SD-JWT-based Verifiable Credentials (SD-JWT VC)

Abstract

This specification describes data formats as well as validation and processing rules to express Verifiable Credentials with JSON payloads with and without selective disclosure based on the SD-JWT [I-D.ietf-oauth-selective-disclosure-jwt] format.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Web Authorization Protocol Working Group mailing list (oauth@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/oauth/.

Source for this draft and an issue tracker can be found at https://github.com/oauth-wg/oauth-sd-jwt-vc.

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 30 August 2024.

Copyright Notice

Copyright (c) 2024 IETF Trust and the persons identified as the document authors. All rights reserved.
Table of Contents

1. Introduction
   1.1. Issuer-Holder-Verifier Model
   1.2. SD-JWT as a Credential Format
   1.3. Requirements Notation and Conventions
   1.4. Terms and Definitions

2. Scope

3. Verifiable Credentials based on SD-JWT
   3.1. Media Type
   3.2. Data Format
      3.2.1. JOSE Header
      3.2.2. JWT Claims Set
   3.3. Example
   3.4. Verification and Processing
   3.5. Issuer-signed JWT Verification Key Validation

4. Presenting Verifiable Credentials
   4.1. Key Binding JWT
   4.2. Examples

5. JWT VC Issuer Metadata
   5.1. JWT VC Issuer Metadata Request
   5.2. JWT VC Issuer Metadata Response
   5.3. JWT VC Issuer Metadata Validation

6. Security Considerations
   6.1. Server-Side Request Forgery
   6.2. Ecosystem-specific Public Key Verification Methods

7. Privacy Considerations
   7.1. Unlinkability
   7.2. Verifiable Credential Type Identifier
   7.3. Issuer Phone-Home

8. Relationships to Other Documents

9. References
   9.1. Normative References
   9.2. Informative References

Appendix A. IANA Considerations
   A.1. JSON Web Token Claims Registration
   A.2. Media Types Registry
      A.2.1. application/vc+sd-jwt
1. Introduction

1.1. Issuer-Holder-Verifier Model

In the so-called Issuer-Holder-Verifier Model, Issuers issue so-called Verifiable Credentials to a Holder, who can then present the Verifiable Credentials to Verifiers. Verifiable Credentials are cryptographically signed statements about a Subject, typically the Holder.

![Issuer-Holder-Verifier Model with optional Status Provider](image)

Verifiers can check the authenticity of the data in the Verifiable Credentials and optionally enforce Key Binding, i.e., ask the Holder...
to prove that they are the intended holder of the Verifiable Credential, for example, by proving possession of a cryptographic key referenced in the credential. This process is further described in [I-D.ietf-oauth-selective-disclosure-jwt].

To support revocation of Verifiable Credentials, revocation information can optionally be retrieved from a Status Provider. The role of a Status Provider can be fulfilled by either a fourth party or by the Issuer.

1.2. SD-JWT as a Credential Format

JSON Web Tokens (JWTs) [RFC7519] can in principle be used to express Verifiable Credentials in a way that is easy to understand and process as it builds upon established web primitives.

Selective Disclosure JWT (SD-JWT) [I-D.ietf-oauth-selective-disclosure-jwt] is a specification that introduces conventions to support selective disclosure for JWTs: For an SD-JWT document, a Holder can decide which claims to release (within bounds defined by the Issuer).

SD-JWT is a superset of JWT as it can also be used when there are no selectively disclosable claims and also supports JWS JSON serialization, which is useful for long term archiving and multi signatures. However, SD-JWT itself does not define the claims that must be used within the payload or their semantics.

This specification uses SD-JWT and the well-established JWT content rules and extensibility model as basis for representing Verifiable Credentials with JSON payloads. These Verifiable Credentials are called SD-JWT VCs. The use of selective disclosure in SD-JWT VCs is OPTIONAL.

SD-JWTs VC can contain claims that are registered in "JSON Web Token Claims" registry as defined in [RFC7519], as well as public and private claims.

Note: This specification does not utilize the W3C's Verifiable Credentials Data Model v1.0, v1.1, or v2.0.

1.3. Requirements Notation and Conventions

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 RFC 2119 [RFC2119].
1.4. Terms and Definitions

This specification uses the terms "Holder", "Issuer", "Verifier", "Key Binding", and "Key Binding JWT" defined by [I-D.ietf-oauth-selective-disclosure-jwt].

Verifiable Credential (VC): An Issuer-signed assertion with claims about a Subject.

SD-JWT-based Verifiable Credential (SD-JWT VC): A Verifiable Credential encoded using the format defined in [I-D.ietf-oauth-selective-disclosure-jwt]. It may or may not contain selectively disclosable claims.

Unsecured Payload of an SD-JWT VC: A JSON object containing all selectively disclosable and non-selectively disclosable claims of the SD-JWT VC. The Unsecured Payload acts as the input JSON object to issue an SD-JWT VC complying to this specification.

Status Provider: An entity that provides status information (e.g. revocation) about a Verifiable Credential.

2. Scope

*This specification defines

- Data model and media types for Verifiable Credentials based on SD-JWTs.
- Validation and processing rules for Verifiers and Holders.

3. Verifiable Credentials based on SD-JWT

This section defines encoding, validation and processing rules for SD-JWT VCs.

3.1. Media Type

SD-JWT VCs compliant with this specification MUST use the media type application/vc+sd-jwt as defined in Appendix A.2.1.

3.2. Data Format

SD-JWT VCs MUST be encoded using the SD-JWT format defined in Section 5 of [I-D.ietf-oauth-selective-disclosure-jwt]. A presentation of an SD-JWT VC MAY contain a Key Binding JWT.

Note that in some cases, an SD-JWT VC MAY have no selectively disclosable claims, and therefore the encoded SD-JWT will not contain any Disclosures.
3.2.1. JOSE Header

This section defines JWT header parameters for the SD-JWT component of the SD-JWT VC.

