LAMPS
Internet-Draft
Intended status: Standards Track
Expires: 25 July 2026

M. Ounsworth
J. Gray
Entrust
J. Klaussner
Bundesdruckerei GmbH
D. Van Geest
CryptoNext Security

21 January 2026

Composite ML-DSA for use in Cryptographic Message Syntax (CMS)
draft-ietf-lamps-cms-composite-sigs-01

Abstract

Composite ML-DSA defines combinations of ML-DSA, as defined by NIST in FIPS 204, with RSA, ECDSA, and EdDSA. This document specifies the conventions for using Composite ML-DSA algorithms within the Cryptographic Message Syntax (CMS).

About This Document

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

The latest revision of this draft can be found at https://lamps-wg.github.io/cms-composite-sigs/draft-ietf-lamps-cms-composite-sigs.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-lamps-cms-composite-sigs/.

Discussion of this document takes place on the LAMPS Working Group mailing list (mailto:spams@ietf.org), which is archived at https://datatracker.ietf.org/wg/lamps/about/. Subscribe at https://www.ietf.org/mailman/listinfo/spams/.

Source for this draft and an issue tracker can be found at https://github.com/lamps-wg/cms-composite-sigs.

Status of This Memo

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 25 July 2026.

Copyright Notice

Table of Contents

1. Introduction

[I-D.ietf-lamps-pq-composite-sigs] defines a collection of signature algorithms, referred to as Composite ML-DSA, which combine ML-DSA [FIPS204] with traditional algorithms RSASSA-PKCS1-v1.5, RSASSA-PSS, ECDSA, Ed25519, and Ed448. This document acts as a companion to [I-D.ietf-lamps-pq-composite-sigs] by providing conventions for using Composite ML-DSA algorithms within the Cryptographic Message Syntax (CMS) [RFC5652].

1.1. Conventions and Terminology

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. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

This document is consistent with the terminology defined in [RFC9794].

2. Composite ML-DSA Algorithm Identifiers

Many ASN.1 data structure types use the AlgorithmIdentifier type to identify cryptographic algorithms. In the CMS, AlgorithmIdentifiers are used to identify Composite ML-DSA signatures in the signed-data content type. They may also appear in X.509 certificates used to verify those signatures. The same AlgorithmIdentifiers are used to identify Composite ML-DSA public keys and signature algorithms. [I-D.ietf-lamps-pq-composite-sigs] describes the use of Composite ML-DSA in X.509 certificates. The AlgorithmIdentifier type is defined as follows:

```
AlgorithmIdentifier{ALGORITHM-TYPE, ALGORITHM-TYPE:AlgorithmSet} ::=
        SEQUENCE {
            algorithm   ALGORITHM-TYPE.&id({AlgorithmSet}),
            parameters  ALGORITHM-TYPE.
                   &Params({AlgorithmSet}{@algorithm}) OPTIONAL
        }
```

> NOTE: The above syntax is from [RFC5911] and is compatible with the 2021 ASN.1 syntax [X680]. See [RFC5280] for the 1988 ASN.1 syntax.

The fields in the AlgorithmIdentifier type have the following meanings:

algorithm: The algorithm field contains an OID that identifies the cryptographic algorithm in use. The OIDs for Composite ML-DSA algorithms are described below.

parameters: The parameters field contains parameter information for the algorithm identified by the OID in the algorithm field. Each Composite ML-DSA parameter set is identified by its own algorithm OID, so there is no relevant information to include in this field. As such, parameters MUST be omitted when encoding a Composite ML-DSA AlgorithmIdentifier.

