LAMPS
Internet-Draft
Intended status: Standards Track
Expires: 30 August 2026

S. Santesson
IDsec Solutions
R. Housley
Vigil Security

26 February 2026

One Signature Certificates
draft-santesson-one-signature-certs-01

Abstract

This document defines a profile for certificates that are issued for validation of the digital signature produced by a single signing operation. Each certificate is created at the time of signing and bound to the signed content. The associated signing key is generated, used to produce a single digital signature, and then immediately destroyed. The certificate never expires and is never revoked, which simplifies long-term validation.

About This Document

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

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-santesson-one-signature-certs/.

Source for this draft and an issue tracker can be found at https://github.com/Razumain/one-signature-certs.

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 2026.

Table of Contents

1. Introduction

The landscape of server-based signing services has changed over the decades. Recently, one type of signature service has gained favor, where the signing private key and the signing certificate are created for each digital signature, rather than re-using a static key and certificate over an extended time period.

Some reasons why this type of signature services has been successful are:

* The certificate will always have a predictable validity time from the time of signing;

* The time of signing is guaranteed by the certificate issue date;

* The identity information in the certificate can be adapted to the signing context;

* Revocation of signing certificates is practically non-existent despite many years of operation and millions of signatures; and

* The signature service holds no pre-stored user keys or certificates.

While this type of signature service solves many problems, it still suffers from the complexity caused by expiring signing certificates. One solution to this problem is the Signature Validation Token (SVT) [RFC9321], where future validation can rely on a previous successful validation rather than validation based on aging data.

This document takes this one step further and allows validation at any time in the future as long as trust in the CA certificate can be established.

1.1. Basic features

One signature certificates have the following common characteristics:

* They never expire;

* They are never revoked;

* They are bound to a specific document content; and

* They assert that the corresponding private key was destroyed immediately after signing.

1.1.1. Revocation

Traditional certificates that are re-used over time have many legitimate reasons for revocation, such as if the private key is lost or compromised. This can lead to large volumes of revocation data.

The fact that the same key is used many times exposes the key for the risks of loss, unauthorized usage, or theft. When many objects are signed with the same private key, the risk of exposure and the number of affected signed documents upon revocation increases, unless properly timestamped and properly verified.

When a signing key is used only once, that risk of exposure is greatly reduced, and it has been shown that most usages of dedicated private keys and certificates no longer require revocation.

The CA can readily attest that a certain procedure was followed when the certificate was issued. As a matter of policy, the certificate itself is an attestation that the CP and CPS [RFC3647] were followed successfully when the signature was created. Certificates issued according to this profile therefore only attest to the validity at the time of issuance and signing, rather than a retroactive state at the time of validation. This profile is intended for those applications where this declaration of validity is relevant and useful.

Applications that require traditional revocation checking that provides the state at the time of validation MUST NOT use this profile.

An example usage where this is useful is in services where the signed document is stored as an internal evidence record, such as when a Tax agency allows citizens to sign their tax declarations. This record is then pulled out and used only in case of a dispute where the identified signer challenges the signature. A revocation service is less likely to contribute to this process. If the challenge is successful, the signed document will be removed without affecting any other signed documents in the archive.

2. Conventions and 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.

3. Certificate content

Conforming certificates SHALL meet all requirements of this section.

Certificates MUST indicate that a certificate has no well-defined expiration date by setting the notAfter field to the GeneralizedTime value 99991231235959Z, as defined in [RFC5280].

Certificates MUST include the id-ce-noRevAvail extension in compliance with [RFC9608], indicating that this certificate is not supported by any revocation mechanism.

Certificates MUST include the signedDocumentBinding extension, binding the certificate to a specific signed content.

3.1. The signedDocumentBinding extension

The signedDocumentBinding extension binds a certificate to a specific signed content. When present, conforming CAs SHOULD mark this extension as non-critical.

```
name           id-pe-signedDocumentBinding
OID            { id-pe TBD }
syntax         SignedDocumentBinding
criticality    SHOULD be FALSE

SignedDocumentBinding ::= SEQUENCE {
dataTbsHash     OCTET STRING,
hashAlg         DigestAlgorithmIdentifier,
bindingType     UTF8String OPTIONAL }
```

The dataTbsHash field MUST contain a hash of the data to be signed.

The hashAlg field MUST contain the AlgorithmIdentifier of the hash algorithm used to generate the dataTbsHash value.

The bindingType field MAY contain an identifier that specifies how the data to be signed is derived from the digital object to be signed.

Adding this extension to a certificate is a statement by the CA that the signing key is generated exclusively for the purpose of signing the document bound by this extension, and that the signing key is destroyed after signing. The details for this procedure and how the destruction of the signing key is assured SHOULD be outlined in the certificate policy [RFC3647] of the issued certificate.

3.2. Defined bindingType identifiers

The bindingType field defines how the data to be signed (dataTbsHash) is derived from the signed document. This field identifies a deterministic procedure for selecting the portion of the signed content that is included in the hash computation. When the field is omitted, the rules for the default binding type apply.