The typ header parameter of the SD-JWT MUST be present. The typ value MUST use vc+sd-jwt. This indicates that the payload of the SD-JWT contains plain JSON and follows the rules as defined in this specification. It further indicates that the SD-JWT is a SD-JWT component of a SD-JWT VC.

The following is a non-normative example of a decoded SD-JWT header:

```json
{
    "alg": "ES256",
    "typ": "vc+sd-jwt"
}
```

3.2.2. JWT Claims Set

This section defines the claims that can be included in the payload of SD-JWT VCs.

3.2.2.1. New JWT Claims

3.2.2.1.1. Verifiable Credential Type - vct Claim

This specification defines the JWT claim vct (for verifiable credential type). The vct value MUST be a case-sensitive StringOrURI (see [RFC7519]) value serving as an identifier for the type of the SD-JWT VC. The vct value MUST be a Collision-Resistant Name as defined in Section 2 of [RFC7515].

A type is associated with rules defining which claims may or must appear in the Unsecured Payload of the SD-JWT VC and whether they may, must, or must not be selectively disclosable. This specification does not define any vct values; instead it is expected that ecosystems using SD-JWT VCs define such values including the semantics of the respective claims and associated rules (e.g., policies for issuing and validating credentials beyond what is defined in this specification).

The following is a non-normative example of how vct is used to express a type:

```json
{
    "vct": "https://credentials.example.com/identity_credential"
}
```
For example, a value of https://credentials.example.com/identity_credential can be associated with rules that define that at least the registered JWT claims given_name, family_name, birthdate, and address must appear in the Unsecured Payload. Additionally, the registered JWT claims email and phone_number, and the private claims is_over_18, is_over_21, and is_over_65 may be used. The type might also indicate that any of the aforementioned claims can be selectively disclosable.

### 3.2.2.2. Registered JWT Claims

SD-JWT VCs MAY use any claim registered in the "JSON Web Token Claims" registry as defined in [RFC7519].

If present, the following registered JWT claims MUST be included in the SD-JWT and MUST NOT be included in the Disclosures, i.e. cannot be selectively disclosed:

*iss

-REQUIRED. The Issuer of the Verifiable Credential. The value of iss MUST be a URI. See [RFC7519] for more information.

*nbf

-OPTIONAL. The time before which the Verifiable Credential MUST NOT be accepted before validating. See [RFC7519] for more information.

*exp

-OPTIONAL. The expiry time of the Verifiable Credential after which the Verifiable Credential is no longer valid. See [RFC7519] for more information.

*cnf

-OPTIONAL unless cryptographic Key Binding is to be supported, in which case it is REQUIRED. Contains the confirmation method identifying the proof of possession key as defined in [RFC7800]. It is RECOMMENDED that this contains a JWK as defined in Section 3.2 of [RFC7800]. For proof of cryptographic Key Binding, the Key Binding JWT in the presentation of the SD-JWT MUST be signed by the key identified in this claim.

*vct

-REQUIRED. The type of the Verifiable Credential, e.g., https://credentials.example.com/identity_credential, as defined in Section 3.2.2.1.1.
*status

- OPTIONAL. The information on how to read the status of the Verifiable Credential. See [I-D.looker-oauth-jwt-cwt-status-list] for more information.

The following registered JWT claims MAY be contained in the SD-JWT or in the Disclosures and MAY be selectively disclosed:

*sub

- OPTIONAL. The identifier of the Subject of the Verifiable Credential. The Issuer MAY use it to provide the Subject identifier known by the Issuer. There is no requirement for a binding to exist between sub and cnf claims.

*iat

- OPTIONAL. The time of issuance of the Verifiable Credential. See [RFC7519] for more information.

3.2.2.3. Public JWT claims

Additional public claims MAY be used in SD-JWT VCs depending on the application.

3.2.2.4. SD-JWT VC without Selectively Disclosable Claims

An SD-JWT VC MAY have no selectively disclosable claims. In that case, the SD-JWT VC MUST NOT contain the _sd claim in the JWT body. It also MUST NOT have any Disclosures.

3.3. Example

The following is a non-normative example of an unsecured payload of an SD-JWT VC.
The following is a non-normative example of how the unsecured payload of the SD-JWT VC above can be used in a SD-JWT where the resulting SD-JWT VC contains only claims about the Subject that are selectively disclosable:

```json
{
  "vct": "https://credentials.example.com/identity_credential",
  "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,
  "is_over_21": true,
  "is_over_65": true
}
```
Note that a cnf claim has been added to the SD-JWT payload to express the confirmation method of the Key Binding.

The following are the Disclosures belonging to the SD-JWT payload above:

**Claim given_name**:

*SHA-256 Hash: jsu9yVu1wQQ1hF1M_3JlzMaSFzglhQG0DpfayQwLUK4
*Disclosure:
  WyIyR0xDNDJzS1F2ZUNmR2ZyeU5STj13IiwgImdpdmVuX25hbWUiLCAiSm9obiIg
  *Contents: ["2GLC42sKQveCfGfryNRN9w", "given_name", "John"]

**Claim family_name**:

*SHA-256 Hash: Tgf4oLbgwd5JQaHyKVQZU9UdGE0w5rtDsrZzfUaomLo
*Disclosure:
  WyIyR0xDNDJzS1F2ZUNmR2ZyeU5STj13IiwgImdpdmVuX25hbWUiLCAiSm9obiIg
  *Contents: ["eluV50g3gSNII8EYnsxA_A", "family_name", "Doe"]
Claim email:

*SHA-256 Hash: JzYjH4svliH0R3PyEMfeZu6Jt69u5qehZo7F7EPY1SE
*Disclosure:
  WyJ2SWo3dE0tYVpVLBHym9TNXRtdlZBIiwgImVtYWl5iisIiwgImpvaG5kb2VA
  ZXhhbXBsZS55jb20iXQ
*Contents: ["6Ij7tM-a51VPGboS5tmvVA", "email", 
  "johndoe@example.com"]