The object identifiers for Composite ML-DSA algorithms are defined in [I-D.ietf-lamps-pq-composite-sigs], and are reproduced here for convenience.

```
id-MLDSA44-RSA2048-PSS-SHA256 OBJECT IDENTIFIER ::= {
   iso(1) org(3) dod(6) internet(1) security(5) mechanisms(5)
   pkix(7) alg(6) 37 }
id-MLDSA44-RSA2048-PKCS15-SHA256 OBJECT IDENTIFIER ::= {
   iso(1) org(3) dod(6) internet(1) security(5) mechanisms(5)
   pkix(7) alg(6) 38 }
id-MLDSA44-Ed25519-SHA512 OBJECT IDENTIFIER ::= {
   iso(1) org(3) dod(6) internet(1) security(5) mechanisms(5)
   pkix(7) alg(6) 39 }
id-MLDSA44-ECDSA-P256-SHA256 OBJECT IDENTIFIER ::= {
   iso(1) org(3) dod(6) internet(1) security(5) mechanisms(5)
   pkix(7) alg(6) 40 }
id-MLDSA65-RSA3072-PSS-SHA512 OBJECT IDENTIFIER ::= {
   iso(1) org(3) dod(6) internet(1) security(5) mechanisms(5)
   pkix(7) alg(6) 41 }
id-MLDSA65-RSA3072-PKCS15-SHA512 OBJECT IDENTIFIER ::= {
   iso(1) org(3) dod(6) internet(1) security(5) mechanisms(5)
   pkix(7) alg(6) 42 }
id-MLDSA65-RSA4096-PSS-SHA512 OBJECT IDENTIFIER ::= {
   iso(1) org(3) dod(6) internet(1) security(5) mechanisms(5)
   pkix(7) alg(6) 43 }
id-MLDSA65-RSA4096-PKCS15-SHA512 OBJECT IDENTIFIER ::= {
   iso(1) org(3) dod(6) internet(1) security(5) mechanisms(5)
   pkix(7) alg(6) 44 }
id-MLDSA65-ECDSA-P256-SHA512 OBJECT IDENTIFIER ::= {
   iso(1) org(3) dod(6) internet(1) security(5) mechanisms(5)
   pkix(7) alg(6) 45 }
id-MLDSA65-ECDSA-P384-SHA512 OBJECT IDENTIFIER ::= {
   iso(1) org(3) dod(6) internet(1) security(5) mechanisms(5)
   pkix(7) alg(6) 46 }
id-MLDSA65-ECDSA-brainpoolP256r1-SHA512 OBJECT IDENTIFIER ::= {
   iso(1) org(3) dod(6) internet(1) security(5) mechanisms(5)
   pkix(7) alg(6) 47 }
```

```
id-MLDSA65-Ed25519-SHA512 OBJECT IDENTIFIER ::= {
   iso(1) org(3) dod(6) internet(1) security(5) mechanisms(5)
   pkix(7) alg(6) 48 }
id-MLDSA87-ECDSA-P384-SHA512 OBJECT IDENTIFIER ::= {
   iso(1) org(3) dod(6) internet(1) security(5) mechanisms(5)
   pkix(7) alg(6) 49 }
id-MLDSA87-ECDSA-brainpoolP384r1-SHA512 OBJECT IDENTIFIER ::= {
   iso(1) org(3) dod(6) internet(1) security(5) mechanisms(5)
   pkix(7) alg(6) 50 }
id-MLDSA87-Ed448-SHAKE256 OBJECT IDENTIFIER ::= {
   iso(1) org(3) dod(6) internet(1) security(5) mechanisms(5)
   pkix(7) alg(6) 51 }
id-MLDSA87-RSA3072-PSS-SHA512 OBJECT IDENTIFIER ::= {
   iso(1) org(3) dod(6) internet(1) security(5) mechanisms(5)
   pkix(7) alg(6) 52 }
id-MLDSA87-RSA4096-PSS-SHA512 OBJECT IDENTIFIER ::= {
   iso(1) org(3) dod(6) internet(1) security(5) mechanisms(5)
   pkix(7) alg(6) 53 }
id-MLDSA87-ECDSA-P521-SHA512 OBJECT IDENTIFIER ::= {
   iso(1) org(3) dod(6) internet(1) security(5) mechanisms(5)
   pkix(7) alg(6) 54 }
```

3. Signed-Data Conventions

3.1. Pre-Hashing

[RFC5652] specifies that digital signatures for CMS are produced using a digest of the message to be signed and the signer’s private key. At the time RFC 5652 was published, all signature algorithms supported in the CMS required a message digest to be calculated externally to that algorithm, which would then be supplied to the algorithm implementation when calculating and verifying signatures. Since then, EdDSA [RFC8032] and ML-DSA [FIPS204] have also been standardized, and these algorithms support both a "pure" and "pre-hash" mode, although their use in CMS has only been defined for "pure" mode.

Composite ML-DSA operates only in a "pre-hash" mode. However, unlike RSA and ECDSA each Composite ML-DSA algorithm is defined to be used with a single digest algorithm which is identified in the Composite ML-DSA algorithm name. For example, id-MLDSA87-ECDSA-P521-SHA512 uses SHA-512 as its pre-hash digest algorithm.

When Composite ML-DSA is used in CMS, the digest algorithm used by CMS SHALL be the same pre-hash digest algorithm used by the Composite ML-DSA algorithm. A Composite ML-DSA algorithm might use additional digest algorithms for the internal component algorithms, these digest algorithms are irrelevant to Composite ML-DSA’s use in CMS.