The purpose of the dataTbsHash value is to bind the certificate to the document being signed in order to prevent re-use of the signing key for multiple signed documents. This enforces the contract that the signing key is used only once for creation of one signature only. Validators SHOULD verify that the signed document matches the certificate's binding information. This verification is not required for the signature to validate successfully but provides an additional safeguard against misuse or substitution of certificates.

This document defines a set of bindingType identifiers. Additional bindingType identifiers MAY be defined by future specifications.

3.2.1. Default Binding

When the bindingType is absent, the default binding applies. In this case, the dataTbsHash value is the hash of the exact data that is hashed and signed by the signature format in use.

Examples include: - For XML Signatures [XMLDSIG11], the hash of the SignedInfo element. - For CMS Signatures [RFC5652], the DER-encoded SignedAttributes structure. - For other formats, the data structure input directly to the signature algorithm.

This bindingType MUST NOT be used when the data to be signed includes either the signer certificate itself or a hash of the signer certificate. This includes JWS and COSE signed documents that can include signer certificates in the protected header. JWS signatures [RFC7515] MUST use the "jws" bindingType and COSE signatures [RFC8152] MUST use the "cose" binding type.

3.2.2. CAdES Binding

Identifier: "cades"

For CMS [RFC5652] or ETSI CAdES [CADES] signatures incorporating SigningCertificate or SigningCertificateV2 attributes [RFC5035] in signedAttrs, the dataTbsHash value is computed over the DER encoding of SignerInfo excluding any instances of SigningCertificate or SigningCertificateV2 attributes from the SignedAttributes set.

This bindingType also applies to PDF [ISOPDF2] and ETSI PAdES [PADES] signed documents when applicable due to its use of CMS for signing.

3.2.3. XAdES Binding

Identifier: "xades"

For ETSI XML Advanced Electronic Signatures [XADES], the dataTbsHash value is computed over the canonicalized SignedInfo element, with any Reference elements whose Type attribute equals "http://uri.etsi.org/01903#SignedProperties" removed prior to hashing. This ensures that the SignedProperties element, which may contain references to the signing certificate, does not create a circular dependency. Extraction of the Reference element MUST be done by removing only the characters from the leading <Reference> tag up to and including the ending </Reference> tag, preserving all other bytes of SignedInfo unchanged, including any white space or line feeds.

Note: This operation is purely textual and does not require XML parsing beyond locating the tag boundaries.

3.2.4. JWS Binding

Identifier: "jws"

For JSON Web Signatures (JWS) [RFC7515], the dataTbsHash value is computed over the payload only. The protected header and any unprotected header parameters MUST NOT be included in the hash calculation.

This exclusion avoids circular dependencies where certificate data may appear in the protected header.

3.2.5. COSE Binding

Identifier: "cose"

For COSE signatures [RFC8152], the dataTbsHash value is computed over the payload only. The protected header and any unprotected header parameters MUST NOT be included in the hash calculation.

This exclusion avoids circular dependencies where certificate data may appear in the protected header.

4. ASN.1 Module

```
<CODE BEGINS>
   SignedDocumentBindingExtn
     { iso(1) identified-organization(3) dod(6) internet(1)
       security(5) mechanisms(5) pkix(7) id-mod(0)
       id-mod-signedDocumentBinding(TBD) }

   DEFINITIONS IMPLICIT TAGS ::=
   BEGIN

   IMPORTS
     EXTENSION, id-pkix, id-pe
     FROM PKIX-CommonTypes-2009  -- RFC 5912
       { iso(1) identified-organization(3) dod(6) internet(1)
         security(5) mechanisms(5) pkix(7) id-mod(0)
         id-mod-pkixCommon-02(57) }

     DigestAlgorithmIdentifier
     FROM CryptographicMessageSyntax-2010 -- RFC 6268
       { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
         pkcs-9(9) smime(16) modules(0) id-mod-cms-2009(58) } ;

   -- signedDocumentBinding Certificate Extension

   ext-SignedDocumentBinding EXTENSION ::= {
     SYNTAX SignedDocumentBinding
     IDENTIFIED BY id-pe-signedDocumentBinding }

   SignedDocumentBinding ::= SEQUENCE {
     dataTbsHash     OCTET STRING,
     hashAlg         DigestAlgorithmIdentifier,
     bindingType     UTF8String OPTIONAL }

   -- signedDocumentBinding Certificate Extension OID

   id-pe-signedDocumentBinding OBJECT IDENTIFIER ::= { id-pe TBD }

   END
<CODE ENDS>
```

5. Security Considerations

5.1. Certificates Without Revocation

Certificates conforming to this profile include the id-ce-noRevAvail extension and therefore do not provide any revocation mechanism. Such certificates attest only to the state of trust and correctness of procedures at the time of issuance.

The Security considerations in [RFC9608] also applies to this document.

