credential with a cryptographically verifiable authorship that contains claims about a subject. SD-JWTs defined in this document enable such selective disclosure of claims.

While JWTs for claims describing natural persons are a common use case, the mechanisms defined in this document can be used for many other use cases as well.

Note: so far agreed to define holder binding (user's public key contained inside an SD-JWT) as an option. It is not mandatory since holder binding is use case specific and orthogonal to the general mechanism of selective disclosure we are trying to define here.

1.1. Conventions and Terminology

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.

base64url denotes the URL-safe base64 encoding without padding defined in Section 2 of [RFC7515].

2. Terms and Definitions

2.1. Selective Disclosure JWT (SD-JWT)

A JWT [RFC7515] created by the issuer, which can be signed as a JWS [RFC7515], that supports selective disclosure as defined in this document.

2.2. SD-JWT Salt/Value Container (SVC)

A JSON object created by the issuer that contains mapping between raw claim values that contained in the SD-JWT and the salts for each claim value.

2.3. SD-JWT Release (SD-JWT-R)

A JWT created by the holder that contains a subset of the claim values of an SD-JWT in a verifiable way.

2.4. Holder binding

Ability of the holder to prove legitimate possession of SD-JWT by proving control over the same private key during the issuance and presentation. SD-JWT signed by the issuer contains a public key or a reference to a public key that matches to the private key controlled by the holder.

2.5. issuer

An entity that creates SD-JWTs (2.1).

2.6. holder

An entity that received SD-JWTs (2.1) from the issuer and has control over them.

2.7. verifier

An entity that entity that requests, checks and extracts the claims from SSD-JWT-R (2.2)

Note: discuss if we want to include Client, Authorization Server for the purpose of ensuring continuity and separating the entity from the actor.

3. Concepts

In the following section, the concepts of SD-JWTs and SD-JWT Releases are described at a conceptual level.

3.1. Creating an SD-JWT

An SD-JWT, at its core, is a digitally signed document containing hashes over the claim values with unique salts, optionally the holder's public key or a reference thereto and other metadata. It is digitally signed using the issuer's private key.

```
SD-JWT-DOC = (METADATA, HOLDER-PUBLIC-KEY?, HS-CLAIMS)
SD-JWT = SD-JWT-DOC | SIG(SD-JWT-DOC, ISSUER-PRIV-KEY)
```

HS-CLAIMS is usually a simple object with claim names mapped to hashes over the claim values with unique salts:

```
HS-CLAIMS = (
    CLAIM-NAME: HASH(SALT | CLAIM-VALUE)
)*
```

HS-CLAIMS can also be nested deeper to capture more complex objects, as will be shown later.

The SD-JWT is sent from the issuer to the holder, together with the mapping of the plain-text claim values, the salt values, and potentially some other information.

3.2. Creating an SD-JWT Release

To disclose to a verifier a subset of the SD-JWT claim values, a holder creates a JWS such as the following:

```
RELEASE-DOC = (METADATA, SALTS)
RELEASE = RELEASE-DOC | SIG(RELEASE-DOC, HOLDER-PRIV-KEY)?
```

Note that the signature over RELEASE-DOC is optional and required if, and only if, holder binding is desired.

SALTS is usually a simple object with claim names mapped to values and salts:

```
SALTS = (
    CLAIM-NAME: (DISCLOSED-SALT, DISCLOSED-VALUE)
)
```

Just as HS-CLAIMS, SALTS can be more complex as well.

The SD-JWT-R is sent together with the SD-JWT from the holder to the verifier.

3.3. Verifying an SD-JWWT Release

A verifier checks that

- if holder binding is desired, the RELEASE was signed by the private key belonging to the public key contained in SD-JWT-DOC.

- for each claim in RELEASE, the hash HASH(DISCLOSED-SALT | DISCLOSED-VALUE) matches the hash under the given claim name in the SD-JWT.

The detailed algorithm is described below.

4. Data Formats

This section defines a data format for SD-JWTs (containing hashes of the salted claim values) and for SD-JWT Salt/Value Containers (containing the mapping of the plain-text claim values and the salt values).

4.1. SD-JWT Format

An SD-JWT is a JWT that is optionally signed using the issuer's private key.

4.1.1. Payload

The payload of an SD-JWT can consist of the following claims.

4.1.1.1. Selectively Disclosable Claims

An SD-JWT MUST include hashes of the salted claim values that are included by the issuer under the property _sd.

The issuer MUST choose a unique salt value for each claim value. Each salt value MUST contain at least 128 bits of pseudorandom data, making it hard for an attacker to guess. The salt value MUST then be encoded as a string. It is RECOMMENDED to base64url encode at least 16 pseudorandom bytes.

The issuer MUST build the hashes by hashing over a string that is formed by JSON-encoding an ordered array containing the salt and the claim value, e.g.: ["6qMQvRL5haj","Peter"]. The hash value is then base64url-encoded. Note that the precise JSON encoding can vary, and therefore, the JSON encodings MUST be sent to the holder along with the SD-JWT, as described below.

The _sd object can be a 'flat' object, directly containing all claim names and hashed claim values without any deeper structure. The _sd object can also be a 'structured' object, where some claims and

their respective hashes are contained in places deeper in the structure. It is up to the issuer to decide how to structure the representation such that it is suitable for the use case. Examples 1 and 2 below show this using the [OIDC] address claim, a structured claim. Appendix 1 shows a more complex example using claims from eKYC (todo: reference).

Note that it is at the issuer's discretion whether to turn the payload of SD-JWT into a 'flat' or 'structured' _sd SD-JWT object.

4.1.1.2. Holder Public Key

If the issuer wants to enable holder binding, it includes a public key associated with the holder, or a reference thereto.

It is out of the scope of this document to describe how the holder key pair is established. For example, the holder MAY provide a key pair to the issuer, the issuer MAY create the key pair for the holder, or holder and issuer MAY use pre-established key material.

Note: need to define how holder public key is included, right now examples are using sub_jwk I think.

4.1.1.3. Other Claims

The SD-JWT payload MAY contain other claims and will typically contain other JWT claims, such as iss, iat, etc.

4.1.2. Example 1 - Flat SD-JWT

This example shows a simple SD-JWT containing user claims. The issuer here decided to use a completely flat structure, i.e., the address claim can only be disclosed in full.

In this example, these claims are the payload of the SD-JWT:

```
{
  "sub": "6c5c0a49-b589-431d-bae7-219122a9ec2c",
  "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"
}
```

The following shows the resulting SD-JWT payload:


```
{
  "iss": "https://example.com/issuer",
  "sub_jwk": {
    "kty": "RSA",
    "n": "6GwTTwcjVyOtKtuGf7ft5PAU0GiDtnD4DGcmtVrFQHVhtx05-DJigfmR-3Tetw-Od5su4TNZYzjh3tQ6Bj1
    "e": "AQAB"
  },
  "iat": 1516239022,
  "exp": 1516247022,
  "_sd": {
    "sub": "LbnhkOr5oS7KjeUrxezAu8TG0CpWz0jSixy6tffuo04",
    "given_name": "fUMdn88aaoyKTHrvZd6AuLmPraGhPJ0zF5r_JhxCVZs",
    "family_name": "9h5vgv6TpFV6GmnPtugiMLl5tHetHeb5X_2cKHjN7cw",
    "email": "fPZ92dtYMCN2Nb-2ac_zSH19p4yakUXrZl_-wSgaazA",
    "phone_number": "QdSffzNzzd0n60MsSmuiKj6Y6Enk2b-BS-KtEePde5M",
    "address": "JFu99NUXPq55f6DFBZ22rMkxMNHayCrfPG0FDsqbyDs",
    "birthdate": "Ia1Tc6_Xnt5CJc2LtKcu6Wvqr42glBGGcjGOye8Zf3U"
  }
}
```


