

Web Authorization Protocol
Internet-Draft
Intended status: Standards Track
Expires: 23 October 2026

G. Oliver
Google
21 April 2026

Delegate SD-JWT
draft-gco-oauth-delegate-sd-jwt-00

Abstract

This document specifies an extension to Selective Disclosure JSON Web Tokens to support further delegation from the Holder to a Delegate Holder. This is done by allowing the Key Binding JWT to also be an SD-JWT, optionally with its own Key Binding. This has particular applicability to Agentic systems.

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 23 October 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Feature Summary	3
2. Conventions and Definitions	4
2.1. Requirements Notation and Conventions	4
2.2. Terms and Definitions	4
3. Flow Diagram	5
4. Concepts	7
4.1. Selective Disclosure	7
4.2. Optional Key Binding	7
4.3. Verification	7
5. Formats	8
5.1. dSD-JWT and dSD-JWT+KB Data Formats	8
5.1.1. Compact Serialization	8
5.1.2. JSON Serialization	9
5.1.3. Delegate Payload Disclosure	9
5.1.4. KB-SD-JWT and KB-SD-JWT+KB	9
6. Verification	10
7. Delegation using Presentation	11
7.1. dSD-JWT Delegation using OpenID4VP	11
8. Security Considerations	12
8.1. Mandatory verification of delegate SD-JWT Chain	12
8.2. Delegation Policy	12
8.3. Delegate SD-JWT Revocation	12
9. Privacy Considerations	12
10. References	12
10.1. Normative References	12
10.2. Informative References	13
Appendix A. IANA Considerations	13
A.1. JSON Web Token Claims Registration	13
A.2. Media Type Registration	14
Appendix B. Acknowledgments	15
Author's Address	15

1. Introduction

This is an extension to SD-JWTs [RFC9901], to allow for the creation and delegation of new credentials using an existing SD-JWT+KB.

SD-JWT provides a mechanism for ensuring minimal disclosure in a three party model. This allows an intermediary party (the Holder) to choose to remove claims when only a subset is needed by a verifier. Additionally SD-JWT+KB allows for proof of possession by the Holder using the cnf claim. The Verifier need only trust the Issuer and its policy regarding the cnf key to trust the resulting presentation.

This is achieved using a composite structure of an Issuer-Signed JWT containing salted hashes and associated optional disclosures that can be omitted by the Holder. An optional KB-JWT containing transaction-specific data and signed by the private key cnf key is used to provide cryptographic key binding.

While it is possible for a Verifier to forward and further down-scope a SD-JWT, it is not able to prove that it was the recipient of the original SD-JWT+KB presentation, nor is it able to downscope parts of the transaction data, which may be relevant for privacy reasons.

This capability is useful for a 'delegation' model, where the Holder delegates further presentations of the SD-JWT+KB to a Delegate Holder. Additionally, the Holder should be able to provide additional (optionally disclosable) claims as part of the Delegation.

One example usecase is to delegate to an AI agent the ability to perform purchases on the users behalf, along with constraints on valid fulfillment conditions. The AI Agent can then prove to a merchant that it has been authorized to perform a purchase at this merchant on a Holders behalf, without revealing what other merchants may have been able to fulfill this purchase.

The approach taken here is to extend te KB-JWT so that it can be used as the start of a new SD-JWT or SD-JWT+KB, linking the resulting SD-JWT to the original one.

1.1. Feature Summary

This specification defines extensions to the SD-JWT and SD-JWT+KB formats:

1. dSD-JWT which is a composite structure consisting of an SD-JWT and a Key Binding SD-JWT. The KB-SD-JWT signs over a delegate JSON Payload with zero or more disclosures.
 - * This includes an alternative format for a KB-JWT (called a KB-SD-JWT) to secure a nested, selectively disclosable JSON object. This is achieved by making the KB-JWT an SD-JWT.
 - * A format for extending the SD-JWT+KB Compact Serialization to include a KB-SD-JWT.
 - * An alternative format to do the same for the SD-JWT JSON Serialization format.
2. dSD-JWT+KB which extends dSD-JWT to include Key Binding, allowing the Delegate Holder to prove proof of possession.
 - * This re-uses the KB-JWT mechanism specified by SD-JWT+KB for associating and presenting a proof of possession of a key pair.