Claim phone_number:

*SHA-256 Hash: PorFbpKuVu6xymJagvkFsFXAbRoc2JGlAUA2BA4o7CI
*Disclosure:
  WyJlSThaV205Uw5LUHB0UGV0ZW5lZGhhIRiwigInBob25lX251bWJlciIsICIr
  MS0yMDINTU1LTAXmDEiXQ
*Contents: ["eI8Zwm9QnKPPnPEnHdhQ", "phone_number", 
  "+1-202-555-0101"]

Claim address:

*SHA-256 Hash: IlDzIKeiZdDwppqPK6ZfbyphFvz5FgnWa-sN6wqXCiw
*Disclosure:
  WyJRZ19PnjR6cUF4ZTQxMmExMDhpcm9BIiwgImFkZHJlc3MiLCB7InN0cmV1
  dF9hZGRyZXNzIjogIjEyMyBNYWluIFN0IiwgImxvY2FsaXR5IjogIkFueXRv
  d24ilCaiicmVnaw9uIljogIkFueXN0YXRlIiwgImNvdw50cnki01AiVMiFV0
*Contents: ["Qg_O64zqAxe412a108iroA", "address", 
  {"street_address": "123 Main St", "locality": "Anytown", "region": "Anystate", 
  "country": "US"}]

Claim birthdate:

*SHA-256 Hash: jdrTE8YcbY4EifugihiAe_BPeKxJQZICeiUQwY9QxI
*Disclosure:
  WyJBsngtMDk1V1BycFR0TjRRTU9xUK9BIiwgImJpcnRoZGF0ZSIiCIxOTQw
  LTAXLTAxIl0
*Contents: ["AJx-095VPrtTn4QMOqROA", "birthdate", "1940-01-01"]

Claim is_over_18:

*SHA-256 Hash: 09vKrJMOlyTWM0sjpu_pd0BVBQ2M1y3KhpH515nXkpY
*Disclosure:
  WyJQYybMzSk0yTGNoY1VfEhnZ3ZfdwZRIiwgImlzlzX292ZXJfMTgiLCB0cnV1
  XQ
*Contents: ["Pc33JM2LchUu_lHggv_uFQ", "is_over_18", true]

Claim is_over_21:

*SHA-256 Hash: 2rsjGbaC0ky8mT0pJrPioWTq0_daw1sX76poUlgCwbI
The SD-JWT and the Disclosures would then be serialized by the Issuer into the following format for issuance to the Holder:

ejbGciOiAiRmVNYTljaHRwIGF1Z2UgaHR0cDovL2RvZ3VhZHkuY29t

The SD-JWT and the Disclosures would then be serialized by the Issuer into the following format for issuance to the Holder:
3.4. Verification and Processing

The recipient (Holder or Verifier) of an SD-JWT VC MUST process and verify an SD-JWT VC as described in Section 8 of [I-D.ietf-oauth-selective-disclosure-jwt].

If Key Binding is required (refer to the security considerations in Section 11.6 of [I-D.ietf-oauth-selective-disclosure-jwt]), the Verifier MUST verify the Key Binding JWT according to Section 8 of [I-D.ietf-oauth-selective-disclosure-jwt]. To verify the Key Binding JWT, the cnf claim of the SD-JWT MUST be used.

Furthermore, the recipient of the SD-JWT VC MUST validate that the public verification key for the Issuer-signed JWT as defined in Section 3.5.

If there are no selectively disclosable claims, there is no need to process the _sd claim nor any Disclosures.

If status is present in the verified payload of the SD-JWT, the status SHOULD be checked. It depends on the Verifier policy to reject or accept a presentation of a SD-JWT VC based on the status of the Verifiable Credential.

Any claims used that are not understood MUST be ignored.

Additional validation rules MAY apply, but their use is out of the scope of this specification.

3.5. Issuer-signed JWT Verification Key Validation

A recipient of an SD-JWT VC MUST apply the following rules to validate that the public verification key for the Issuer-signed JWT corresponds to the iss value:

*JWT VC Issuer Metadata: If a recipient supports JWT VC Issuer Metadata and if the iss value contains an HTTPS URI, the recipient MUST obtain the public key using JWT VC Issuer Metadata as defined in Section 5.
*X.509 Certificates: If the recipient supports X.509 Certificates, the recipient MUST obtain the public key from the leaf X.509 certificate defined by the x5c JWT header parameters of the Issuer-signed JWT and validate the X.509 certificate chain in the following cases:

- If the iss value contains a DNS name encoded as a URI using the DNS URI scheme [RFC4501], the DNS name MUST match a dNSName Subject Alternative Name (SAN) [RFC5280] entry of the leaf certificate.
In all other cases, the iss value MUST match a uniformResourceIdentifier SAN entry of the leaf certificate.

DID Document Resolution: If a recipient supports DID Document Resolution and if the iss value contains a DID [W3C.DID], the recipient MUST retrieve the public key from the DID Document resolved from the DID in the iss value. In this case, if the kid JWT header parameter is present, the kid MUST be a relative or absolute DID URL of the DID in the iss value, identifying the public key.

Separate specifications or ecosystem regulations MAY define rules complementing the rules defined above, but such rules are out of scope of this specification. See Section 6.2 for security considerations.

If a recipient cannot validate that the public verification key corresponds to the iss value of the Issuer-signed JWT, the SD-JWT VC MUST be rejected.

4. Presenting Verifiable Credentials

This section defines encoding, validation and processing rules for presentations of SD-JWT VCs.

4.1. Key Binding JWT

If the presentation of the SD-JWT VC includes a Key Binding JWT, the Key Binding JWT MUST adhere to the rules defined in Section 5.3 of [I-D.ietf-oauth-selective-disclosure-jwt].

The Key Binding JWT MAY include additional claims which, when not understood, MUST be ignored by the Verifier.