The SD-JWT is then signed by the issuer to create a document like the following:

```
eyJhbGciOiAiUlMyNTYifQ.eyJpc3MiOiAiaHR0cHM6Ly9leGFtcGxlLmNvbS9pc3N1ZXI
iLCAic3ViX2p3ayI6IHsia3R5IjogIlJTQSIsICJuIjogInRYMnhjV3ZqQWtMbDV1TjVnY
VREZURWRnBldk9jTmpqUE0tcnlDV0tqYlVvaDVueURFYUZwcXZ2M1dEZUd3V2VCak1YZWQ
0aHRvdlItM2ZKeVc1UzBkX3RVV2NkRW1TR2VxcmNVbmI4QzcxVlVfcENRWUQ2TUg5WWpSQ
210M1JqbnVGZUhaaUp5bWJnVDhoN2NWTDBpdlRYN3FMOWZ0TUNONUpSb05ZRjVETUdCWGx
2Q2dMQ0dJYmRLNy10a1B5aGRlR2dzRkphX2FNNGZ0NVNSU3NxenJQNzZ2eXY1NVZ4UE1tR
G9HSnJBSm4zOEhsa3FvSUVYaGpqTWgwRUFDdUM4SFhNOVh3b3Z0T3FlWEFPNEdnSndLRXd
XZ3lwYTVRd3FSZUhvUUljVVlxWVRKQ3dpa3J6a2h2dElscUlka1NSS1VTWC1BOWdqdFFfR
S1neXQ5M05HUSIsICJlIjogIkFRQUIifSwgImlhdCI6IDE1MTYyMzkwMjIsICJleHAiOiA
xNTE2MjQ3MDIyLCAiX3NkIjogeyJzdWIiOiAiTGJuaGtPcjVvUzdLamVVcnhlekF1OFRHM
ENwV3owalNpeHk2dGZmdW8wNCIsICJnaXZlbl9uYW1lIjogImZVTWRuODhhYW95S1RIcnZ
aZDZBdUxtUHJhR2hQSjB6RjVyX0poeENWWnMiLCAiZmFtaWx5X25hbWUiOiAiOWg1dmd2N
lRwRlY2R21uUHR1Z2lNTGw1dEhldEhlYjVYXzJjS0hqTjdjdyIsICJlbWFpbCI6ICJmUFo
5MmR0WU1DTjJOYi0yYWNfelNIMTlwNHlha1VYclpsXy13U2dhYXpBIiwgInBob25lX251b
WJlciI6ICJRZFNmZnpOenpkMG42ME1zU211aUtqNlk2RW5rMmItQlMtS3RFZVBkZTVNIiw
gImFkZHJlc3MiOiAiSkZ1OTlOVVhQcTU1ZjZERkJaMjJyTWt4TU5IYXlDcmZQRzBGRHNxY
nlEcyIsICJiaXJ0aGRhdGUiOiAiSWExVGM2X1hudDVDSmMyTHRLY3U2V3ZxcjQyZ2xCR0d
jakdPeWU4WmYzVSJ9fQ.IS4oc1f3XuxhNSnecIXbpT-3ZVwgbjpMSfpyqhFUEE2T_ij3uW
Bqb1_zn0nLvIvXDs8rn6l10ilHwCgpMaPmYAE8_nfZtNwvfAFnwBFjzdrJOJWhZ5dp6UJe
VULOZvjsCw1EpyRbBgIyZ9QiLzRJ_5JS1C1AelDDyXxI3FZYYc3-1MqQMnaXR7AWOct698
t-LsookAA_LxXx-RYKG1wygEp9e9BzgCxQugsdGejMPTZyfaQewGrJalQm8bYvSXKcJ1DG
-T297kFEV_VTaeLCOoan1DS1DtaH48Q13yUUmdwil8jqjpVgf_lU0A7dO4AYmojgvdng-c
MLWSp5YtL_Gw.ewogICAgIl9zZCI6IHsKICAgICAgICAic3ViIjogIltcImVsdVY1T2czZ
1NOSUk4RVluc3hBX0FcIiwgXCI2YzVjMGE0OS1iNTg5LTQzMWQtYmFlNy0yMTkxMjJhOWV
jMmNcIl0iLAogICAgICAgICJnaXZlbl9uYW1lIjogIltcIjZJajd0TS1hNWlWUEdib1M1d
G12VkFcIiwgXCJKb2huXCJdIiwKICAgICAgICAiZmFtaWx5X25hbWUiOiAiW1wiZUk4Wld
tOVFuS1BwTlBlTmVuSGRoUVwiLCBcIkRvZVwiXSIsCiAgICAgICAgImVtYWlsIjogIltcI
lFnX082NHpxQXhlNDEyYTEwOGlyb0FcIiwgXCJqb2huZG9lQGV4YW1wbGUuY29tXCJdIiw
KICAgICAgICAicGhvbmVfbnVtYmVyIjogIltcIkFKeC0wOTVWUHJwVHRONFFNT3FST0FcI
iwgXCIrMS0yMDItNTU1LTAxMDFcIl0iLAogICAgICAgICJhZGRyZXNzIjogIltcIlBjMzN
KTTJMY2hjVV9sSGdndl91ZlFcIiwge1wic3RyZWV0X2FkZHJlc3NcIjogXCIxMjMgTWFpb
iBTdFwiLCBcImxvY2FsaXR5XCI6IFwiQW55dG93blwiLCBcInJlZ2lvblwiOiBcIkFueXN
0YXRlXCIsIFwiY291bnRyeVwiOiBcIlVTXCJ9XSIsCiAgICAgICAgImJpcnRoZGF0ZSI6I
CJbXCJHMDJOU3JRZmpGWFE3SW8wOXN5YWpBXCIsIFwiMTk0MC0wMS0wMVwiXSIKICAgIH0
KfQ
```


(Line breaks for presentation only.)

4.1.3. Example 2 - Structured SD-JWT

In this example, the issuer decided to create a structured object for the hashes. This allows for the release of individual members of the address claim separately.