- * This also allows for further delegation of the dSD-JWT+KB with the use of additional KB-SD-JWTs.

2. Conventions and Definitions

2.1. 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].

2.2. Terms and Definitions

This specification uses the terms "Disclosure", "Selectively Disclosable JWT (SD-JWT)", "Key Binding", "Key Binding JWT (KB-JWT)", "Selectively Disclosable JWT with Key Binding (SD-JWT+KB)" defined by [RFC9901].

Delegated SD-JWT(dSD-JWT): A composite structure consisting of an SD-JWT and a Key Binding SD-JWT (KB-SD-JWT). It acts as a chain of SD-JWTs where the Key Binding of the previous SD-JWT is used to sign the new one, allowing the Verifier to link presentations back to the initial Issuer.

Delegated SD-JWT with Key Binding (dSD-JWT+KB): An extension of the dSD-JWT that includes a final Key Binding JWT (KB-JWT). This allows the Delegate Holder to demonstrate proof of possession of the dSD-JWT.

Key Binding SD-JWT (KB-SD-JWT): An alternative format for a KB-JWT used to secure a nested, selectively disclosable JSON object (the Delegate Payload). It serves as the KB-JWT for the preceding SD-JWT in a chain.

Key Binding SD-JWT+KB (KB-SD-JWT+KB): A specific type of KB-SD-JWT where the typ parameter is set to "kb+sd-jwt+kb". This type requires the Delegate Payload to include a cnf claim.

Delegate Payload: A JSON object (which may be nested and selectively disclosable) over which the KB-SD-JWT signs. In a dSD-JWT+KB, this payload MUST include a cnf claim to establish the Delegate Holder's key.

Issuer: An entity that creates the initial SD-JWT.

Holder: An entity that receives the initial SD-JWT from the Issuer and has control of it. They may present the SD-JWT to a Verifier directly or delegate it to a Delegate Holder.

Delegate Holder: An intermediary entity to whom the original Holder delegates the SD-JWT. The Delegate Holder can perform further presentations or delegations of the SD-JWT.

Verifier: An entity that requests, checks and extracts the claims of the SD-JWT or dSD-JWT.

Delegation: Delegation in this document refers to a Holder or Delegate Holder presenting a credential to a Delegate Holder for the purpose of further presentation.