4.2. Examples

The following is a non-normative example of a presentation of the SD-JWT shown above including a Key Binding JWT:
In this presentation, the Holder provides only the Disclosure for the claim address. Other claims are not disclosed to the Verifier. The following example shows a presentation of a (different) SD-JWT without a Key Binding JWT:

```
eyJhbGciOiAiRVMyNTYiLCAidHlwIjogInZjK3NkLWp3dCIsICJraWQiOiAiZG9jLXNpZ25lci0wNS0yNS0yMDIyIn0.eyJfc2QiOiBbIjA5dktySk1PbHluUV00wcvp2dV9wZE9CVK9RMaixTENLhAYBi1hrcfK1iCA1Mn3zakdiYUwma3k4bVQwEpyUj1vM1RxMF9kYXcxc1g3NnBvWxw7Q3diS1ISICJrF0a84ZGhXMGRIRupid1VieEVfKn1dUM5dVJFTE9pZUXaaGg3WGVJHVRBIiwgIKl5rShpJS2VpWmREd3BxcEs2WmZieXBOmZ6NUZndlhdLXN0NnduXVhDxcilCA1SnzpZagU9c3Zsaugw7UjNqevUNZmVadT3KzDy5dTXZwaabzdGN6VQWWxTRSisiCJq3JGYnBLdVZ1NhshbUphZ32runNGWEFiUm9jMkphEBFtCQTRVN2NJIIwgi1RHZjRvTGlnd2Q1ISIhSH1LVfaVT1VZEDFMHc1cnREc3JaeMVYVW9tTG8iLCAiamRyVEU4WwNiTRFaWZ12loaUF1X0JQZw4SlFaSUNlaVVRvd1k5UXF4SSisICJqc3U5eVZ1bHdRUXwoRmxNNKbHPNYVNGemdsaFFHMERWzmF5UXdMVUs0I10sICJpc3M10iaAiHR0cHM6Ly9leGFtcGxlLmNvbS9pZGVudGl0eV9jcmVkZw50aWFscy5leGFtcGxlLmNvbS9pZGVudGl0eV9jcmVkZw50aWFscy5leGFtcGxlLmNvbS9pZGVudGl0eV9jcmVkZw50aWFscy5leGFtcGxlLmNvbS9pZGVudGl0eV9jcmVkZw50aWFscy5leGFtcGxl.
```

In this claim address, the Holder has disclosed only the Disclosure for the claim address. Other claims are not disclosed to the Verifier.
5. JWT VC Issuer Metadata

This specification defines the JWT VC Issuer Metadata to retrieve the JWT VC Issuer Metadata configuration of the Issuer of the SD-JWT VC. The Issuer is identified by the iss claim in the JWT. Use of the JWT VC Issuer Metadata is OPTIONAL.

Issuers publishing JWT VC Issuer Metadata MUST make a JWT VC Issuer Metadata configuration available at the location formed by inserting the well-known string "/.well-known/jwt-vc-issuer" between the host component and the path component (if any) of the iss claim value in the JWT. The iss MUST be a case-sensitive URL using the HTTPS scheme that contains scheme, host and, optionally, port number and path components, but no query or fragment components.

5.1. JWT VC Issuer Metadata Request

A JWT VC Issuer Metadata configuration MUST be queried using an HTTP GET request at the path defined in Section 5.

The following is a non-normative example of an HTTP request for the JWT VC Issuer Metadata configuration when iss is set to https://example.com:

GET /.well-known/jwt-vc-issuer HTTP/1.1
Host: example.com

If the iss value contains a path component, any terminating / MUST be removed before inserting /.well-known/ and the well-known URI suffix between the host component and the path component.
The following is a non-normative example of a HTTP request for the JWT VC Issuer Metadata configuration when iss is set to https://example.com/tenant/1234:

GET /well-known/jwt-vc-issuer/tenant/1234 HTTP/1.1
Host: example.com

5.2. JWT VC Issuer Metadata Response

A successful response MUST use the 200 OK HTTP and return the JWT VC Issuer Metadata configuration using the application/json content type.

An error response uses the applicable HTTP status code value.

This specification defines the following JWT VC Issuer Metadata configuration parameters:

*issuer REQUIRED. The Issuer identifier, which MUST be identical to the iss value in the JWT.

*jwks_uri

- OPTIONAL. URL string referencing the Issuer's JSON Web Key (JWK) Set [RFC7517] document which contains the Issuer's public keys. The value of this field MUST point to a valid JWK Set document.

*jwks

- OPTIONAL. Issuer's JSON Web Key Set [RFC7517] document value, which contains the Issuer's public keys. The value of this field MUST be a JSON object containing a valid JWK Set.

JWT VC Issuer Metadata MUST include either jwks_uri or jwks in their JWT VC Issuer Metadata, but not both.

It is RECOMMENDED that the JWT contains a kid JWT header parameter that can be used to look up the public key in the JWK Set included by value or referenced in the JWT VC Issuer Metadata.

The following is a non-normative example of a JWT VC Issuer Metadata configuration including jwks:
The following is a non-normative example of a JWT VC Issuer Metadata configuration including jwks_uri:

```
{
    "issuer":"https://example.com",
    "jwks":{
        "keys": [
            {
                "kid":"doc-signer-05-25-2022",
                "e":"AQAB",
                "n":"nj3Y3wsLUF19BmpAbk0swCNVx17Eh9wMO-_AReZwBqfaWbCFH
HRZXsIV2VMCNVU8Tp4obUaSXcRcQ-VMsfQPJm9IzgtRdAY8NN8Xb7PECYy
l8jvTTvPpbziapzUXNDFlAo0krIo13WmflPUUgMKULBNOEdifpO70p
RM0r1p_gg_WNUKoW1V-3keyUJoX9NztEDm_D2MQXj9eGOJJ8yPgL8PAZMLe
2R7jb9Tx0CPDED7tY_TU4nFP1xptw59A42mldEmViXsKQt60s1SLboazxFKve
qXC_jpLULt22OC6UGU63p-REw-Z0r3r845z50wMuziFqRM9bQ",
                "kty":"RSA"
            }
        ]
    }
}
```

```
{
    "issuer":"https://example.com",
    "jwks_uri":"https://jwt-vc-issuer.example.org/my_public_keys.jwks"
}
```

Additional JWT VC Issuer Metadata configuration parameters MAY also be used.