3.2. SignedData digestAlgorithms

The SignedData digestAlgorithms field includes the identifiers of the message digest algorithms used by one or more signer. There MAY be any number of elements in the collection, including zero. When signing with a Composite ML-DSA algorithm, the list of identifiers MAY include a digest algorithm from Table 1. The digest algorithm(s) included will depend on the Composite ML-DSA algorithm(s) used for signing. If such a digest algorithm is present, the algorithm parameters field MUST be absent.

3.3. Signature Generation and Verification

[RFC5652] describes the two methods that are used to calculate and verify signatures in the CMS. One method is used when signed attributes are present in the signedAttrs field of the relevant SignerInfo, and another is used when signed attributes are absent. Use of signed attributes is preferred, but the conventions for signed-data without signed attributes is also described below for completeness.

When signed attributes are absent, Composite ML-DSA signatures are computed over the content of the signed-data. As described in Section 5.4 of [RFC5652], the "content" of a signed-data is the value of the encapContentInfo eContent OCTET STRING. The tag and length octets are not included.

When signed attributes are included, Composite ML-DSA signatures are computed over the complete DER encoding of the SignedAttrs value contained in the SignerInfo’s signedAttrs field. As described in Section 5.4 of [RFC5652], this encoding includes the tag and length octets, but an EXPLICIT SET OF tag is used rather than the IMPLICIT [0] tag that appears in the final message. At a minimum, the signedAttrs field MUST include a content-type attribute and a message-digest attribute. The message-digest attribute contains a hash of the content of the signed-data, where the content is as described for the absent signed attributes case above. Recalculation of the hash value by the recipient is an important step in signature verification.

Composite ML-DSA has a context string input that can be used to ensure that different signatures are generated for different application contexts. When using Composite ML-DSA as specified in this document, the context string is set to the empty string.

3.4. SignerInfo Content

When using Composite ML-DSA, the fields of a SignerInfo are used as follows:

digestAlgorithm: Per Section 5.3 of [RFC5652], the digestAlgorithm field identifies the message digest algorithm used by the signer and any associated parameters. This MUST be the same digest algorithm used by the Composite ML-DSA algorithm. Per [RFC8933], if the signedAttrs field is present in the SignerInfo, then the same digest algorithm MUST be used to compute both the digest of the SignedData encapContentInfo eContent, which is carried in the message-digest attribute, and the digest of the DER-encoded signedAttrs, which is passed to the signature algorithm. See Table 1 for exact algorithm mappings.

[RFC5754] defines the use of SHA-256 [FIPS180] (id-sha256) and SHA-512 [FIPS180] (id-sha512) in CMS. [RFC8702] defines the used of SHAKE256 [FIPS202] in CMS (id-shake256). When id-sha256 or id-sha512 is used, the parameters field MUST be omitted. When id-shake256 is used the parameters field MUST be omitted and the digest length MUST be 64 bytes.

Signature Algorithm	Digest Algorithms
id-MLDSA44-RSA2048-PSS-SHA256	id-sha256
id-MLDSA44-RSA2048-PKCS15-SHA256	id-sha256
id-MLDSA44-Ed25519-SHA512	id-sha512
id-MLDSA44-ECDSA-P256-SHA256	id-sha256
id-MLDSA65-RSA3072-PSS-SHA512	id-sha512
id-MLDSA65-RSA3072-PKCS15-SHA512	id-sha512
id-MLDSA65-RSA4096-PSS-SHA512	id-sha512
id-MLDSA65-RSA4096-PKCS15-SHA512	id-sha512
id-MLDSA65-ECDSA-P256-SHA512	id-sha512
id-MLDSA65-ECDSA-P384-SHA512	id-sha512
id-MLDSA65-ECDSA-brainpoolP256r1-SHA512	id-sha512
id-MLDSA65-Ed25519-SHA512	id-sha512
id-MLDSA87-ECDSA-P384-SHA512	id-sha512
id-MLDSA87-ECDSA-brainpoolP384r1-SHA512	id-sha512
id-MLDSA87-Ed448-SHAKE256	id-shake256
id-MLDSA87-RSA3072-PSS-SHA512	id-sha512
id-MLDSA87-RSA4096-PSS-SHA512	id-sha512
id-MLDSA87-ECDSA-P521-SHA512	id-sha512

Table 1: Digest Algorithms for Composite ML-DSA

signatureAlgorithm: The signatureAlgorithm field MUST contain one of the Composite ML-DSA signature algorithm OIDs, and the parameters field MUST be absent. The algorithm OID MUST be one of the OIDs described in Section 2.

signature: The signature field contains the signature value resulting from the use of the Composite ML-DSA signature algorithm identified by the signatureAlgorithm field. The Composite ML-DSA signature-generation operation is specified in Section 4.2 of [I-D.ietf-lamps-pq-composite-sigs], and the signature-verification operation is specified in Section 4.3 of [I-D.ietf-lamps-pq-composite-sigs]. Note that Section 5.6 of [RFC5652] places further requirements on the successful verification of a signature.