3. Flow Diagram

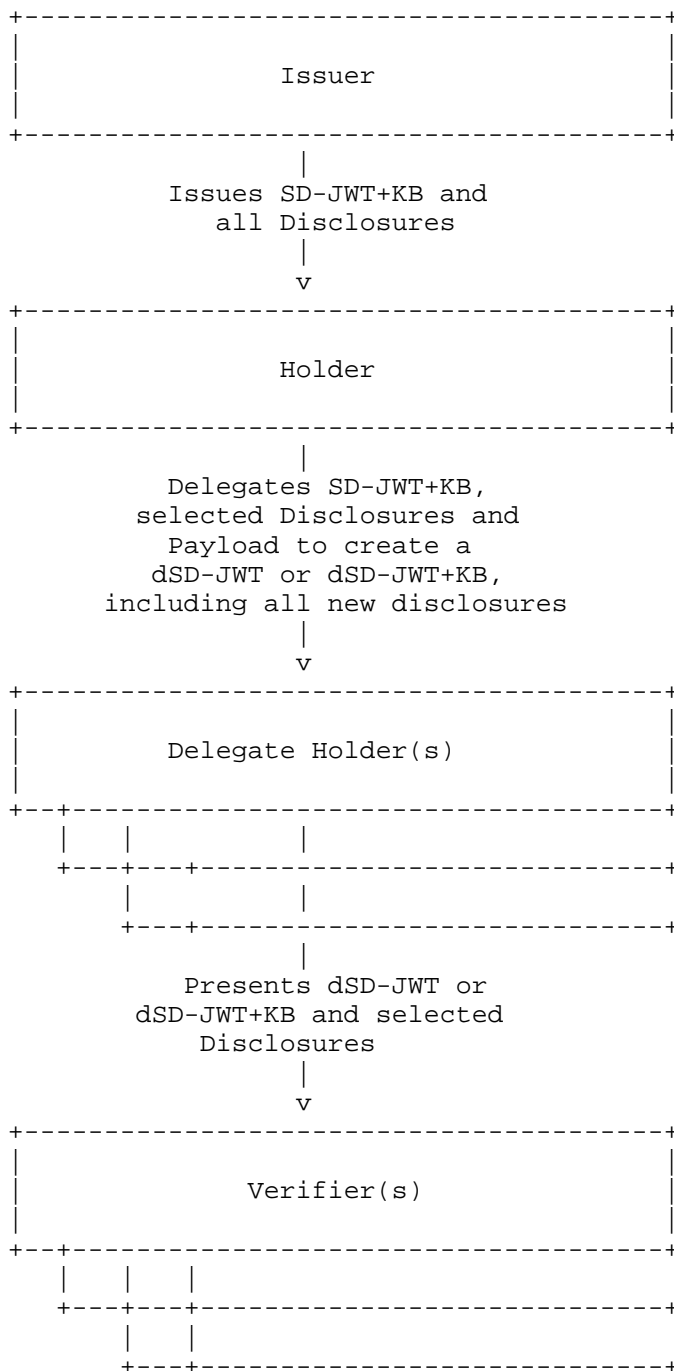


Figure 1: Delegate SD-JWT Issuance, Delegation and Presentation.

4. Concepts

At a high level, a dSD-JWT acts as a chain of SD-JWTs where the KB-JWT in the proceeding SD-JWT+KB fulfills the role of the Issuer-JWT for the next. This allows the Verifier to know the full chain that the dSD-JWT went through and link the presentations back to the initial Issuer while preserving the principles of Data Minimization.

4.1. Selective Disclosure

An dSD-JWT has two different sets of Selective Disclosures:

- * Selective Disclosures from the original SD-JWT
- * Selective Disclosures from the new Delegate Payload

A Delegate Holder may always choose to include or omit selective disclosures from the Delegate Payload. If the Holder wishes to allow the Delegate Holder to omit selective disclosures from the proceeding SD-JWT it MUST omit the `sd_hash` claim from the KB-JWT and include the `issuer_jwt_hash` claim instead.

4.2. Optional Key Binding

Keybinding is required for the initial SD-JWTs and all KB-SD-JWTs except the final one. It is optional for key binding to be used for the final KB-SD-JWT. When it is used, the Delegate Holder's public key information is included in the Delegate Payload of the final KB-SD-JWT. This allows the Delegate Holder to create a KB-JWT in future presentations using the private key associated with the included public key, demonstrating proof of possession.

4.3. Verification

At a high level verification works as follows:

- * The Verifier receives a dSD-JWT or dSD-JWT+KB from the Delegated Holder.
- * The Verifier splits the dSD-JWT into a chain with an SD-JWT, one or more KB-SD-JWT and (in the case of a dSD-JWT+KB) a final KB-JWT.
- * The Verifier verifies the first SD-JWT+KB using the issuer key, using the second KB-SD-JWT as the KB-JWT.
- * For each KB-SD-JWT, the Verifier verifies it as an SD-JWT+KB, treating the Delegate Payload as the JWT payload. The key from the `cnf` claim of the previous SD-JWT is used.

- * If required by policy, the final KB-SD-JWT MUST also be validated as an SD-JWT+KB, along with the transaction binding data included there.

5. Formats

5.1. dSD-JWT and dSD-JWT+KB Data Formats

A dSD-JWT with a single delegation is composed of:

- * A SD-JWT
 - i.e. An issuer-signed JWT
 - zero or more disclosures
- * A KB-SD-JWT
 - Serves as the KB-JWT for the proceeding SD-JWT
 - This is also the SD-JWT containing the Delegated Payload.