The user claims are as in Example 1 above. The resulting SD-JWT payload is as follows:

```
{
  "iss": "https://example.com/issuer",
  "sub_jwk": {
    "kty": "RSA",
    "n": "lg9Nie6g-pgoUrDK5Kyni4xZd5ILVnGtBcWx-caAq2FLmtGNIHD9qEzlcLjJCNhrGAUNYOBlkpS0ySJPBlC
    "e": "AQAB"
  },
  "iat": 1516239022,
  "exp": 1516247022,
  "_sd": {
    "sub": "LbnhkOr5oS7KjeUrxezAu8TG0CpWz0jSixy6tffuo04",
    "given_name": "fUMdn88aaoyKTHrvZd6AuLmPraGhPJ0zF5r_JhxCVZs",
    "family_name": "9h5vgv6TpFV6GmnPtugiMLl5tHetHeb5X_2cKHjN7cw",
    "email": "fPZ92dtYMCN2Nb-2ac_zSH19p4yakUXrZl_-wSgaazA",
    "phone_number": "QdSffzNzzd0n60MsSmuiKj6Y6Enk2b-BS-KtEePde5M",
    "address": {
      "street_address": "4FpVpd563Owh9G3HkGNTN9FiSHT0e6y9-Abk_IuG86M",
      "locality": "Kr0BpdZz6yU8HMhjyYHh1EEgJxeUyLIpJEi47iXhp8Y",
      "region": "QXxWKvcV4Bc9t3M7MF43W5vdCnWtA9hsYX8ycWLu1LQ",
      "country": "3itkoMzrDrinn7T0MUbAmrMm1ya1LzbBgif_50WoFOs"
    },
    "birthdate": "fvLCnDm3r4VSYcBF3pIlXP4ulEoHuHOfG_YmFZEuxpQ"
  }
}
```


4.2. SD-JWT Salt/Value Container (SVC)

Besides the SD-JWT itself, the holder needs to learn the raw claim values that are contained in the SD-JWT, along with the precise input to the hash calculation, and the salts. There MAY be other information the issuer needs to communicate to the holder, such as a private key key if the issuer selected the holder key pair.

4.2.1. Payload

A SD-JWT Salt/Value Container (SVC) is a JSON object containing at least the top-level property _sd. Its structure mirrors the one of _sd in the SD-JWT, but the values are the inputs to the hash calculations the issuer used, as strings.

The SVC MAY contain further properties, for example, to transport the holder private key.

4.2.2. Example 1 - SVC for a Flat SD-JWT

The SVC for Example 1 is as follows:

```
{
  "_sd": {
    "sub": "[\"eluV5Og3gSNII8EYnsxA_A\", \"6c5c0a49-b589-431d-bae7-219122a9ec2c\"]",
    "given_name": "[\"6Ij7tM-a5iVPGboS5tmvVA\", \"John\"]",
    "family_name": "[\"eI8ZWm9QnKPpNPeNenHdhQ\", \"Doe\"]",
    "email": "[\"Qg_O64zqAxe412a108iroA\", \"johndoe@example.com\"]",
    "phone_number": "[\"AJx-095VPrpTtN4QMOqROA\", \"+1-202-555-0101\"]",
    "address": "[\"Pc33JM2LchcU_lHggv_ufQ\", {\"street_address\": \"123 Main St\", \"locality
    "birthdate": "[\"G02NSrQfjFXQ7Io09syajA\", \"1940-01-01\"]"
  }
}
```


4.2.3. Example 2 - SVC for a Structured SD-JWT

The SVC for Example 2 is as follows:


```
{
  "_sd": {
    "sub": "[\"eluV5Og3gSNII8EYnsxA_A\", \"6c5c0a49-b589-431d-bae7-219122a9ec2c\"]",
    "given_name": "[\"6Ij7tM-a5iVPGboS5tmvVA\", \"John\"]",
    "family_name": "[\"eI8ZWm9QnKPpNPeNenHdhQ\", \"Doe\"]",
    "email": "[\"Qg_O64zqAxe412a108iroA\", \"johndoe@example.com\"]",
    "phone_number": "[\"AJx-095VPrpTtN4QMOqROA\", \"+1-202-555-0101\"]",
    "address": {
      "street_address": "[\"Pc33JM2LchcU_lHggv_ufQ\", \"123 Main St\"]",
      "locality": "[\"G02NSrQfjFXQ7Io09syajA\", \"Anytown\"]",
      "region": "[\"lklxF5jMYlGTPUovMNIvCA\", \"Anystate\"]",
      "country": "[\"nPuoQnkRFq3BIeAm7AnXFA\", \"US\"]"
    },
    "birthdate": "[\"5bPs1IquZNa0hkaFzzzZNw\", \"1940-01-01\"]"
  }
}
```


4.3. SD-JWT and SVC Combined Format

For transporting the SVC together with the SD-JWT from the issuer to the holder, the SVC is base64ur-encoded and appended to the SD-JWT using . as the separator. For Example 1, the combined format looks as follows:

eyJhbGciOiAiUlMyNTYifQ.eyJpc3MiOiAiaHR0cHM6Ly9leGFtcGxlLmNvbS9pc3N1ZXI
iLCAic3ViX2p3ayI6IHsia3R5IjogIlJTQSIsICJuIjogIjZHd1RUd2NqVnlPdEt0dUdmN
2Z0NVBBVTBHaUR0bkQ0REdjbXRWckZRSFZodHgwNS1ESmlnZm1SLTNUZXR3LU9kNXN1NFR
OWll6amgzdFE2QmoxSFJkT2ZHbVg5RTlZYlB3NGdvS2dfZDBrTTRvWk1VZDY0dG1sQVVGd
FgwTllhWW5Sa2pRdG9rMkNKQlVxMjJ3dWNLOTNKVjExVDM4UFlEQVRxYks5VUZxTU0zdnU
wN1hYbGFRR1hQMXZoNGlYMDR3NGRVNGQyeFRBQ1hob193S0tjVjg1eXZJR3JPMWVHd3duU
2lsVGlxUWJhazMxX1ZuSEdOVlZaRWs0ZG5WTzdlT2M2TVZaYS1xUGtWajc3R2FJTE81M1R
NcTY5VnAxZmFKb0dGSGpoYV9VZTVEOHpmcGlBRXgyQXNBZW90SXdOazJRVDBVWmtlWm9LM
jNRLXM0cDFkUSIsICJlIjogIkFRQUIifSwgImlhdCI6IDE1MTYyMzkwMjIsICJleHAiOiA
xNTE2MjQ3MDIyLCAiX3NkIjogeyJzdWIiOiAiTGJuaGtPcjVvUzdLamVVcnhlekF1OFRHM
ENwV3owalNpeHk2dGZmdW8wNCIsICJnaXZlbl9uYW1lIjogImZVTWRuODhhYW95S1RIcnZ
aZDZBdUxtUHJhR2hQSjB6RjVyX0poeENWWnMiLCAiZmFtaWx5X25hbWUiOiAiOWg1dmd2N
lRwRlY2R21uUHR1Z2lNTGw1dEhldEhlYjVYXzJjS0hqTjdjdyIsICJlbWFpbCI6ICJmUFo
5MmR0WU1DTjJOYi0yYWNfelNIMTlwNHlha1VYclpsXy13U2dhYXpBIiwgInBob25lX251b
WJlciI6ICJRZFNmZnpOenpkMG42ME1zU211aUtqNlk2RW5rMmItQlMtS3RFZVBkZTVNIiw
gImFkZHJlc3MiOiAiSkZ1OTlOVVhQcTU1ZjZERkJaMjJyTWt4TU5IYXlDcmZQRzBGRHNxY
nlEcyIsICJiaXJ0aGRhdGUiOiAiSWExVGM2X1hudDVDSmMyTHRLY3U2V3ZxcjQyZ2xCR0d
jakdPeWU4WmYzVSJ9fQ.rJmWAVghpour5wvdqw8xwdpSEEDMwGJKX1UZ-4mLxYUFv2qCJJ
gQrwtXNccxHpR86F3_51zT9v2TffwZcuU3q4xi-YdyUrVtB6PHHo8F11qanGtnhxqAcFMM
XXQRb7iO_P2Vr7j0Ad8yMcxLituyVLxwjJ0T1s3X-PTomH_zb2wsNsSgrltpjNdoVDHE9k
K8uOWmvx8VMXlaxks74gWjFQoBpnySrlo6PDy2V8zGnj7qc93Qo2Ei01rLYua2jMZJQlRE
ZEp1mI25WYGuz4lJMMjq_SsysLr_r1qGCk1YU12yVz9-xtgL7zVz7KEUY-8TjQEsr_UTbg
vcSUDyd3Smgg.ewogICAgIl9zZCI6IHsKICAgICAgICAic3ViIjogIltcImVsdVY1T2czZ
1NOSUk4RVluc3hBX0FcIiwgXCI2YzVjMGE0OS1iNTg5LTQzMWQtYmFlNy0yMTkxMjJhOWV
jMmNcIl0iLAogICAgICAgICJnaXZlbl9uYW1lIjogIltcIjZJajd0TS1hNWlWUEdib1M1d
G12VkFcIiwgXCJKb2huXCJdIiwKICAgICAgICAiZmFtaWx5X25hbWUiOiAiW1wiZUk4Wld
tOVFuS1BwTlBlTmVuSGRoUVwiLCBcIkRvZVwiXSIsCiAgICAgICAgImVtYWlsIjogIltcI
lFnX082NHpxQXhlNDEyYTEwOGlyb0FcIiwgXCJqb2huZG9lQGV4YW1wbGUuY29tXCJdIiw
KICAgICAgICAicGhvbmVfbnVtYmVyIjogIltcIkFKeC0wOTVWUHJwVHRONFFNT3FST0FcI
iwgXCIrMS0yMDItNTU1LTAxMDFcIl0iLAogICAgICAgICJhZGRyZXNzIjogIltcIlBjMzN
KTTJMY2hjVV9sSGdndl91ZlFcIiwge1wic3RyZWV0X2FkZHJlc3NcIjogXCIxMjMgTWFpb
iBTdFwiLCBcImxvY2FsaXR5XCI6IFwiQW55dG93blwiLCBcInJlZ2lvblwiOiBcIkFueXN
0YXRlXCIsIFwiY291bnRyeVwiOiBcIlVTXCJ9XSIsCiAgICAgICAgImJpcnRoZGF0ZSI6I
CJbXCJHMDJOU3JRZmpGWFE3SW8wOXN5YWpBXCIsIFwiMTk0MC0wMS0wMVwiXSIKICAgIH0
KfQ