4. ASN.1 Module

```
<CODE BEGINS>
Composite-MLDSA-CMS-2026
  { iso(1) identified-organization(3) dod(6) internet(1)
        security(5) mechanisms(5) pkix(7) id-mod(0)
        id-mod-composite-mldsa-cms-2026(TBDMOD) }

DEFINITIONS IMPLICIT TAGS ::= BEGIN

EXPORTS ALL;

IMPORTS
  SIGNATURE-ALGORITHM, SMIME-CAPS
    FROM AlgorithmInformation-2009  -- [RFC5911]
      { iso(1) identified-organization(3) dod(6) internet(1)
        security(5) mechanisms(5) pkix(7) id-mod(0)
        id-mod-algorithmInformation-02(58) }

  sa-MLDSA44-RSA2048-PSS-SHA256, sa-MLDSA44-RSA2048-PKCS15-SHA256,
  sa-MLDSA44-Ed25519-SHA512, sa-MLDSA44-ECDSA-P256-SHA256,
  sa-MLDSA65-RSA3072-PSS-SHA512, sa-MLDSA65-RSA3072-PKCS15-SHA512,
  sa-MLDSA65-RSA4096-PSS-SHA512, sa-MLDSA65-RSA4096-PKCS15-SHA512,
  sa-MLDSA65-ECDSA-P256-SHA512, sa-MLDSA65-ECDSA-P384-SHA512,
  sa-MLDSA65-ECDSA-brainpoolP256r1-SHA512, sa-MLDSA65-Ed25519-SHA512,
  sa-MLDSA87-ECDSA-P384-SHA512, sa-MLDSA87-ECDSA-brainpoolP384r1-SHA512,
  sa-MLDSA87-Ed448-SHAKE256, sa-MLDSA87-RSA3072-PSS-SHA512,
  sa-MLDSA87-RSA4096-PSS-SHA512, sa-MLDSA87-ECDSA-P521-SHA512
   FROM Composite-MLDSA-2025
      { iso(1) identified-organization(3) dod(6) internet(1)
        security(5) mechanisms(5) pkix(7) id-mod(0)
        id-mod-composite-mldsa-2025(TBDCompositeMOD) }
;

--
-- Expand the signature algorithm set used by CMS [RFC5911]
--

SignatureAlgorithmSet SIGNATURE-ALGORITHM ::= {
```

```
     sa-MLDSA44-RSA2048-PSS-SHA256 |
     sa-MLDSA44-RSA2048-PKCS15-SHA256 |
     sa-MLDSA44-Ed25519-SHA512 |
     sa-MLDSA44-ECDSA-P256-SHA256 |
     sa-MLDSA65-RSA3072-PSS-SHA512 |
     sa-MLDSA65-RSA3072-PKCS15-SHA512 |
     sa-MLDSA65-RSA4096-PSS-SHA512 |
     sa-MLDSA65-RSA4096-PKCS15-SHA512 |
     sa-MLDSA65-ECDSA-P256-SHA512 |
     sa-MLDSA65-ECDSA-P384-SHA512 |
     sa-MLDSA65-ECDSA-brainpoolP256r1-SHA512 |
     sa-MLDSA65-Ed25519-SHA512 |
     sa-MLDSA87-ECDSA-P384-SHA512 |
     sa-MLDSA87-ECDSA-brainpoolP384r1-SHA512 |
     sa-MLDSA87-Ed448-SHAKE256 |
     sa-MLDSA87-RSA3072-PSS-SHA512 |
     sa-MLDSA87-RSA4096-PSS-SHA512 |
     sa-MLDSA87-ECDSA-P521-SHA512,
     ... }

   --
   -- Expand the S/MIME capabilities set used by CMS [RFC5911]
   --

   SMimeCaps SMIME-CAPS ::= {
     sa-MLDSA44-RSA2048-PSS-SHA256.&smimeCaps |
     sa-MLDSA44-RSA2048-PKCS15-SHA256.&smimeCaps |
     sa-MLDSA44-Ed25519-SHA512.&smimeCaps |
     sa-MLDSA44-ECDSA-P256-SHA256.&smimeCaps |
     sa-MLDSA65-RSA3072-PSS-SHA512.&smimeCaps |
     sa-MLDSA65-RSA3072-PKCS15-SHA512.&smimeCaps |
     sa-MLDSA65-RSA4096-PSS-SHA512.&smimeCaps |
     sa-MLDSA65-RSA4096-PKCS15-SHA512.&smimeCaps |
     sa-MLDSA65-ECDSA-P256-SHA512.&smimeCaps |
     sa-MLDSA65-ECDSA-P384-SHA512.&smimeCaps |
     sa-MLDSA65-ECDSA-brainpoolP256r1-SHA512.&smimeCaps |
     sa-MLDSA65-Ed25519-SHA512.&smimeCaps |
     sa-MLDSA87-ECDSA-P384-SHA512.&smimeCaps |
     sa-MLDSA87-ECDSA-brainpoolP384r1-SHA512.&smimeCaps |
     sa-MLDSA87-Ed448-SHAKE256.&smimeCaps |
     sa-MLDSA87-RSA3072-PSS-SHA512.&smimeCaps |
     sa-MLDSA87-RSA4096-PSS-SHA512.&smimeCaps |
     sa-MLDSA87-ECDSA-P521-SHA512.&smimeCaps,
     ... }

   END
   <CODE ENDS>
```