A dSD-JWT+KB with a single delegation is composed of:

- * A dSD-JWT
- * A KB-SD-JWT
- * A KB-JWT

A dSD-JWT or dSD-JWT+KB with multiple delegations is composed of:

- * A SD-JWT
- * Two or more KB-SD-JWTs
- * When the format is dDS-JWT-KB, a KB-JWT.

5.1.1. Compact Serialization

The dSD-JWT format extends the existing SD-JWT+KB format as follows:

<SD-JWT>~~~<KB-JWT>~<Delegated Disclosure 1>~...~<Delegated Disclosure M>~

The <KB-JWT> along with its disclosures is called a KB-SD-JWT.

The dSD-JWT+KB format further extends the dSD-JWT format by appending a delegate KB-JWT:

<SD-JWT>~~~<KB-SD-JWT>~<delegate KB-JWT>

This can be chained by replacing the KB-JWT with another KB-SD-JWT instead:

<SD-JWT>~~~<KB-SD-JWT 1>~...~<KB-SD-JWT n>~<delegate KB-JWT>

To process a dSD-JWT or a dSD-JWT+KB, the string is split on ~ as usual. The resulting array of components MUST have an empty component between the last disclosure of each SD-JWT before the following KB-SD-JWT. The following KB-SD-JWT is then used as the KB-JWT for the proceeding SD-JWT.

dSD-JWT+KB and dSD-JWT formats are differentiated by a trailing ~ for dSD-JWT.

5.1.2. JSON Serialization

For both the General and Flattened JSON Serialization, the dSD-JWT or dSD-JWT+KB is represented as a JSON object. The only change in encoding is to the format of the kb-jwt in the unprotected header, which now MUST conform to the general or flattened JSON serialization of an sd-jwt.

5.1.3. Delegate Payload Disclosure

The delegate payload disclosure is an Array disclosure, which is the base64urlencoding of [salt, JSON Object Payload].

The Disclosure Payload MUST follow all rules for the payload of the Issuer-signed JWT specified in [RFC9901] Section 4.1.

The Delegate Disclosures are created as per [RFC9901] Section 4.2.

The Delegate KB-JWT is a KB-JWT as per [RFC9901] Section 4.3 except that sd_hash, when present, is calculated over the proceeding SD-JWT or KB-SD-JWT and its associated disclosures.

5.1.4. KB-SD-JWT and KB-SD-JWT+KB

This specifies two new extensions to the KB-JWT. The following additional parameter is include:

- * “_delegate_payload_” : An array of JSON Objects. If it contains more than one element then they MUST all be replaced with disclosures. During the presentation of a dSD-JWT from a Delegate Holder to a Verifier exactly one of these MUST be disclosed.
 - When presenting from a Holder to a new Delegate Holder, multiple values being present allows for multiple dSD-JWTs to be delegated with a single signature which may be convenient when using a signing key that requires a user action per signing event.

KB-SD-JWTs MUST conform to all the requirements of a KB-JWT and an SD-JWT except as listed below:

KB-JWT changes:

- * The typ parameter value MUST be replaced with "kb+sd-jwt" for a KB-SD-JWT, and "kb+sd-jwt+kb" for a KB-SD-JWT+KB.
 - If the typ is KB-SD-JWT+KB then the Delegate Payload MUST include a cnf claim.
- * The sd_hash parameter is OPTIONAL. If it is not present it MUST instead include a issuer_jwt_hash parameter that hashes over only the proceeding Issuer-signed jwt or KB-SD-JWT+KB and not any disclosures.

When calculating the sd_hash it is calculated from the proceeding Issuer JWT or KB-SD-JWT+KB.

SD-JWT changes:

- * All claims that are expected to be found in the issuer-signed JWT Payload except the _sd_hash payload MUST instead be claims in the Delegate Payload.