5.3. JWT VC Issuer Metadata Validation

The issuer value returned MUST be identical to the iss value of the JWT. If these values are not identical, the data contained in the response MUST NOT be used.

6. Security Considerations

The Security Considerations in the SD-JWT specification [I-D.ietf-oauth-selective-disclosure-jwt] apply to this specification. Additionally, the following security considerations need to be taken into account when using SD-JWT VCs:

6.1. Server-Side Request Forgery

The JWT VC Issuer Metadata configuration is retrieved from the JWT VC Issuer by the Holder or Verifier. Similar to other metadata endpoints, the URL for the retrieval MUST be considered an untrusted
value and could be a vector for Server-Side Request Forgery (SSRF) attacks.

Before making a request to the JWT VC Issuer Metadata endpoint, the Holder or Verifier MUST validate the URL to ensure that it is a valid HTTPS URL and that it does not point to internal resources. This requires, in particular, ensuring that the host part of the URL does not address an internal service (by IP address or an internal host name) and that, if an external DNS name is used, the resolved DNS name does not point to an internal IPv4 or IPv6 address.

When retrieving the metadata, the Holder or Verifier MUST ensure that the request is made in a time-bound and size-bound manner to prevent denial of service attacks. The Holder or Verifier MUST also ensure that the response is a valid JWT VC Issuer Metadata configuration document before processing it.

Additional considerations can be found in [OWASP SSRF].

6.2. Ecosystem-specific Public Key Verification Methods

When defining ecosystem-specific rules for the verification of the public key, as outlined in Section 3.5, it is critical that those rules maintain the integrity of the relationship between the iss value within the Issuer-signed JWT and the public keys of the Issuer.

It MUST be ensured that for any given iss value, an attacker cannot influence the type of verification process used. Otherwise, an attacker could attempt to make the Verifier use a verification process not intended by the Issuer, allowing the attacker to potentially manipulate the verification result to their advantage.

7. Privacy Considerations

The Privacy Considerations in the SD-JWT specification [I-D.ietf-oauth-selective-disclosure-jwt] apply to this specification. Additionally, the following privacy considerations need to be taken into account when using SD-JWT VCs.

7.1. Unlinkability

The Privacy Considerations in Section 12.5 of [I-D.ietf-oauth-selective-disclosure-jwt] apply especially to the cnf claim.

7.2. Verifiable Credential Type Identifier

Issuers and Holders have to be aware that while this specification supports selective disclosure of claims of a given SD-JWT VC, the
vct claim is not selectively disclosable. In certain situations this could lead to unwanted leakage of additional context information.

In general, Issuers are advised to choose vct values following data minimization principles. For example, government Issuers issuing an SD-JWT VC to their citizens to enable them to prove their age, might consider using a vct value that does not allow third-parties to infer additional personal information about the Holder, e.g., country of residency or citizenship.

Additionally, Holders have to be informed that, besides the actual requested claims, the vct information is shared with the Verifier.

7.3. Issuer Phone-Home

A malicious Issuer can choose the Issuer identifier of the SD-JWT VC to enable tracking the usage behavior of the Holder if the Issuer identifier is Holder-specific and if the resolution of the key material to verify the Issuer-signed JWT requires the Verifier to phone home to the Issuer.

For example, a malicious Issuer could generate a unique value for the Issuer identifier per Holder, e.g., https://example.com/issuer/holder-1234 and host the JWT VC Issuer Metadata. The Verifier would create a HTTPS GET request to the Holder-specific well-known URI when the SD-JWT VC is verified. This would allow the malicious Issuer to keep track where and how often the SD-JWT VC was used.

Verifiers are advised to establish trust in an SD-JWT VC by pinning specific Issuer identifiers and should monitor suspicious behaviour such as frequently rotating Issuer identifiers. If such behaviour was detected, Verifiers are advised to reject SD-JWT VCs issued by such Issuers.

Holders are advised to reject SD-JWT VCs if they contain easily correlatable information in the Issuer identifier.

8. Relationships to Other Documents

This specification defines validation and processing rules for verifiable credentials using JSON payloads and secured by SD-JWT [I-D.ietf-oauth-selective-disclosure-jwt]. Other specifications exist that define their own verifiable credential formats; for example, W3C Verifiable Credential Data Model (VCDM) 2.0 [W3C.VCDM] defines a data model for verifiable credentials encoded as JSON-LD, and ISO/IEC 18013-5:2021 [ISO.18013-5] defines a representation of verifiable credentials in the mobile document (mdoc) format encoded as CBOR and secured using COSE.
9. References

9.1. Normative References


9.2. Informative References


[ISO.18013-5]  


Appendix A. IANA Considerations

A.1. JSON Web Token Claims Registration

*Claim Name: "vct"
*Claim Description: Verifiable credential type identifier
*Change Controller: IETF
*Specification Document(s): [[ Section 3.2.2.1.1 of this of this specification ]]

A.2. Media Types Registry

A.2.1. application/vc+sd-jwt

The Internet media type for a SD-JWT VC is application/vc+sd-jwt.

*Type name: application
*Subtype name: vc+sd-jwt
*Required parameters: n/a
*Optional parameters: n/a
Encoding considerations: 8-bit code points; SD-JWT VC values are encoded as a series of base64url-encoded values (some of which may be the empty string) separated by period ('.') and tilde ('~') characters.


Interoperability considerations: n/a

Published specification: [[ this specification ]]

Applications that use this media type: Applications that issue, present, and verify SD-JWT-based verifiable credentials.