(Line breaks for presentation only.)

4.4. SD-JWT-R Format

The following shows the contents of an SD-JWT-R for Example 1:

```
{
  "nonce": "2GLC42sKQveCfGfryNRN9w",
  "aud": "https://example.com/verifier",
  "_sd": {
    "given_name": "[\"6Ij7tM-a5iVPGboS5tmvVA\", \"John\"]",
    "family_name": "[\"eI8ZWm9QnKPpNPeNenHdhQ\", \"Doe\"]",
    "address": "[\"Pc33JM2LchcU_lHggv_ufQ\", {\"street_address\": \"123 Main St\", \"locality
  }
}
```

For each claim, an array of the salt and the claim value is contained in the _sd object.

Again, the SD-JWT-R follows the same structure as the _sd in the SD-JWT. For Example 2, a SD-JWT-R limiting address to region and country only could look as follows:

```
{
  "nonce": "2GLC42sKQveCfGfryNRN9w",
  "aud": "https://example.com/verifier",
  "_sd": {
    "given_name": "[\"6Ij7tM-a5iVPGboS5tmvVA\", \"John\"]",
    "family_name": "[\"eI8ZWm9QnKPpNPeNenHdhQ\", \"Doe\"]",
    "birthdate": "[\"5bPs1IquZNa0hkaFzzzZNw\", \"1940-01-01\"]",
    "address": {
      "region": "[\"lklxF5jMYlGTPUovMNIvCA\", \"Anystate\"]",
      "country": "[\"nPuoQnkRFq3BIeAm7AnXFA\", \"US\"]"
    }
  }
}
```

The SD-JWT-R MAY contain further claims, for example, to ensure a binding to a concrete transaction (in the example the nonce and aud claims).

If holder binding is desired, the SD-JWT-R is signed by the holder. If no holder binding is to be used, the none algorithm is used, i.e., the document is not signed.

In any case, the result is encoded as described in [RFC7519] (here for Example 1):

eyJhbGciOiAiUlMyNTYifQ.eyJub25jZSI6ICIyR0xDNDJzS1F2ZUNmR2ZyeU5STjl3Iiw
gImF1ZCI6ICJodHRwczovL2V4YW1wbGUuY29tL3ZlcmlmaWVyIiwgIl9zZCI6IHsiZ2l2Z
W5fbmFtZSI6ICJbXCI2SWo3dE0tYTVpVlBHYm9TNXRtdlZBXCIsIFwiSm9oblwiXSIsICJ
mYW1pbHlfbmFtZSI6ICJbXCJlSThaV205UW5LUHBOUGVOZW5IZGhRXCIsIFwiRG9lXCJdI
iwgImFkZHJlc3MiOiAiW1wiUGMzM0pNMkxjaGNVX2xIZ2d2X3VmUVwiLCB7XCJzdHJlZXR
fYWRkcmVzc1wiOiBcIjEyMyBNYWluIFN0XCIsIFwibG9jYWxpdHlcIjogXCJBbnl0b3duX
CIsIFwicmVnaW9uXCI6IFwiQW55c3RhdGVcIiwgXCJjb3VudHJ5XCI6IFwiVVNcIn1dIn1
9.b0hG3v71rzHvtoDTdroZ9m-lt9tf8nobFKb2YGiyGOjIklfcKc2KWj72oi_tBKcOCqZh
dX6IV4BRXIw-aspQfLh-xBrNLuGqiC-Y3rZBBlWw0WWnbbtsy1tj8yZOiXBr8vO6mCgZGA
d4MgPYPd-QzOr9ukObYDRB4I24xHrqlAEYPJIzSw9MI_dEmIkNnAuIfLQKiuyTqVVVp6Ly
pBIz6fBLm6NOLC4-uVXlOzI91iT4zlkrhP0-vj8TmfB-XL9aD3-xqytvLBHTESct49OSRZ
FrwkLUKTM56_6KW3pG7Ucuv8VnpHXHIka0SGRaOh8x6v5-rCQJl_IbM8wb7CSHvQ

(Line breaks for presentation only.)

4.5. Presentation Format

The SD-JWT and the SD-JWT-R can be combined into one document using . as a separator (here for Example 1):