6. Verification

To perform verification of an dSD-JWT or dSD-JWT+KB the following steps must be followed.

1. Split the dSD-JWT into its component SD-JWTs
 1. For a dSD-JWT this will be:
 - * A SD-JWT
 - * zero or more KB-SD-JWTs with typ "kb-sd-jwt+kb"
 - * one KB-SD-JWT with typ "kb-sd-jwt"
 2. For a dSD-JWT+KB this will be:
 - * A SD-JWT
 - * One or more KB-SD-JWTs with typ kb-sd-jwt+kb"
 - * One KB-JWT
2. Validate and process the initial SD-JWT according to [RFC9901] Section 7.1.
3. For each KB-SD-JWT except the final one:
 1. Validate and process it according to [RFC9901] Section 7.1
 - * The cnf claim of the proceeding component is used as the Issuer public key.
 - * Treat the Delegate Payload as the JWT Payload for finding all claims other than the _sd_alg claim.
 2. Verify that there is exactly one disclosed element in the delegate_payload array.
 3. If the sd_hash claim is present, calculate the digest over the proceeding SD-JWT or KB-SD-JWT and it's disclosures, as described in [RFC9901] Section 9.10. d. Otherwise, verify that the issuer_jwt_hash is present and matches the base64url

encoded digest of the proceeding Issuer signed JWT or KB-SD-JWT+KB. e. Verify the typ in the JWT Payload is "kb-sd-jwt+kb"

4. For the final KB-SD-JWT:
 1. Validate and process it according to 3.1 - 3.3
 2. If the credential is a dSD-JWT then the type MUST be "kb-sd-jwt" otherwise it MUST be "kb-sd-jwt+kb"
5. If the credential is a dSD-JWT+KB and Key Binding is required
 1. Follow section 7.3 step 5 to to verify the KB-JWT, using the final KB-SD-JWT to retrieve the Delegate Holder public key.
 2. If the sd_hash claim is present, calculate the digest over the proceeding SD-JWT or KB-SD-JWT+KB and it's disclosures as described in [RFC9901] Section 9.10.

If any of the steps fail then the presentation is invalid and processing MUST be aborted. Otherwise the list of processed SD-JWT and Delegate Payloads MAY be passed to the application to be used for their intended purpose.

7. Delegation using Presentation

Presentation mechanisms that allow specification of additional transaction data within the KB-JWT can be used to perform delegation. Below is a description of how this can be done using the OpenID4VP presentation protocol.

7.1. dSD-JWT Delegation using OpenID4VP

OpenId4VP [OIDF.OID4VP] specifies transaction_data within the request, which is included within the KB-JWT. To delegate an dSD-JWT or dSD-JWT+KB the transaction_type delegate MAY be used. The following additional parameters are included in the transaction_data object:

- * "format" : REQUIRED string containing either dSD-JWT or dSD-JWT+KB
- * "delegate_payload_disclosure" : REQUIRED String containing the Array Disclosure of the delegate payload.
- * "delegate_disclosures" : OPTIONAL Array of Strings containing the delegate_payload_disclosure

The delegate payload MUST NOT contain any disclosures not provided in delegate_disclosures. The KB-JWT includes the digest of the delegate_payload in the delegate_payload claim of the KB-JWT. _sd_hash may use any hash algorithm specified for hashing the transaction_data. The Wallet MAY include additional decoy digests.

Multiple delegate transaction_data MAY be included in the same request. In that case, each MUST have their digest included in the delegate_payload.

8. Security Considerations

Security Considerations as described in [RFC9901] also apply to delegate SD-JWTs. When the Holder is performing a delegation, they are acting as an Issuer of an SD-JWT and so all security considerations of an issuer apply to them.

8.1. Mandatory verification of delegate SD-JWT Chain

It is critical that the Verifier verifies each KB-SD-JWT in the chain and ensures that it is bound to the proceeding Issuer JWT or KB-SD-JWT with the sd_hash or issuer_jwt_hash. Without verifying the binding in both directions a malicious Delegate Holder may mis-match parts of the chain if the Holder cnf is reused.