5. IANA Considerations

IANA is requested to allocate a value from the "SMI Security for PKIX Module Identifier" registry for the included ASN.1 module.

* Decimal: IANA Assigned - *Replace TBDCompositeMOD*

* Description: Composite-Signatures-CMS-2026 - id-mod-composite-mldsa-cms-2026

* References: This Document

6. Security Considerations

All security considerations from [I-D.ietf-lamps-pq-composite-sigs] apply.

Security of the Composite ML-DSA private key is critical. Compromise of the private key will enable an adversary to forge arbitrary signatures.

Composite ML-DSA depends on high-quality random numbers that are suitable for use in cryptography. The use of inadequate pseudo-random number generators (PRNGs) to generate such values can significantly undermine the security properties offered by a cryptographic algorithm. For instance, an attacker may find it much easier to reproduce the PRNG environment that produced any private keys, searching the resulting small set of possibilities, rather than brute-force searching the whole key space. The generation of random numbers of a sufficient level of quality for use in cryptography is difficult; see Section 3.6.1 of [FIPS204] for some additional information.

To avoid algorithm substitution attacks, the CMSAlgorithmProtection attribute defined in [RFC6211] SHOULD be included in signed attributes.

7. References

7.1. Normative References

[FIPS180] "Secure hash standard", National Institute of Standards and Technology (U.S.), DOI 10.6028/nist.fips.180-4, 2015, <https://doi.org/10.6028/nist.fips.180-4>.

[FIPS202] "SHA-3 standard :: permutation-based hash and extendable-output functions", National Institute of Standards and Technology (U.S.), DOI 10.6028/nist.fips.202, 2015, <https://doi.org/10.6028/nist.fips.202>.

[FIPS204] "Module-lattice-based digital signature standard", National Institute of Standards and Technology (U.S.), DOI 10.6028/nist.fips.204, August 2024, <https://doi.org/10.6028/nist.fips.204>.

[I-D.ietf-lamps-pq-composite-sigs]
Ounsworth, M., Gray, J., Pala, M., Klauテ殤er, J., and S. Fluhrer, "Composite ML-DSA for use in X.509 Public Key Infrastructure", Work in Progress, Internet-Draft, draft-ietf-lamps-pq-composite-sigs-14, 7 January 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-pq-composite-sigs-14>.

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

[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>.

[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>.

[RFC5911] Hoffman, P. and J. Schaad, "New ASN.1 Modules for Cryptographic Message Syntax (CMS) and S/MIME", RFC 5911, DOI 10.17487/RFC5911, June 2010, <https://www.rfc-editor.org/rfc/rfc5911>.

[RFC8933] Housley, R., "Update to the Cryptographic Message Syntax (CMS) for Algorithm Identifier Protection", RFC 8933, DOI 10.17487/RFC8933, October 2020, <https://www.rfc-editor.org/rfc/rfc8933>.

[RFC5754] Turner, S., "Using SHA2 Algorithms with Cryptographic Message Syntax", RFC 5754, DOI 10.17487/RFC5754, January 2010, <https://www.rfc-editor.org/rfc/rfc5754>.

[RFC8702] Kampanakis, P. and Q. Dang, "Use of the SHAKE One-Way Hash Functions in the Cryptographic Message Syntax (CMS)", RFC 8702, DOI 10.17487/RFC8702, January 2020, <https://www.rfc-editor.org/rfc/rfc8702>.