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: Oliver Terbu oliver.terbu@mattr.global

Intended usage: COMMON

Restrictions on usage: none

Author: Oliver Terbu oliver.terbu@mattr.global

Change controller: IETF

A.3. Well-Known URI Registry

This specification requests the well-known URI defined in Section 5 in the IANA "Well-Known URIs" registry [IANA.well-known] established by [RFC5785].

A.3.1. Registry Contents

-URI suffix: jwt-vc-issuer
-Change controller: IETF
-Specification document: [[ Section 5 of this of this specification ]]
-Related information: (none)

Appendix B. Examples

Important: The following examples are not normative and provided for illustrative purposes only. In particular, neither the structure of the claims nor the selection of selectively disclosable claims are normative.

Line breaks have been added for readability.

B.1. Example 1: Person Identification Data (PID) Credential

This example shows how the artifacts defined in this specification could be used to represent the concept of a Person Identification Data (PID) [EUDIW.ARF] using the data of a German citizen.
Key Binding is applied using the Holder's public key passed in a cnf claim in the SD-JWT.

The Issuer is using the following input claims set:

```json
{
  "vct": "https://bmi.bund.example/credential/pid/1.0",
  "given_name": "Erika",
  "family_name": "Mustermann",
  "birthdate": "1963-08-12",
  "source_document_type": "id_card",
  "address": {
    "street_address": "Heidestraße 17",
    "locality": "Köln",
    "postal_code": "51147",
    "country": "DE"
  },
  "nationalities": [
    "DE"
  ],
  "gender": "female",
  "birth_family_name": "Gabler",
  "place_of_birth": {
    "locality": "Berlin",
    "country": "DE"
  },
  "also_known_as": "Schwester Agnes",
  "age_equal_or_over": {
    "12": true,
    "14": true,
    "16": true,
    "18": true,
    "21": true,
    "65": false
  }
}
```

The following is the issued SD-JWT:
The following payload is used for the SD-JWT:

```
{
  "sd": [
    "0HzmnsIPz337kSWe7C341--88gzJi-eBJ2Vz_HJwATg",
    "9zbp1C7TdeW7qal6BBZ1MtqjDmdE0iXevdJloXVjDrQ",
    "IO0tfCUoDXCup5yy2uujQssDVGaWNiULiNZ_awD0gc",
    "IEBYsJGhX1rQo58yXm2Zx3yl19Z1TtToPo17QPiY",
    "Lai6IU6d7GGqagXR7AvGRmTrnXgSld3z8Eig_fV3f0Z1Wg",
    "hvDxhwm6cQiBSB2a0tjuLAcwAMpDsau0nkvK0qWNE",
    "ikuurQ4k8v3CyA7d-c-nJzZBkReDTU-C4gniTE7OTU",
    "qvZNLj2vh9o4SEXOFMlYDuvTykdsWCNg0wTdlr0Aeim",
    "wzW15bhCkxksxVvuJ8RF3xi8641n1jo_76BC2oa1ug",
    "Z0eBxhvm4SzmQcLkxKuEAO6GBy0qa1z2IoVX_YDQ"
  ],
  "iss": "https://example.com/issuer",
  "iat": 1683000000,
  "exp": 1883000000,
  "vct": "https://bmi.bund.example/credential/pid/1.0",
  "age_equal_or_over": {
    "sd": [
      "BwjAYeZetfPLeXhclC8hYziIYfPwqCyEYcbHci0bBzo",
      "C8qi_--1QN0bibkbX8UVQWpu7VF3uruU7guukReehZA",
      "DVv9Pz-iaEz9KqT-olIIuHxrvToOBzDj27Ev79gE",
      "FV_93e13CP9bmvE1j7hc8aNq-iqQPQBkVSnqrwF474f4",
      "lnlmLPMEWkBwutuBBPgy0vDfnqG4Z0feqXmnhubwIQ",
      "Lu9Fz75Qexj7n6zyv-MgPNjsRELws5mH7mQW1N4Qok"
    ]
  },
  "sd_alg": "sha-256",
  "cnf": {
    "jwk": {
      "kty": "EC",
      "crv": "P-256",
      "x": "TCAER19Zvu30HF4j4W4vfSVoHIP1IiDlDs7vCeGmc",
      "y": "ZxjiWwb2MqGHVwKVQ4hbSIirsFvuEC6t4jT9F2HZQ"
    }
  }
}
```

The following Disclosures are created by the Issuer:

**Claim given_name:**

*SHA-256 Hash: 0HzmnsIPz337kSWe7C341--88gzJi-eBJ2Vz_HJwATg*

*Disclosure:
WyIyR0xDNDJzS1F2ZUNmR2ZyeU5STj13IiwgImdpdmVuX25hbWUilCAiRXJp a2EiXQ*
Claim family_name:

SHA-256 Hash: I00fcFUoDXCucp5yy2ujqPssDVGawNiUliNz_awD0gc
*Disclosure:
Wyj1bHVWNu9mMz2TTk1j0EVEZbnN4QV9BIiwgImZhbWlseV9uYW1iIiwgIk11c3Rlc3hmbm4iXQ
*Contents: ["eluV5Og3gSNII8EYnsx_A_A", "family_name", "Mustermann"]

Claim birthdate:

SHA-256 Hash: Lai6IU6d7GQagXr7AvGTnXgSld3z8EIG_fv3f0Z1Wg
*Disclosure:
WyI2SWo3dE0tYTVpV1BHym9TNXRtd1ZBIiwgImJpcnRoZGF0ZSIIsIClxOTYzLTA4LTEyI10
*Contents: ["6Ij7tM-a5iVPGboStmvVA", "birthdate", "1963-08-12"]

Claim source_document_type:

SHA-256 Hash: qvzNLj2vh9o4SEX0fMiYDuvTykdsWCNg0wTdlr0AEIIM
*Disclosure:
Wyj1St4aV205UW5LUHBOUGV0ZW5IZGhRIiwgInNvdXjzV9kb2N1bWVudF90eXB1IiwgImlkX2NhcmQiXQ
*Contents: ["ei8ZWm9QnKppNPeNenHdhQ", "source_document_type", "id_card"]

Claim street_address:

SHA-256 Hash: bd1EvzgNopUk4EW3_eQ2n3_NU4iuXOHv9XbGHN3g1TE
*Disclosure:
WyJRZ19PnjR6cUF42TQxMmExMDhcsm9BIiwgInN0cmVldF9hZGRyZXNzIiwgIkhlaWRLc3ryYVx1MDBkZmUgMTciXQ
*Contents: ["Qg_064zqAxe412a108iroA", "street_address", "Heidestra\u00dfe 17"]

Claim locality:

SHA-256 Hash: f_FQYgvQWvyVqNnIXsARlNye7YGp8E77gRAjaq-wvnw
*Disclosure:
WyjBSngtMDk1V1bysCR0TjRRTu9xUK9BIiwgImxvY2FsARX5IiwgIktcdTAwZjZsb1Jd
*Contents: ["Ajx-095VPpTtN4QMOqR0A", "locality", "K\u007fu00f61n"]

Claim postal_code:

SHA-256 Hash: XFc7zXPm7zzVdMywm2EuBf1ka5HHqv8Up_szNGqvig
*Disclosure:
WyjQYzMsSk0yTGN0Y1VfbEhnZ3ZfdWZRIiwgInBvc3Rhb9j2RlIiwgIjUXMTQ3I10
Contents: ["Pc33JM2LchcU_lHggv_ufQ", "postal_code", "51147"]

Claim country:

*SHA-256 Hash: v4kkb_pP1jlvUbStjyc5bqcWyA-i8XLvhVYY7ZT0tb0
*Disclosure:
WyJHMDJOU3JZRzmpGWE3SW8wWN5YWpB1wgImNvW50cnkiLCAiREUixQ
*Contents: ["G02NSrQfjFXQ7Io09syajA", "country", "DE"]

Claim address:

*SHA-256 Hash: z0eBXhxvIS4ZzmQcLLxKuEAOGBbyjOqa1z2IoVx_YDQ
*Disclosure:
WyJsax2Xr4jVqTVlsR1RQVw92TU5JdkNBIwImFkZHJlc3MiLCB7I9zZCI6
IFSiWEZJn3pYUG03enpwZEi5d20yRXVZmrxrTVISHF2ZjhyVC9Fz6k5HcXZp
ZYIsIClJZDFVvnpmTm9vW0s0Rcvxz2VRMm4zX05VG1lWE9IdjlYYkdITJnN
MVRFIwImZfR1FZZ3ZRV3Z5VnF0bk1Yc0FSbE55ZTdrZ3A4RTc3Z1JBamFx
LXd2bnciLCAidjJra2JfcFAxamx2VWJTAteR5YzVicWNEUEtaThYTHZoV1LZ
N1pUMHRiMCJdfV0
*Contents: ["lk1xF5jMY1GTPUovMNIvCA", "address", {"_sd": [
"XFc7zXpm7zzVDMywms2EuBF1ka5HqfV8Up_zNQvig",
"bd1EzgNvpUK4E3_eQ2n3_NU4iuXOHv9xGHN3g1TE",
"f_FQygvQWvYvQnnIXsAR1Nye7Y6p8E77gRAjaq-wvuw",
"v4kkb_pP1jlvUbStjyc5bqcWyA-i8XLvhVYY7ZT0tb0"}]

Claim nationalities:

*SHA-256 Hash: hvDXhwmGcJQsBCA2otjuLAcwAMpDsau0nkovcK0qWNE
*Disclosure:
WyJuUHVVUw5RukZXm0JZUFTn0FuWEZBIwIm5hdGlVbmFsaXRpZXMiLCBb
IkRF11d
*Contents: ["nPuoQnkRFq3BIeAm7AnXFA", "nationalities", ["DE"]]

Claim gender:

*SHA-256 Hash: IEBYSJGNhXIrlR0q58ykXm2Zx3yll9Z1TtToPo17QQiY
*Disclosure:
WyI1Y1BzMU1xdVpOYTBoa2FGenp6Wk53IiwImd1bmRlcjISICJmZW1hbGU
XQ
*Contents: ["5bPs1IquZNa0hkaFzzzZNw", "gender", "female"]

Claim birth_family_name:

*SHA-256 Hash: ikuur8Q4k8q3VcyA7dC-mNjZBkReDTU-CG4niTE70TU
*Disclosure:
WyI1YTJXMF90cmxFWnmpcmW1rXzdQs13IiwImJpcoR0XZhbwlseV9uYW1l
Iiw1kdhYmx1c1jd
*Contents: ["5a2W0_Nr1EZzfqmk_7Pq-w", "birth_family_name", "Gabler"]
Claim locality:

*SHA-256 Hash: WphHoIDyoWbApDC4zbuwR40xl0xLhDC_cv4tsS71rEA
*Disclosure: WyJ5MXNWVTv3ZGZKYWhwZGd3UGdTN1JRIiwgImxvY2FsaXR5IiwigIkjlcmxpbiJd
*Contents: ["y1sVUSwdfJahVdgwPgS7RQ", "locality", "Berlin"]

Claim place_of_birth:

*SHA-256 Hash: wzW15bhCkvksxVvuJ8RF3xi8i64ln1jo_76BC2oa1ug
*Disclosure: WyJIYi0WDhzclXZM1FeE65JSmRxewU9BIiwgInBsYWlX29mX2JpcnRoIiwg
  eyJfc2QioiBbIldwaEhvSUR5b1diQXBEQzR6YnV3UjQweGwweExoRENFy3Y0dHNTNzFyRUEiXSwgImNvdW50cnkiOAiREuifV0
*Contents: ["HbQ4X8srvW3QDxnIjdqy0A", "place_of_birth", {"_sd": ["WphHoIDyoWbApDC4zbuwR40xl0xLhDC_cv4tsS71rEA"], "country": "DE"}]