5.2. Signed Document Binding

The signedDocumentBinding extension binds the certificate to specific signed content by including a hash of the data to be signed. Verification of this binding is not required for successful cryptographic validation of the signature. A signature can therefore validate correctly even if the binding is not checked.

However, a relying party SHOULD verify that the signed content matches the dataTbsHash value in the signedDocumentBinding extension. Performing this check ensures that the certificate is used only with the content for which it was issued and enforces the intended scope of the certificate.

The security model of this profile states that the associated private key is generated for, and used in, exactly one signing operation and is then destroyed. This property holds independently of whether the binding is verified by the relying party. Nevertheless, failure to verify the binding weakens the protections provided by this profile and increases the risk of certificate substitution or unintended certificate reuse.

When verified, the signedDocumentBinding extension provides an additional safeguard against the use of the certificate for any signature other than the one for which it was issued.

6. IANA Considerations

6.1. Registry for signedDocumentBinding bindingType Identifiers

IANA is requested to create a new registry entitled: “Signed Document Binding Type Identifiers”

This registry shall contain identifiers used in the bindingType field of the signedDocumentBinding certificate extension defined in this document.

6.1.1. Registry Contents

Each registry entry shall contain the following fields:

* Identifier: A UTF-8 string identifying the binding type.

* Description: A brief description of how the dataTbsHash value is computed.

* Reference: A reference to the document that defines the binding type.

6.1.2. Registration Policy

The registration policy for this registry is Specification Required as defined in [RFC8174].

The designated expert(s) SHALL ensure that:

* The binding type definition clearly specifies a deterministic and unambiguous procedure for computing the dataTbsHash value.

* The specification explains how circular dependencies with certificate inclusion are avoided, where applicable.

* The identifier is unique within the registry.

6.1.3. Initial Registry Contents

IANA is requested to populate the registry with the following initial values:

* Identifier: (absent)

* Description: Default binding as defined in this document

* Reference: This document

* Identifier: cades

* Description: CMS/CAdES binding excluding SigningCertificate attributes

* Reference: This document

* Identifier: xades

* Description: XAdES binding excluding SignedProperties reference

* Reference: This document

* Identifier: jws

* Description: JWS payload-only binding

* Reference: This document

* Identifier: cose

* Description: COSE payload-only binding

* Reference: This document

7. References

7.1. Normative References

[CADES] ETSI, "Electronic Signatures and Infrastructures (ESI); CAdES digital signatures; Part 1: Building blocks and CAdES baseline signatures", ETSI EN 319 122-1 v1.2.1, October 2021.

[ISOPDF2] ISO, "Document management -- Portable document format -- Part 2: PDF 2.0", ISO 32000-2, July 2017.

[PADES] ETSI, "Electronic Signatures and Infrastructures (ESI); PAdES digital signatures; Part 1: Building blocks and PAdES baseline signatures", ETSI EN 319 142-1 v1.2.1, January 2024.

[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/rfc/rfc2119>.

[RFC3647] Chokhani, S., Ford, W., Sabett, R., Merrill, C., and S. Wu, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework", RFC 3647, DOI 10.17487/RFC3647, November 2003, <https://www.rfc-editor.org/rfc/rfc3647>.

[RFC5035] Schaad, J., "Enhanced Security Services (ESS) Update: Adding CertID Algorithm Agility", RFC 5035, DOI 10.17487/RFC5035, August 2007, <https://www.rfc-editor.org/rfc/rfc5035>.

[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/rfc/rfc5280>.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <https://www.rfc-editor.org/rfc/rfc5652>.

[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/rfc/rfc7515>.

[RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", RFC 8152, DOI 10.17487/RFC8152, July 2017, <https://www.rfc-editor.org/rfc/rfc8152>.

[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/rfc/rfc8174>.

[RFC9608] Housley, R., Okubo, T., and J. Mandel, "No Revocation Available for X.509 Public Key Certificates", RFC 9608, DOI 10.17487/RFC9608, June 2024, <https://www.rfc-editor.org/rfc/rfc9608>.

[XADES] ETSI, "Electronic Signatures and Infrastructures (ESI); XAdES digital signatures; Part 1: Building blocks and XAdES baseline signatures", ETSI EN 319 132-1 v1.3.1, July 2024.

[XMLDSIG11] Eastlake, D., Reagle, J., Solo, D., Hirsch, F., Nystrom, M., Roessler, T., and K. Yiu, "XML Signature Syntax and Processing Version 1.1", W3C Proposed Recommendation, 11 April 2013.

7.2. Informative References

[RFC9321] Santesson, S. and R. Housley, "Signature Validation Token", RFC 9321, DOI 10.17487/RFC9321, October 2022, <https://www.rfc-editor.org/rfc/rfc9321>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Stefan Santesson
IDsec Solutions AB
Forskningsbyn Ideon
SE-223 70 Lund
Sweden
Email: sts@aaa-sec.com

Russ Housley
Vigil Security, LLC
Herndon, VA,
United States of America
Email: housley@vigilsec.com