[RFC6211] Schaad, J., "Cryptographic Message Syntax (CMS) Algorithm Identifier Protection Attribute", RFC 6211, DOI 10.17487/RFC6211, April 2011, <https://www.rfc-editor.org/rfc/rfc6211>.

7.2. Informative References

[X680] ITU-T, "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation", ITU-T Recommendation X.680, ISO/IEC 8824-1:2021, February 2021, <https://www.itu.int/rec/T-REC-X.680>.

[RFC9794] Driscoll, F., Parsons, M., and B. Hale, "Terminology for Post-Quantum Traditional Hybrid Schemes", RFC 9794, DOI 10.17487/RFC9794, June 2025, <https://www.rfc-editor.org/rfc/rfc9794>.

[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>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <https://www.rfc-editor.org/rfc/rfc8032>.

[RFC9882] Salter, B., Raine, A., and D. Van Geest, "Use of the ML-DSA Signature Algorithm in the Cryptographic Message Syntax (CMS)", RFC 9882, DOI 10.17487/RFC9882, October 2025, <https://www.rfc-editor.org/rfc/rfc9882>.

[RFC8411] Schaad, J. and R. Andrews, "IANA Registration for the Cryptographic Algorithm Object Identifier Range", RFC 8411, DOI 10.17487/RFC8411, August 2018, <https://www.rfc-editor.org/rfc/rfc8411>.

Appendix A. Examples

This appendix contains an example signed-data encoding with the id-MLDSA65-ECDSA-P256-SHA512 signature algorithm.

It can be verified using the example public keys and certificates specified in Appendix E of [I-D.ietf-lamps-pq-composite-sigs]. Specifically, the following example:

* tcId: id-MLDSA65-ECDSA-P256-SHA512

* x5c: Base64 of the DER encoding of the certificate. Wrap this in PEM headers and footers to get a PEM certificate.

To keep example size down, the signing certificate is not included in the CMS encoding. The example certificate from [I-D.ietf-lamps-pq-composite-sigs] used to sign the CMS content is self-signed.