Claim also_known_as:

*SHA-256 Hash: 9ZbplC7TdEW7qa16BBZ1MtqJdmeEOixevdJl0XVjdRQ
*Disclosure: WyJDOUDtB3VqdmlKcXVFZ1lmmb2pDYjFBIiwgImFsc29fa25vd25fYXMiLCAi
  U2Nod2VzdGVyeFnbmVzIl0
*Contents: ["C9GSoujviJquEgYfojCb1A", "also_known_as", "Schwester Agnes"]

Claim 12:

*SHA-256 Hash: DVv9Px-i_aEZz9KqT-011IuHxrVtoDG0zDj27Eev7gE
*Disclosure: WyJreDVrRjE3Vi14MEptd1V4OXZndnR3IiwgMTIsIHRydWVd
*Contents: ["kx5kf17V-x0JmwUx9vgvtw", 12, true]

Claim 14:

*SHA-256 Hash: BwjAYezetfPLeXhc1C8hYzIIYfpwqCyEYcbHci0bBzo
*Disclosure: WyJiM28xdXN3UDc3M5ZpMn1lR2RwQ0VRiIwgMTQsIHRydWVd
*Contents: ["H3o1uswP760Fi2yeGdVCEQ", 14, true]

Claim 16:

*SHA-256 Hash: InImLpMEWkBwtuBBPpgyx0vDfkqG4Z0feqXmnhubwIQ
*Disclosure: WyJPQktsVFZsdkxnLUFkd3FZR2J0QFpBIiwgMTYsIHRydWVd
*Contents: ["OBK1TV1vlG-AdwgYGbP8ZA", 16, true]
Claim 18:

*SHA-256 Hash: FV_93eI3CP9bnvEIj7hc8aNq-1qpQBkVSnqrFw474f4
*Disclosure: 
  WyJNMEpiNTd0NDF1YnJrU3V5ckRUM3hBIiwgMTgsIHRydWVd
*Contents: ["M0Jb57t41ubrkSuyrDT3xA", 18, true]

Claim 21:

*SHA-256 Hash: C8qi_-1QN0bibkbX8UVQUwPz7VF3uruU7guukReehZA
*Disclosure: 
  WyJEc210S05ncFY0ZEFIcGpyY2Fvc0F3IiwgMjEsIHRydWVd
*Contents: ["DsmtKNgpV4dAHpjrc aosAw", 21, true]

Claim 65:

*SHA-256 Hash: Lu9Fz75Qexj7n6zyv-MgPNjsRELwsS5mH7mQW1Nm4Qok
*Disclosure: 
  WyJlSzVvNXBIZmd1cFBwbHRqMXFoQUp3IiwgNjUsIGZhbHNlXQ
*Contents: ["eK5o5pHfgu Pp1tjqhAJw", 65, false]

The following shows a presentation of the SD-JWT with a Key Binding JWT that discloses only the nationality of the Holder:
The following is the payload of a corresponding Key Binding JWT:

```json
{
  "nonce": "1234567890",
  "aud": "https://example.com/verifier",
  "iat": 1709062487,
  "sd_hash": "F29Z6xOvk9X2bwpHMwrgvrd_Kdj0LrEHzdrCstAw50rY"
}
```

After the validation, the Verifier will have the following data for further processing:

```
eyJhbGciOiAiRVMyNTYiLCAidHlwIjogImt3dCIJ9.eyJub25jZSI6ICIxMjM0NTY3ODkwIiwgImlhdCI6IDE3MDkwNjI0ODcsICJzZF9oYXNoIjogIiM5YzN2YzU2IiwgIm5hdGlvbmFsaXRpZXMiLCBbIklXaiki
```

The following is a payload of a corresponding Key Binding JWT:

```json
{
  "nonce": "1234567890",
  "aud": "https://example.com/verifier",
  "iat": 1709062487,
  "sd_hash": "F29Z6xOvk9X2bwpHMwrgvrd_Kdj0LrEHzdrCstAw50rY"
}
```
Appendix C. Acknowledgements

We would like to thank Alen Horvat, Andres Uribe, Christian Bormann, Giuseppe De Marco, Michael Jones, Mike Prorock, Orie Steele, Paul Bastian, Torsten Lodderstedt, Tobias Looker, and Kristina Yasuda for their contributions (some of which substantial) to this draft and to the initial set of implementations.

Appendix D. Document History

-02

*Made specific rules for public verification key validation conditional
*Finetuned rules for obtaining public verification key
*Editorial changes
*added Brian Campbell as co-author
*Renamed JWT Issuer Metadata to JWT VC Issuer Metadata
*’iat’ is now optional and allowed to be selectively disclosable
*Fix inconstancy in the .well-known path construction
*Added registration request to IANA for the well-known URI
*Fix some formatting and text in the media type and JWT claim registration requests
*Clarify the optionality of the cnf claim
*Added relationships to other documents
*Added PID example

-01

*Introduce rules for type identifiers (Collision-Resistant Name)
*Rename type to vct
*Removed duplicated and inconsistent requirements on KB-JWT
*Editorial changes
*Added issuer public verification key discovery section.

-00

*Upload as draft-ietf-oauth-sd-jwt-vc-00
*Aligned terminology and descriptions with latest version of SD-JWT

[[ pre Working Group Adoption: ]]

-00

*Initial Version
*Removed W3C VCDM transformation algorithm
*Various editorial changes based on feedback
*Adjusted terminology based on feedback
*Added non-selectively disclosable JWT VC
*Added a note that this is not W3C VCDM

Authors' Addresses

Oliver Terbu
MATTR

Email: oliver.terbu@mattr.global

Daniel Fett
Authlete Inc.

Email: mail@danielfett.de

Brian Campbell
Ping Identity

Email: bcampbell@pingidentity.com