Workgroup: Network Working Group
Internet-Draft: draft-santesson-svt-jws-01
Published: 21 March 2022
Intended Status: Informational
Expires: 22 September 2022
Authors: S. Santesson R. Housley
IDsec Solutions Vigil Security

JWS Signature Validation Token

Abstract

This document defines a JSON Web Signature (JWS) profile for the Signature Validation Token defined in [SVT].

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 22 September 2022.

Table of Contents

1. Introduction

The "Signature Validation Token" specification [SVT] defines the basic token to support signature validation in a way that can significantly extend the lifetime of a signature.

This specification defines a profile for implementing SVT with a JWS signed payload according to [RFC7515], and defines the following aspects of SVT usage:

- How to include reference data related to JWS signatures in an SVT.

- How to add an SVT token to JWS signatures.

A JWS may have one or more signatures depending on its serialization format, signing the same payload data. A JWS either contains the data to be signed (enveloping) or may sign any externally associated payload data (detached).

To provide a generic solution for JWS, an SVT is added to each present signature as a JWS Unprotected Header. If a JWS includes multiple signatures, then each signature includes its own SVT.

2. Definitions

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.

The definitions in [SVT] and [RFC7515] apply also to this document.

3. SVT in JWS

An SVT token MAY be added to any signature of a JWS to support validation of that signature. If more than one signature is present then each present SVT MUST provide information exclusively related to one associated signature and MUST NOT include information about any other signature in the JWS.

Each SVT is stored in its associated signature's "svt" header as defined in Section 3.1.

3.1. "svt" Header Parameter

The "svt" (Signature Validation Token) Header Parameter is used to contain an array of SVT tokens to support validation of the associated signature. Each SVT token in the array has the format of a JWT as defined in [RFC7519] and is stored using its natural string representation without further wrapping or encoding.

The "svt" Header Parameter, when used, MUST be included as a JWS Unprotected Header.

Note: JWS Unprotected Header is not supported with JWS Compact Serialization. A consequence of adding an SVT token to a JWS is therefore that JWS JSON Serialization MUST be used, either in the form of general JWS JSON Serialization (for one or more signatures) or in the form of flattened JWS JSON Serialization (optionally used when only one signature is present in the JWS).

3.2. Multiple SVT in a signature

If a new SVT is stored in a signature which already contains a previously issued SVT, implementations can choose to either replace the existing SVT or to store the new SVT in addition to the existing SVT.

If a JWS signature already contains an array of SVTs and a new SVT is to be added, then the new SVT MUST be added to the array of SVT tokens in the existing "svt" Header Parameter.

4. SVT Claims

4.1. Profile Identifer

When this profile is used the SigValidation object MUST contain a "profile" claim with the value "JWS".

4.2. Signature Reference Data

The SVT Signature object MUST contain a "sig_ref" claim (SigReference object) with the following elements:

* "sig_hash" -- The hash over the associated signature value (the bytes of the base64url-decoded signature parameter).

* "sb_hash" -- The hash over all bytes signed by the associated signature (the JWS Signing Input according to [RFC7515]).

4.3. Signed Data Reference Data

The SVT Signature object MUST contain one instance of the "sig_data" claim (SignedData object) with the following elements:

* "ref" -- This parameter MUST hold one of the following thee possible values.

    1. The explicit string value "payload" if the signed JWS Payload is embedded in a "payload" member of the JWS.

    2. The explicit string value "detached" if the JWS signs detached payload data without explicit reference.

    3. A URI that can be used to identify or fetch the detached signed data. The means to determine the URI for the detached signed data is outside the scope of this specification.

* "hash" -- The hash over the JWS Payload data bytes (not its base64url-encoded string representation).

4.4. Signer Certificate References

The SVT Signature object MUST contain a "signer_cert_ref" claim (CertReference object). The "type" parameter of the "signer_cert_ref" claim MUST be either "chain" or "chain_hash".

* The "chain" type MUST be used when signature validation was performed using one or more certificates where some or all of the certificates in the chain are not present in the target signature.

* The "chain_hash" type MUST be used when signature validation was performed using one or more certificates where all of the certificates are present in the target signature JOSE header using the "x5c" Header Parameter.

5. SVT JOSE Header

5.1. SVT Signing Key Reference

The SVT JOSE header must contain one of the following header parameters in accordance with [RFC7515], for storing a reference to the public key used to verify the signature on the SVT:

* "x5c" -- Holds an X.509 certificate [RFC5280] or a chain of certificates. The certificate holding the public key that verifies the signature on the SVT MUST be the first certificate in the chain.

* "kid" -- A key identifier holding the Base64 encoded hash value of the certificate that can verify the signature on the SVT. The hash algorithm MUST be the same hash algorithm used when signing the SVT as specified by the alg header parameter.

6. IANA Considerations

6.1. Header Parameter Names Registration

This section registers the "svt" Header Parameter in the IANA "JSON Web Signature and Encryption Header Parameters" registry established by [RFC7515].

6.1.1. Registry Contents

* Header Parameter Name: "svt"

* Header Parameter Description: Signature Validation Token

* Header Parameter Usage Location(s): JWS

* Change Controller: IESG

* Specification Document(s): Section 3.1 of {this document}

NOTE to RFC editor: Please replace {this document} with its assigned RFC number.

7. Security Considerations

The security considerations of [SVT] applies also to this document.

8. Normative References

[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>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[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>.

[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>.

[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>.

[SVT] Santesson, S. and R. Housley, "Signature Validation Token", IETF draft-santesson-svt-02, September 2021.

Authors' Addresses

Stefan Santesson
IDsec Solutions AB
Forskningsbyn Ideon
SE-223 70 Lund
Sweden

Email: sts@aaa-sec.com

Russ Housley
Vigil Security, LLC
516 Dranesville Road
Herndon, VA, 20170
United States of America

Email: housley@vigilsec.com