The following is an example of a signed-data with a single id-MLDSA65-ECDSA-P256-SHA512 signer, with signed attributes included:


```
-----BEGIN CMS-----
MIIOxQYJKoZIhvcNAQcCoIIOtjCCDrICAQExDTALBglghkgBZQMEAgMwVgYJKoZI
hvcNAQcBoEkER2lkLU1MRFNBNjUtRUNEU0EtUDI1Ni1TSEE1MTIgc2lnbmVkLWRh
dGEgZXhhbXBsZSB3aXRoIHNpZ25lZCBhdHRyaWJ1dGVzMYIORDCCDkACAQEwXjBG
MQ0wCwYDVQQKDARJRVRGMQ4wDAYDVQQLDAVMQU1QUzElMCMGA1UEAwwcaWQtTUxE
U0E2NS1FQ0RTQS1QMjU2LVNIQTUxMgIUW0MoLO0np7/Ch09mfDIxAm9wH3AwCwYJ
YIZIAWUDBAIDoIGJMBgGCSqGSIb3DQEJAzELBgkqhkiG9w0BBwEwHAYJKoZIhvcN
AQkFMQ8XDTI2MDEyMTIwMzkyMFowTwYJKoZIhvcNAQkEMUIEQIjYc0f2iK/i/r30
83ouERXhQHSSXulhH8t6jiLSUlMK6EbW5xNFsnRLbVI9PYdOvhVLqKaooVBrbVvx
iZPIX00wCgYIKwYBBQUHBi0Egg00EkQcFLL9GAh5+6zNBEQDr4xPJjTZjEgwgf0E
5kXiNLtzPJ0GMzrL/cUJz8LMhEEm1HMtLDWc3JOwSH5s5WyFRNSr9jhbDcwzHFDd
dZO3mAFBibQphfec+yU4E5CtDwBKR1yL0s4FjAQ2PuXVqRoxBy2z7fxCGrksY6C7
Zpesxy65vqnkTKmc2FF8D97Y+VsySeACY+3ZzTDZ1jEIApTzqqTiKy5mLpHeJ002
b4tdTtVdXa7JZT2bKlQwtSwOVnPEPfViCbCBklaH0K8oo+fGst6Xq+vd41V+XrLp
rPwned8k9ck8zJZq3YfVXzYxTwbpBjMiMyaMch0fs+umxe1ItgFZ3fJ5pI7Shiw+
j2ec7W7w1fmsXzF0ltk2wDwxIZF8g0qAiH3SOx7Gs2hfQnb0V9zhbjPt2NHiTM7r
QTTwvdPhxDQWl8uBcasPxOIDHnjrlcveghpXWgtSJAVm4wCTK55PXBshE2jFtbcC
xwXenC2pds4eH8J+zNwUeL9VTZg+l8WkXHvai7aFMlaWBSZK0fceBM/rIwAOP1gL
pBaurCr8nRxcjV7OKVDoRT+mEFPAjgk6Egyz+N1Pn8rjsY89aVizA7yIZmPU8qbG
BkBCVoJbqWYaHFY1wS+qFB9dIdOXK6rvqx1PO3yaAn8Yy1Sg3Wogie5m23hqysB+
j+G/VFwTBVc4DZTBPaHI7sPLEgCJGVX4saGhWjY7uUbK6VXTZxWrBmXUS3nka7OV
ZERgaIXlIBNYDEPAEClj6l4/WXhZOcA1HlnxEwvr+/4XyTWuOv9jFZL5W7qb7sEw
ug17x7xFVd5YZHvQtWkizGgAKjQj7oMjdF17vbnrTgowCOUpSnlCjJrWz5qst3r8
Mnjr0mFwv1Yv9S06EcCY62Bmks0eElfwwZAv8Je2IVtTidPft7AtdCg1o5Fjz4Ts
n4FbJ/quccY5e0g3FHjSRMXepW2nSov1yQuA9M+KNi0fllJLQDMg2yY3g7j5QReN
S1ue6/A2i0wsFLaRLOWsLyF0lZVf1VdLxI2UaxwQQOREHb9EXI4mgVKX76vyrvt1
tWO6yYlsNPeQ5+f4QiF2KSYo7/Gysb6fjOuLfT1s2mysefNNVSLXsfGM5ZGoSh1l
2veiQsLEGGqQ22b11ICzrzkzbCiM1SLFC41/4Dr8d0R/cSCEBAkOG0m16lHbkt16
J169myt5mBljy3s5QnRYEJqAFymimWerSsgwS/N8Mhem71QbM/0CZLHIXx1H91dO
dW9ZqLUfkuzN1RInouGNKPG7sW84eSWqrwc1JD4AKSs3lZQ+pUgF2DDl04Lc4ZN+
qjM5lTsuax5G3ywhfTvZVaKJL79u7f/2Wt9hZRNqc8NbsxkIpZ4HDrkclSJ6j/f4
```

```
C8fbhL5meVLHZ4dl2ZoVxcvQGNFA2NQMwlc3dlaEwRWDr4Mz8vTeBENt0Pzd5aW6
sLUuG3KWf95ONJFj0maO50RMPCRe8gnS/A2gR3tVp0+BVuJeD0DV8qGtaZScrtXr
eVRnvbeSfE5lxmOxji1UZ6rKxxQuzXjM8bY42fNqOktERXPCPewG+mBYfo0L+M1v
dmeRo4YKfye6HqeVBoVB7uIzB4TlGgFSMI+aLEYU9+lvXxbn9XxkKqj/Nxkuo4qi
qqmqO95Fpwhb86OmfJQ0xpR1d0o30eGysaj0Ii7u5VT5ni2gJCvblgL26QT8KfbL