eyJhbGciOiAiUlMyNTYifQ.eyJpc3MiOiAiaHR0cHM6Ly9leGFtcGxlLmNvbS9pc3N1ZXI
iLCAic3ViX2p3ayI6IHsia3R5IjogIlJTQSIsICJuIjogIjZHd1RUd2NqVnlPdEt0dUdmN
2Z0NVBBVTBHaUR0bkQ0REdjbXRWckZRSFZodHgwNS1ESmlnZm1SLTNUZXR3LU9kNXN1NFR
OWll6amgzdFE2QmoxSFJkT2ZHbVg5RTlZYlB3NGdvS2dfZDBrTTRvWk1VZDY0dG1sQVVGd
FgwTllhWW5Sa2pRdG9rMkNKQlVxMjJ3dWNLOTNKVjExVDM4UFlEQVRxYks5VUZxTU0zdnU
wN1hYbGFRR1hQMXZoNGlYMDR3NGRVNGQyeFRBQ1hob193S0tjVjg1eXZJR3JPMWVHd3duU
2lsVGlxUWJhazMxX1ZuSEdOVlZaRWs0ZG5WTzdlT2M2TVZaYS1xUGtWajc3R2FJTE81M1R
NcTY5VnAxZmFKb0dGSGpoYV9VZTVEOHpmcGlBRXgyQXNBZW90SXdOazJRVDBVWmtlWm9LM
jNRLXM0cDFkUSIsICJlIjogIkFRQUIifSwgImlhdCI6IDE1MTYyMzkwMjIsICJleHAiOiA
xNTE2MjQ3MDIyLCAiX3NkIjogeyJzdWIiOiAiTGJuaGtPcjVvUzdLamVVcnhlekF1OFRHM
ENwV3owalNpeHk2dGZmdW8wNCIsICJnaXZlbl9uYW1lIjogImZVTWRuODhhYW95S1RIcnZ
aZDZBdUxtUHJhR2hQSjB6RjVyX0poeENWWnMiLCAiZmFtaWx5X25hbWUiOiAiOWg1dmd2N
lRwRlY2R21uUHR1Z2lNTGw1dEhldEhlYjVYXzJjS0hqTjdjdyIsICJlbWFpbCI6ICJmUFo
5MmR0WU1DTjJOYi0yYWNfelNIMTlwNHlha1VYclpsXy13U2dhYXpBIiwgInBob25lX251b
WJlciI6ICJRZFNmZnpOenpkMG42ME1zU211aUtqNlk2RW5rMmItQlMtS3RFZVBkZTVNIiw
gImFkZHJlc3MiOiAiSkZ1OTlOVVhQcTU1ZjZERkJaMjJyTWt4TU5IYXlDcmZQRzBGRHNxY
nlEcyIsICJiaXJ0aGRhdGUiOiAiSWExVGM2X1hudDVDSmMyTHRLY3U2V3ZxcjQyZ2xCR0d
jakdPeWU4WmYzVSJ9fQ.rJmWAVghpour5wvdqw8xwdpSEEDMwGJKX1UZ-4mLxYUFv2qCJJ
gQrwtXNccxHpR86F3_51zT9v2TffwZcuU3q4xi-YdyUrVtB6PHHo8F11qanGtnhxqAcFMM
XXQRb7iO_P2Vr7j0Ad8yMcxLituyVLxwjJ0T1s3X-PTomH_zb2wsNsSgrltpjNdoVDHE9k
K8uOWmvx8VMXlaxks74gWjFQoBpnySrlo6PDy2V8zGnj7qc93Qo2Ei01rLYua2jMZJQlRE
ZEp1mI25WYGuz4lJMMjq_SsysLr_r1qGCk1YU12yVz9-xtgL7zVz7KEUY-8TjQEsr_UTbg
vcSUDyd3Smgg.eyJhbGciOiAiUlMyNTYifQ.eyJub25jZSI6ICIyR0xDNDJzS1F2ZUNmR2
ZyeU5STjl3IiwgImF1ZCI6ICJodHRwczovL2V4YW1wbGUuY29tL3ZlcmlmaWVyIiwgIl9z
ZCI6IHsiZ2l2ZW5fbmFtZSI6ICJbXCI2SWo3dE0tYTVpVlBHYm9TNXRtdlZBXCIsIFwiSm
9oblwiXSIsICJmYW1pbHlfbmFtZSI6ICJbXCJlSThaV205UW5LUHBOUGVOZW5IZGhRXCIs
IFwiRG9lXCJdIiwgImFkZHJlc3MiOiAiW1wiUGMzM0pNMkxjaGNVX2xIZ2d2X3VmUVwiLC
B7XCJzdHJlZXRfYWRkcmVzc1wiOiBcIjEyMyBNYWluIFN0XCIsIFwibG9jYWxpdHlcIjog
XCJBbnl0b3duXCIsIFwicmVnaW9uXCI6IFwiQW55c3RhdGVcIiwgXCJjb3VudHJ5XCI6IF
wiVVNcIn1dIn19.b0hG3v71rzHvtoDTdroZ9m-lt9tf8nobFKb2YGiyGOjIklfcKc2KWj7
2oi_tBKcOCqZhdX6IV4BRXIw-aspQfLh-xBrNLuGqiC-Y3rZBBlWw0WWnbbtsy1tj8yZOi
XBr8vO6mCgZGAd4MgPYPd-QzOr9ukObYDRB4I24xHrqlAEYPJIzSw9MI_dEmIkNnAuIfLQ
KiuyTqVVVp6LypBIz6fBLm6NOLC4-uVXlOzI91iT4zlkrhP0-vj8TmfB-XL9aD3-xqytvL
BHTESct49OSRZFrwkLUKTM56_6KW3pG7Ucuv8VnpHXHIka0SGRaOh8x6v5-rCQJl_IbM8w
b7CSHvQ

(Line breaks for presentation only.)

5. Verification

Verifiers MUST follow [RFC8725] for checking the SD-JWT and, if signed, the SD-JWT Release.

Verifiers MUST go through (at least) the following steps before trusting/using any of the contents of an SD-JWT:

1. Determine if holder binding is to be checked for the SD-JWT. Refer to Section 6.1 for details.

2. Check that the presentation consists of six .-separated elements; if holder binding is not required, the last element can be empty.

3. Separate the SD-JWT from the SD-JWT Release.

4. Validate the SD-JWT:

    1. Ensure that a signing algorithm was used that was deemed secure for the application. Refer to [RFC8725], Sections 3.1 and 3.2 for details.

    2. Validate the signature over the SD-JWT.

    3. Validate the issuer of the SD-JWT and that the signing key belongs to this issuer.

    4. Check that the SD-JWT is valid using nbf, iat, and exp claims, if provided in the SD-JWT.

    5. Check that the claim _sd is present in the SD-JWT.

5. Validate the SD-JWT Release:

    1. If holder binding is required, validate the signature over the SD-JWT using the same steps as for the SD-JWT plus the following steps:

        1. Determine that the public key for the private key that used to sign the SD-JWT-R is bound to the SD-JWT, i.e., the SD-JWT either contains a reference to the public key or contains the public key itself.

        2. Determine that the SD-JWT-R is bound to the current transaction and was created for this verifier (replay protection). This is usually achieved by a nonce and aud field within the SD-JWT Release.

    2. For each claim in the SD-JWT Release:

        1. Ensure that the claim is present as well in _sd in the SD-JWT. If _sd is structured, the claim MUST be present at the same place within the structure.

        2. Check that the base64url-encoded hash of the claim value in the SD-JWT-R (which includes the salt and the actual claim value) matches the hash provided in the SD-JWT.

3. Ensure that the claim value in the SD-JWT-R is a JSON-encoded array of exactly two values.

4. Store the second of the two values.

3. Once all necessary claims have been verified, their values can be validated and used according to the requirements of the application. It MUST be ensured that all claims required for the application have been released.

If any step fails, the input is not valid and processing MUST be aborted.

6. Security Considerations

For the security of this scheme, the following properties are required of the hash function:

* Given a claim value, a salt, and the resulting hash, it is hard to find a second salt value so that HASH(salt | claim_value) equals the hash.

Add: The Salts must be random/long enough so that the attacker cannot brute force them.

Note: No need for the wallet-generated hashes? to prevent issuer-verifier collusion

6.1. Holder Binding

7. Privacy Considerations

7.1. Claim Names

Claim names are not hashed in the SD-JWT and are used as keys in a key-value pair, where the value is the hash. This is because SD-JWT already reveals information about the issuer and the schema, and revealing the claim names does not provide any additional information.

