
Stream: Internet Engineering Task Force (IETF)
RFC: [9278](#)
Category: Standards Track
Published: August 2022
ISSN: 2070-1721
Authors: M. Jones K. Yasuda
Microsoft Microsoft

RFC 9278

JWK Thumbprint URI

Abstract

This specification registers a kind of URI that represents a JSON Web Key (JWK) Thumbprint value. JWK Thumbprints are defined in RFC 7638. This enables JWK Thumbprints to be used, for instance, as key identifiers in contexts requiring URIs.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc9278>.

Copyright Notice

Copyright (c) 2022 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. Requirements Notation and Conventions
- 3. JWK Thumbprint URI
- 4. Hash Algorithms Identifier
- 5. Mandatory to Implement Hash Algorithm
- 6. Example JWK Thumbprint URI
- 7. Security Considerations
 - 7.1. Multiple Public Keys per Private Key
- 8. IANA Considerations
 - 8.1. OAuth URI Registration
 - 8.1.1. Registry Contents
- 9. References
 - 9.1. Normative References
 - 9.2. Informative References

[Acknowledgements](#)

[Authors' Addresses](#)

1. Introduction

A JSON Web Key (JWK) Thumbprint [[RFC7638](#)] is a URL-safe representation of a hash value over a JWK [[RFC7517](#)]. This specification defines a URI prefix indicating that the portion of the URI following the prefix is a JWK Thumbprint. This enables JWK Thumbprints to be communicated in contexts requiring URIs, including in specific JSON Web Token (JWT) [[RFC7519](#)] claims.

JWK Thumbprint URIs are being used in the [[SIOPv2](#)] specification as one kind of subject identifier in a context requiring that the identifier be a URI. In this case, the subject identifier is derived from a public key represented as a JWK. Expressing the identifier as a JWK Thumbprint URI enables this kind of identifier to be differentiated from other kinds of identifiers that are also URIs, such as Decentralized Identifiers (DIDs) [[DID-Core](#)].

2. 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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. JWK Thumbprint URI

The following URI prefix is defined to indicate that the portion of the URI following the prefix is a JWK Thumbprint:

```
urn:ietf:params:oauth:jwk-thumbprint
```

To make the hash algorithm being used explicit in a URI, the prefix is followed by a hash algorithm identifier and a JWK Thumbprint value, each separated by a colon character to form a URI representing a JWK Thumbprint.

4. Hash Algorithms Identifier

Hash algorithm identifiers used in JWK Thumbprint URIs **MUST** be values from the "Hash Name String" column in the IANA "Named Information Hash Algorithm Registry" [IANA.Hash.Algorithms]. JWK Thumbprint URIs with hash algorithm identifiers not found in this registry are not considered valid and applications will need to detect and handle this error, should it occur.

5. Mandatory to Implement Hash Algorithm

To promote interoperability among implementations, the SHA-256 hash algorithm is mandatory to implement.

6. Example JWK Thumbprint URI

Section 3.1 of [RFC7638] contains the following example JWK Thumbprint value:

```
NzbLsXh8uDCcd-6MNwXF4W_7noWZFZAfHkxZsRGC9Xs
```

A complete JWK Thumbprint URI using the above JWK Thumbprint and SHA-256 hash algorithm is as follows:

```
urn:ietf:params:oauth:jwk-  
thumbprint:sha-256:NzbLsXh8uDCcd-6MNwXF4W_7noWZFZAfHkxZsRGC9Xs
```

7. Security Considerations

The security considerations of [RFC7638] also apply when using this specification.

7.1. Multiple Public Keys per Private Key

There are cryptographic algorithms for which multiple public keys correspond to the same private key. This is described in the security considerations of [RFC7748] as follows:

Designers using these curves should be aware that for each public key, there are several publicly computable public keys that are equivalent to it, i.e., they produce the same shared secrets. Thus using a public key as an identifier and knowledge of a shared secret as proof of ownership (without including the public keys in the key derivation) might lead to subtle vulnerabilities.

This consideration for public keys as identifiers equally applies to JWK Thumbprint URIs used as identifiers. A recommended way to ensure that the JWK Thumbprint URI corresponds to the actual public key used is to sign a message containing the correct public key with the private key. This signed message could also contain the JWK Thumbprint URI (although, by definition, it could also be computed directly from the public key).

8. IANA Considerations

8.1. OAuth URI Registration

This specification registers the following value in the IANA "OAuth URI" registry [IANA.OAuth.Parameters] established by [RFC6755].

8.1.1. Registry Contents

URN: urn:ietf:params:oauth:jwk-thumbprint

Common Name: JWK Thumbprint URI

Change controller: IESG

Specification Document: RFC 9278

9. References

9.1. Normative References

[IANA.OAuth.Parameters] IANA, "OAuth Parameters", <<http://www.iana.org/assignments/oauth-parameters>>.

- [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>>.
- [RFC7638] Jones, M. and N. Sakimura, "JSON Web Key (JWK) Thumbprint", RFC 7638, DOI 10.17487/RFC7638, September 2015, <<https://www.rfc-editor.org/info/rfc7638>>.
- [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>>.

9.2. Informative References

- [DID-Core] Sporny, M., Guy, A., Sabadello, M., and D. Reed, "Decentralized Identifiers (DIDs) v1.0", August 2021, <<https://www.w3.org/TR/2021/PR-did-core-20210803/>>.
- [IANA.Hash.Algorithms] IANA, "Named Information Hash Algorithm Registry", <<https://www.iana.org/assignments/named-information>>.
- [RFC6755] Campbell, B. and H. Tschofenig, "An IETF URN Sub-Namespace for OAuth", RFC 6755, DOI 10.17487/RFC6755, October 2012, <<https://www.rfc-editor.org/info/rfc6755>>.
- [RFC7517] Jones, M., "JSON Web Key (JWK)", RFC 7517, DOI 10.17487/RFC7517, May 2015, <<https://www.rfc-editor.org/info/rfc7517>>.
- [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>>.
- [RFC7748] Langley, A., Hamburg, M., and S. Turner, "Elliptic Curves for Security", RFC 7748, DOI 10.17487/RFC7748, January 2016, <<https://www.rfc-editor.org/info/rfc7748>>.
- [SIOPv2] Yasuda, K., Jones, M., and T. Lodderstedt, "Self-Issued OpenID Provider v2", June 2022, <https://openid.net/specs/openid-connect-self-issued-v2-1_0.html>.

Acknowledgements

Use cases for this specification were developed in the OpenID Connect Working Group of the OpenID Foundation. Specifically, it is being used as a key identifier in the [SIOPv2] specification.

The following individuals also contributed to the creation of this specification: John Bradley, Scott Bradner, Brian Campbell, Roman Danyliw, Vladimir Dzhuvinov, Lars Eggert, Warren Kumari, Adam Lemmon, Neil Madden, James Manger, Francesca Palombini, Aaron Parecki, Gonzalo Salgueiro, Rifaat Shekh-Yusef, Robert Sparks, David Waite, Robert Wilton, and Paul Wouters.

Authors' Addresses

Michael B. Jones

Microsoft

Email: mbj@microsoft.com

URI: <https://self-issued.info/>

Kristina Yasuda

Microsoft

Email: kryasuda@microsoft.com

URI: <https://twitter.com/kristinayasuda>