8.2. Delegation Policy

An Issuer or Holder that wishes to limit delegation MAY include such constraints as visible claims in the Issuer signed JWT or KB-SD-JWT+KB.

8.3. Delegate SD-JWT Revocation

While traditional mechanisms of credential revocation can be used with delegate SD-JWTs, they present a practical challenge as, unlike a traditional Issuer, an individual Holder can not easily distribute revocation information to Verifiers.

Having a short exp and using claims to constraining the usage of the delegated SD-JWT limits this problem, as does cases where the Holder and the Delegate Holder are managed by the same entity.

9. Privacy Considerations

The privacy considerations in [RFC9901] Section 10 also apply to Delegate SD-JWTs.

10. References

10.1. Normative References

[OIDF.OID4VP]

Terbu, O., Lodderstedt, T., Yasuda, K., Fett, D., and J. Heenan, "OpenID for Verifiable Presentations 1.0", 9 July 2025, <https://openid.net/specs/openid-4-verifiable-presentations-1_0.html>.

[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>>.

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

[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>>.

[RFC9901] Fett, D., Yasuda, K., and B. Campbell, "Selective Disclosure for JSON Web Tokens", RFC 9901, DOI 10.17487/RFC9901, November 2025, <<https://www.rfc-editor.org/info/rfc9901>>.

10.2. Informative References

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

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

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

Appendix A. IANA Considerations

A.1. JSON Web Token Claims Registration

This specification requests registration of the following Claims in the IANA "JSON Web Token Claims" registry [IANA.JWT] established by [RFC7519].

* Claim Name: issuer_jwt_hash

- * Claim Description: Digest of the Issuer-JWT to which the KB-SD-JWT is tied
- * Change Controller: IETF
- * Specification Document(s): [[Section 5.1.4 of this specification]]

A.2. Media Type Registration

This section requests registration of the following media types [RFC2046] in the "Media Types" registry [IANA.MediaTypes] in the manner described in [RFC6838].

To indicate that the content is a Key Binding SD-JWT:

- * Type name: application
- * Subtype name: kb+sd-jwt
- * Required parameters: n/a
- * Optional parameters: n/a
- * Encoding considerations: binary; A Key Binding SD-JWT is a SD-JWT; SD-JWT values are a series of base64url-encoded values (some of which may be the empty string) separated by period ('.') and tilde ('~') characters.
- * Security considerations: See the Security Considerations section of [[this specification]] and [RFC9901].
- * Interoperability considerations: n/a
- * Published specification: [[this specification]]
- * Applications that use this media type: Applications utilizing a JWT based proof of possession mechanism with further selective disclosure.
- * Fragment identifier considerations: n/a
- * 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: Gareth Oliver, gco@google.com
- * Intended usage: COMMON
- * Restrictions on usage: none
- * Author: Gareth Oliver, gco@google.com
- * Change Controller: IETF
- * Provisional registration? No

To indicate that the content is a Key Binding SD-JWT+KB:

- * Type name: application
- * Subtype name: kb+sd-jwt+kb
- * Required parameters: n/a
- * Optional parameters: n/a

- * Encoding considerations: binary; A Key Binding SD-JWT+KB is a SD-JWT+KB; SD-JWT values are a series of base64url-encoded values (some of which may be the empty string) separated by period ('.') and tilde('~') characters.
- * Security considerations: See the Security Considerations section of [[this specification]] and [RFC9901].
- * Interoperability considerations: n/a
- * Published specification: [[this specification]]
- * Applications that use this media type: Applications utilizing a JWT based proof of possession mechanism with further selective disclosure.
- * Fragment identifier considerations: n/a
- * 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: Gareth Oliver, gco@google.com
- * Intended usage: COMMON
- * Restrictions on usage: none
- * Author: Gareth Oliver, gco@google.com
- * Change Controller: IETF
- * Provisional registration? No

Appendix B. Acknowledgments

Author's Address

Gareth Oliver
Google
Email: gco@google.com