7.2. Unlinkability

It is also important to note that this format enables selective disclosure of claims, but in itself it does not achieve unlinkability of the subject of a JWS document.

8. Acknowledgements

We would like to thank ...

9. IANA Considerations

TBD

10. Normative References

[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>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/info/rfc7515>.

[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>.

[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>.

11. Informative References

[RFC8725] Sheffer, Y., Hardt, D., and M. Jones, "JSON Web Token Best Current Practices", BCP 225, RFC 8725, DOI 10.17487/RFC8725, February 2020, <https://www.rfc-editor.org/info/rfc8725>.

[OIDC] Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., and C. Mortimore, "OpenID Connect Core 1.0 incorporating errata set 1", 8 November 2014, <https://openid.net/specs/openid-connect-core-1_0.html>.

Appendix A. Additional Examples

A.1. Example 3 - Complex Structured SD-JWT

In this example, a complex object such as those used for ekyc (todo reference) is used.

These claims are the payload of the SD-JWT:

```
{
  "verified_claims": {
    "verification": {
      "trust_framework": "de_aml",
      "time": "2012-04-23T18:25Z",
      "verification_process": "f24c6f-6d3f-4ec5-973e-b0d8506f3bc7",
      "evidence": [
        {
          "type": "document",
          "method": "pipp",
          "time": "2012-04-22T11:30Z",
          "document": {
            "type": "idcard",
            "issuer": {
              "name": "Stadt Augsburg",
              "country": "DE"
            },
            "number": "53554554",
            "date_of_issuance": "2010-03-23",
            "date_of_expiry": "2020-03-22"
          }
        }
      ]
    },
    "claims": {
      "given_name": "Max",
      "family_name": "Meier",
      "birthdate": "1956-01-28",
      "place_of_birth": {
        "country": "DE",
        "locality": "Musterstadt"
      },
      "nationalities": [
        "DE"
      ],
      "address": {
        "locality": "Maxstadt",
        "postal_code": "12344",
        "country": "DE",
        "street_address": "An der Weide 22"
      }
    }
  },
  "birth_middle_name": "Timotheus",
  "salutation": "Dr.",
  "msisdn": "49123456789"
}
```

The following shows the resulting SD-JWT payload:

```
{
  "iss": "https://example.com/issuer",
  "sub_jwk": {
    "kty": "RSA",
    "n": "wlcpuJjcdshL2NqMsT2HMJiqyPFEPQZ2FMXKD_r3an-09_k-cdoJifVodKUZ8QBiU6w_JUYh3lScrJ-TSEu
    "e": "AQAB"
  },
  "iat": 1516239022,
  "exp": 1516247022,
  "_sd": {
    "verified_claims": {
      "verification": {
        "trust_framework": "UI-SRNlQFy-YEFE46yyHKqc64jmM65q8ma9cq2V_erY",
        "time": "jI-FYlteydXzsjRIrXBZs9foBSNF1Od1Q-4XnuqpgjI",
        "verification_process": "F979I7b5ZhADtyYMlYxctdc9-IalD_Td0HpfcFBzVXs",
        "evidence": [
          {
            "type": "i2w3mrKAQV2nhTa5c2koZ-aQTBDoSaVfvYk7aLQianc",
            "method": "fEQ0tVPD67GfO30h_SRs8ZPbnZ_vwEt5S8lUOR77va0",
            "time": "9jueDP5r0gTB64DqdCZbek3yaS5AJJnW8FEkWtPTaOk",
            "document": {
              "type": "K-rZQk89w89YBhjUNUho07suLxhG8Sl2JTPAcoAJB34",
              "issuer": {
                "name": "BkCULCU-txVGvzNqnWe5DxefFvJE8LMib8GV3I3WO90",
                "country": "DSyF5TtmYgLk92u4GkDQzSdFbvIbw5rkFjzSsJJsyw4"
              },
              "number": "epH3OuU51TBelOE4PX6ueHwr1ZtoUjzG-7pZjIAsXg8",
              "date_of_issuance": "cVvqTueVq6OWz-dJj2cdo19A0Ajj859eGDzDfwPYyN4",
              "date_of_expiry": "nxJBNdtwvb2TKKJNGvF6_1ywEdKrotj66C88WPomLfo"
            }
          }
        ]
      },
      "claims": {
        "given_name": "y9uFPHAVqNAZ7PJyk1-1yQJZZWZzKGP5FLt9txKM84M",
        "family_name": "XyUikY8V8MWeBfXUOp8gI7F7-yC28Jr5IyDgvBxXzd4",
        "birthdate": "7GlieMLJhM78C_uQQp9wUXSZLeqBN1YGQT87BIubyKU",
        "place_of_birth": {
          "country": "RN3xcnLYX_GDhVwfPvtisuLPfi0d74zqihFbQrd_UG0",
          "locality": "iNkpWqJ9kIZQq95dzSyEZjbPJs6Fqu7GFBKouEC3OxE"
        },
        "nationalities": "-tinYGK0GXnkfARxiNIWq0VnzNRl-Kv3KY3m5g5Femg",
        "address": "63EzPV0yvTpeOgV34yCwweCvO-2wxts2Wqbja_SuwPQ"
      }
    },
    "birth_middle_name": "vM68I6XnrVlyt1LxK9xxgFycsjtw2vLdGpNgk3E8QQ4",
    "salutation": "iThfCu2ulLoe5i6gCEq--Y6R-gxHHtIukXb9qnfjH5k",
    "msisdn": "xUpU-azBYdXeJidc8Yw5MXtfPz4_4kArJhflXcxzkzs"
```

```
  }
}
```

The SD-JWT is then signed by the issuer to create a document like the following:

eyJhbGciOiAiUlMyNTYifQ.eyJpc3MiOiAiaHR0cHM6Ly9leGFtcGxlLmNvbS9pc3N1ZXI
iLCAic3ViX2p3ayI6IHsia3R5IjogIlJTQSIsICJuIjogIndsY3B1SmpjZHNoTDJOcU1zV
DJITUppcXlQRkVQUVoyRk1YS0RfcjNhbi0wOV9rLWNkb0ppZlZvZEtVWjhRQmlVNndfSlV
ZaDNsU2NySi1UU0V1ZVMtbUdVOUtrbl85cTV4elhiblRFeTJQNk5fNGI3Tk5hZUplMjVEb
mtsRVBVSlU1dFJjak9EdEhETzdNSWdzSVBLLU50Q1c2eDdZQjJ4Nl8zSDJ0NkVmUnREalp
lLWFkZWxPQTA5VVE4eDlkOERCT1IycnhhRm0zX1FBbEFmSHZEN2xMbGV3QWlHdmpNVGQ2M
TRGa0E4Q0VxbG5TV2w0Z0QyUGQzQTVKM1hfMlVESzBJcUliOGVkOF9iMnlUb1JJd3F6VTF
nMXAyVkVaQkowWEdTYmVTWTllLUVOX1NjclZUNVlCd29JV292QjBBZVNzTWo1Szhla2VkY
Ws5UHdFZk12USIsICJlIjogIkFRQUIifSwgImlhdCI6IDE1MTYyMzkwMjIsICJleHAiOiA
xNTE2MjQ3MDIyLCAiX3NkIjogeyJ2ZXJpZmllZF9jbGFpbXMiOiB7InZlcmlmaWNhdGlvb
iI6IHsidHJ1c3RfZnJhbWV3b3JrIjogIlVJLVNSTmxRRnktWUVGRTQ2eXlIS3FjNjRqbU0
2NXE4bWE5Y3EyVl9lclkiLCAidGltZSI6ICJqSS1GWWx0ZXlkWHpzalJJclhCWnM5Zm9CU
05GMU9kMVEtNFhudXFwZ2pJIiwgInZlcmlmaWNhdGlvbl9wcm9jZXNzIjogIkY5NzlJN2I
1WmhBRHR5WU1sWXhjdGRjOS1JYWxEX1RkMEhwZmNGQnpWWHMiLCAiZXZpZGVuY2UiOiBBe
yJ0eXBlIjogImkydzNtcktBUVYybmhUYTVjMmtvWi1hUVRCRG9TYVZmdllrN2FMUWlhbmM
iLCAibWV0aG9kIjogImZFUTB0VlBENjdHZk8zMGhfU1JzOFpQYm5aX3Z3RXQ1UzhsVU9SN
zd2YTAiLCAidGltZSI6ICI5anVlRFA1cjBnVEI2NERxZENaYmVrM3lhUzVBSkpuVzhGRWt
XdFBUYU9rIiwgImRvY3VtZW50IjogeyJ0eXBlIjogIkstclpRazg5dzg5WUJoalVOVWhvM
DdzdUx4aEc4U2wySlRQQWNvQUpCMzQiLCAiaXNzdWVyIjogeyJuYW1lIjogIkJrQ1VMQ1U
tdHhWR3Z6TnFuV2U1RHhlZkZ2SkU4TE1pYjhHVjNJM1dPOTAiLCAiY291bnRyeSI6ICJEU
3lGNVR0bVlnTGs5MnU0R2tEUXpTZEZidklidzVya0ZqelNzSkpzeXc0In0sICJudW1iZXI
iOiAiZXBIM091VTUxVEJlbE9FNFBYNnVlSHdyMVp0b1VqekctN3BaaklBc1hnOCIsICJkY
XRlX29mX2lzc3VhbmNlIjogImNWdnFUdWVWcTZPV3otZEpqMmNkbzE5QTBBamo4NTllR0R
6RGZ3UFl5TjQiLCAiZGF0ZV9vZl9leHBpcnkiOiAibnhKQk5kdHd2YjJUS0tKTkd2RjZfM
Xl3RWRLcm90ajY2Qzg4V1BvbUxmbyJ9fV19LCAiY2xhaW1zIjogeyJnaXZlbl9uYW1lIjo
gInk5dUZQSEFWcU5BWjdQSnlrMS0xeVFKWlpXWnpLR1A1Rkx0OXR4S004NE0iLCAiZmFta
Wx5X25hbWUiOiAiWHlVaWtZOFY4TVdlQmZYVU9wOGdJN0Y3LXlDMjhKcjVJeURndkJ4WHp
kNCIsICJiaXJ0aGRhdGUiOiAiN0dsaWVNTEpoTTc4Q191UVFwOXdVWFNaTGVxQk4xWUdRV
Dg3Qkl1YnlLVSIsICJwbGFjZV9vZl9iaXJ0aCI6IHsiY291bnRyeSI6ICJSTjN4Y25MWVh
fR0RoVndmUHZ0aXN1TFBmaTBkNzR6cWloRmJRcmRfVUcwIiwgImxvY2FsaXR5IjogImlOa
3BXcUo5a0laUXE5NWR6U3lFWmpiUEpzNkZxdTdHRkJLb3VFQzNPeEUifSwgIm5hdGlvbmF
saXRpZXMiOiAiLXRpbllHSzBHWG5rZkFSeGlOSVdxMFZuek5SbC1LdjNLWTNtNWc1RmVtZ
yIsICJhZGRyZXNzIjogIjYzRXpQVjB5dlRwZU9nVjM0eUN3d2VDdk8tMnd4dHMyV3FiamF
fU3V3UFEifX0sICJiaXJ0aF9taWRkbGVfbmFtZSI6ICJ2TTY4STZYbnJWbHl0MUx4Szl4e
GdGeWNzanR3MnZMZEdwTmdrM0U4UVE0IiwgInNhbHV0YXRpb24iOiAiaVRoZkN1MnVsTG9
lNWk2Z0NFcS0tWTZSLWd4SEh0SXVrWGI5cW5makg1ayIsICJtc2lzZG4iOiAieFVwVS1he
kJZZFhlSmlkYzhZdzVNWHRmUHo0XzRrQXJKaGZsWGN4emt6cyJ9fQ.PSEqS4wRCkLuFfGT
gNjw63kewpAxNWu1kgo_tat17ElyetqMO49w3PL1D4Z67AeVbOMT3DhG7WiJB9UoKVs3XJ
yJuZ0DBRkCQ8iqaIw3vyA-P_lkTw7EwSO-Klo0UHBUfHvvJhKOeZ6jhSvZhtlD0yYxoS9e
fxOHM2tUnpm7gaWQ6OqXiTTLGrnuA-1k99IL_ag5oJJym2JvlWt2R1S8tLvKrKZumrPi5R
LYskZlEiZ_l4h_n7FHva9S66R_tvZNcXRQqyXGKdp66rzmfHzxoHHYBIUfgxBZ0re3zkHm
JcgAgoPLaIpXm3cR-4dmMpDHgntMOWP8s0hnBsYStArM4Q.ewogICAgIl9zZCI6IHsKICA
gICAgICAidmVyaWZpZWRfY2xhaW1zIjogewogICAgICAgICAidmVyaWZpY2F0aW9uI
jogewogICAgICAgICAgICAgInRydXN0X2ZyYW1ld29yayI6ICJbXCJlbHVWNU9nM2d
TTklJOEVZbnN4QV9BXCIsIFwiZGVfYW1sXCJdIiwKICAgICAgICAgICAgICAgICJ0aW1lI
jogIltcIjZJajd0TS1hNWlWUEdib1M1dG12VkFcIiwgXCIyMDEyLTA0LTIzVDE4OjI1Wlw
iXSIsCiAgICAgICAgICAgICAgICAidmVyaWZpY2F0aW9uX3Byb2Nlc3MiOiAiW1wiZUk4W
ldtOVFuS1BwTlBlTmVuSGRoUVwiLCBcImYyNGM2Zi02ZDNmLTRlYzUtOTczZS1iMGQ4NTA
2ZjNiYzdcIl0iLAogICAgICAgICAgICAgICAgImV2aWRlbmNlIjogWwogICAgICAgICAgI

CAgICAgICAgIHsKICAgICAgICAgICAgICAgICAgICAgICAgInR5cGUiOiAiW1wiUWdfTzY
0enFBeGU0MTJhMTA4aXJvQVwiLCBcImRvY3VtZW50XCJdIiwKICAgICAgICAgICAgICAgI
CAgICAgICAgIm1ldGhvZCI6ICJbXCJBSngtMDk1VlBycFR0TjRRTU9xUk9BXCIsIFwicGl
wcFwiXSIsCiAgICAgICAgICAgICAgICAgICAgICAgICJ0aW1lIjogIltcIlBjMzNKTTJMY
2hjVV9sSGdndl91ZlFcIiwgXCIyMDEyLTA0LTIyVDExOjMwWlwiXSIsCiAgICAgICAgICA
gICAgICAgICAgICAgICJkb2N1bWVudCI6IHsKICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICJ0eXBlIjogIltcIkcwMk5TclFmakZYUTdJbzA5c3lhakFcIiwgXCJpZGNhcmRcIl0
iLAogICAgICAgICAgICAgICAgICAgICAgICAgICAgImlzc3VlciI6IHsKICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAibmFtZSI6ICJbXCJsa2x4RjVqTVlsR1RQVW92TU5
JdkNBXCIsIFwiU3RhZHQgQXVnc2J1cmdcIl0iLAogICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICJjb3VudHJ5IjogIltcIm5QdW9RbmtSRnEzQkllQW03QW5YRkFcIiwgXCJ
ERVwiXSIKICAgICAgICAgICAgICAgICAgICAgICAgICAgIH0sCiAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAibnVtYmVyIjogIltcIjViUHMxSXF1Wk5hMGhrYUZ6enpaTndcIiw
gXCI1MzU1NDU1NFwiXSIsCiAgICAgICAgICAgICAgICAgICAgICAgICAiZGF0ZV9vZ
l9pc3N1YW5jZSI6ICJbXCI1YTJXMF9OcmxFWnpmcW1rXzdQcS13XCIsIFwiMjAxMC0wMy0
yM1wiXSIsCiAgICAgICAgICAgICAgICAgICAgICAgICAiZGF0ZV9vZl9leHBpcnkiO
iAiW1wieTFzVlU1d2RmSmFoVmRnd1BnUzdSUVwiLCBcIjIwMjAtMDMtMjJcIl0iCiAgICA
gICAgICAgICAgICAgICAgIH0KICAgICAgICAgICAgICAgICB9CiAgICAgICAgI
CAgICAgICBdCiAgICAgICAgICAgIH0sCiAgICAgICAgICJjbGFpbXMiOiB7CiAgICA
gICAgICAgICAiZ2l2ZW5fbmFtZSI6ICJbXCJIYlE0WDhzclZXM1FEeG5JSmRxeU9BX
CIsIFwiTWF4XCJdIiwKICAgICAgICAgICAgICJmYW1pbHlfbmFtZSI6ICJbXCJDOUd
Tb3VqdmlKcXVFZ1lmb2pDYjFBXCIsIFwiTWVpZXJcIl0iLAogICAgICAgICAgICAgICAgI
mJpcnRoZGF0ZSI6ICJbXCJreDVrRjE3Vi14MEptd1V4OXZndnR3XCIsIFwiMTk1Ni0wMS0
yOFwiXSIsCiAgICAgICAgICAgICAicGxhY2Vfb2ZfYmlydGgiOiB7CiAgICAgICAgI
CAgICAgICAgImNvdW50cnkiOiAiW1wiSDNvMXVzd1A3NjBGaTJ5ZUdkVkNFUVwiLCB
cIkRFXCJdIiwKICAgICAgICAgICAgICAgICAibG9jYWxpdHkiOiAiW1wiT0JLbFRWb
HZMZy1BZHdxWUdiUDhaQVwiLCBcIk11c3RlcnN0YWR0XCJdIgogICAgICAgICAgICA
gfSwKICAgICAgICAgICAgICJuYXRpb25hbGl0aWVzIjogIltcIk0wSmI1N3Q0MXVic
mtTdXlyRFQzeEFcIiwgW1wiREVcIl1dIiwKICAgICAgICAgICAgICJhZGRyZXNzIjo
gIltcIkRzbXRLTmdwVjRkQUhwanJjYW9zQXdcIiwge1wibG9jYWxpdHlcIjogXCJNYXhzd
GFkdFwiLCBcInBvc3RhbF9jb2RlXCI6IFwiMTIzNDRcIiwgXCJjb3VudHJ5XCI6IFwiREV
cIiwgXCJzdHJlZXRfYWRkcmVzc1wiOiBcIkFuIGRlciBXZWlkZSAyMlwifV0iCiAgICAgI
CAgICAgIH0KICAgICAgICB9LAogICAgICAgICJiaXJ0aF9taWRkbGVfbmFtZSI6ICJbXCJ
lSzVvNXBIZmd1cFBwbHRqMXFoQUp3XCIsIFwiVGltb3RoZXVzXCJdIiwKICAgICAgICAic
2FsdXRhdGlvbiI6ICJbXCJqN0FEZGIwVVZiMExpMGNpUGNQMGV3XCIsIFwiRHIuXCJdIiw
KICAgICAgICAibXNpc2RuIjogIltcIldweEpyRnVYOHVTaTJwNGh0MDlqdndcIiwgXCI0O
TEyMzQ1Njc4OVwiXSIKICAgIH0KfQ

(Line breaks for presentation only.)

A SD-JWT-R for some of the claims:

```
{
  "nonce": "2GLC42sKQveCfGfryNRN9w",
  "aud": "https://example.com/verifier",
  "_sd": {
    "verified_claims": {
      "verification": {
        "trust_framework": "[\"eluV5Og3gSNII8EYnsxA_A\", \"de_aml\"]",
        "time": "[\"6Ij7tM-a5iVPGboS5tmvVA\", \"2012-04-23T18:25Z\"]",
        "evidence": [
          {
            "type": "[\"Qg_O64zqAxe412a108iroA\", \"document\"]"
          }
        ]
      },
      "claims": {
        "given_name": "[\"HbQ4X8srVW3QDxnIJdqyOA\", \"Max\"]",
        "family_name": "[\"C9GSoujviJquEgYfojCb1A\", \"Meier\"]",
        "birthdate": "[\"kx5kF17V-x0JmwUx9vgvtw\", \"1956-01-28\"]",
        "place_of_birth": {
          "country": "[\"H3o1uswP760Fi2yeGdVCEQ\", \"DE\"]"
        }
      }
    }
  }
}
```

A.2. Example 4 - W3C VC


```
{
  "sub": "did:example:ebfeb1f712ebc6f1c276e12ec21",
  "jti": "http://example.edu/credentials/3732",
  "iss": "https://example.com/keys/foo.jwk",
  "nbf": 1541493724,
  "iat": 1541493724,
  "exp": 1573029723,
  "nonce": "660!6345FSer",
  "vc": {
    "@context": [
      "https://www.w3.org/2018/credentials/v1",
      "https://www.w3.org/2018/credentials/examples/v1"
    ],
    "type": [
      "VerifiableCredential",
      "UniversityDegreeCredential"
    ]
  },
  "_sd": {
    "given_name": "LbnhkOr5oS7KjeUrxezAu8TG0CpWz0jSixy6tffuo04",
    "family_name": "9h5vgv6TpFV6GmnPtugiMLl5tHetHeb5X_2cKHjN7cw",
    "birthdate": "fPZ92dtYMCN2Nb-2ac_zSH19p4yakUXrZl_-wSgaazA"
  }
}
```


Appendix B. Document History

[[To be removed from the final specification]]

-00

- *Renamed to SD-JWT (focus on JWT instead of JWS since signature is optional)
- *Make holder binding optional
- *Rename proof to release, since when there is no signature, the term "proof" can be misleading
- *Improved the structure of the description
- *Described verification steps
- *All examples generated from python demo implementation
- *Examples for structured objects

Authors' Addresses

Daniel Fett
yes.com

Email: mail@danielfett.de
URI: https://danielfett.de/

Kristina Yasuda
Microsoft

Email: Kristina.Yasuda@microsoft.com