pvEYZbq7RbsB/cbd9ldkAwVuTNkRWmyxeLr8SWuO7Uk3okAkJKTFn4zXUwUrgHLO
lH0xkfsWTvZBGqTNOMkc5hLeZnticjd4Agn60dMgybWWZ0NOl50xgwTbtZlh5dV9
xr6+CVZmYhAcB1srIS+4HF37KHm0O+c7bUDXvzGPNyt2DOz1UcnrJBcIhDU1iLvL
Yu8FMAdKlncqMRcO2C3KgpqpRMKQrOnwSuPqPHfiFAIuB2gdUDn1ob/+XqZ4UANR
nZHmXfn+n2bH+eT8EcEZ1XjeLyFeyJHRDR0aZyE0dKJyVA9a569YwHQYTpKVMEPp
9RtolkNBlMZYpQwvwhlg1Rcry4v6WamXKaagbOc3QQ48nNdZIyUFTiFAhGPReABe
uwVTUhFFNm0hqzcy7AKohg6ZPhYLdxrkfR3Q0bjMZvs6rqa17VvyzxQt0HDgZCJO
9H/X1BsOBnYlpjcnoU8WLPPKgOE/Nfvtipx8pC2V5Iq5v6ui7jHZ6G1TobI1x7ar
OIm5+nR0gS5R7gWx1UH4esEaVGDxkLOkTV1kxwq3BzRnl2LUrulod1sLVu1y/mVP
+g/uBq+G3lTWj486rTVCvqWj+P7iZXgsESobx7kMNlvCTeDaUPdAf4w0UHAa0tBH
eEkH0Piz08/iaWHD0uZ6tkQ1OpY3GxmWcvqol541X3EXJ5pdQA7DUz+2+KwqvzcN
wU799hOjHnxOXWc1+BWNVW4M4T/L2Y07+CW7KaKvH6Tq434aWeD0VhfuYCVioW7F
XO2yS9amsCuCcaw/nxSbMmdTEFt65ceLo22q57TtugbQCS5gECZKqA2vcyEKs+gd
Dx1HGS/9O3Xf9VpYvEaNW5NZib044A3gXRZbq3ndN2cZyUo3FF/XwsfPXrarBZug
xw7vnbdmdKkLb9Ig5nT0ZXZQV9cK5e6iUfbWR3s6+LvGPqSbiA1DQLM5A0tdybCD
yOBvuvTRsBJqUe13D9ypFnp3rut1LWdf8ZSxXVCwQVVGjyT1S//AsHVvGNGFQ8WD
U07fl37Qc9CD/HK12mdBRv/QQw3iMAD5hxHmz74bGIKmuNmsPPtcs/Itfmxovsuj
cPgwHsah8KgdSQv2somK9tx/Ba5iC9nLtm8DV3BPpUx55sv23lvVO5T+5PHl+XAK
TYedCVBsEfsQK4RlMbuRx3kFHO9FKGun/vAzEqxmTGr6alWmAtoyLJ7ZDEXrTZK+
ELduqCvwIHZ5Hoh6HzWLPzDma+JMK5X/BBVpKoBu0TvQl8x59En1+NYfHFFcl1Mi
jkQDHGlB56LBMXtwyJDn6iG52skJTzdNsFh3V4CgJSC8HXrmefPws1UGt0uXVNBC
EMUmSZTGBIA0Uw8tLBW9rBbklbBE3q+nU6fKASE4Yq2t+9eU5rLnMT51xfuXXVxD
KPBU2M0XPYJKVFwUIqa5KX4cJiMNnfupIVHZA1JXrz5tXNFmUhQcbtMl9sWBdPE+
vKwAnToBD3JYI68CURKGb8nkxJLT08Psmjz9gqYWJSJxlUyLsaCbrYz8vWivSFBn
cGw2hOYpwrmvwNrK+JehlcflRO3EPcUJYI7NRHt8soSV4J6LhrAWWXjQJKu7D12P
4KAStIulBHsyDbVm/Wm6lYbYnGOSWih7zxMhdNDU0tvls1whpZg3fod6mJ6crNcP
tKUjmlSf/MGrfsstwW7LiMwvkerA5HdhTLA9tZp+WmqJ9LSRhTOkXFlQwnZFlN3R
UVUIysYdH8qi/OjNdvDh0xp7Kanf1bIquMTPbg5sdJNKCVUjYLZ21aP3D2vHDDW9
CrN3KcFgd+uQUaEvWdr0VgJaK8VNFtH+5jZOdPyQjJGy2Gp4t53FtELeaQtNZmbC
mL07uESIO78AoDXdVN+G3q56OoaYvhAkOjUXnqTxlIyRpRrZycxg3yqYwKRrMkWs
EoqWdmqh8zsInNTGYsAdo+Dol7pY9Smk2nMVPRZHTDCmoNCdezrzFN5K0RydBwMK
ozvfSkBEkW7UqahNU3BhtpkBX72YYcHMYWMZNI77rYWRDjL99c9rMQ9LRuynYGF5
KD2jMf9NLkelelJ8g93JTRY1AdKs1viOvQ3j/lLjuen8B420Pb0ueLJSiQr+dego
C+jN4qM4auC4W+zwxxbnTviSUDPPxKGSIEFrLrQqVfxqUrXI/KYB1KgLl8Se4p50
Ho/58ZYcr3kKuAJ8QVJAX0cQCyqio9jyJUZDb9Cc85GNp61B8UghTqfgOReuFAd8
dvXz20ZLfTxXghljKrv/bWRSjVF2X7hLihOiaNQJAgx+pFXaehh5lTdFckUMDzLU
1J8ktL6m6VhcdRh9enogOLkvr4DM4o/1j48fIgPIurtYX+kKuxWRL088y7wsU4np
awjk+6c8Uy7q4NdrhuwME1fBeRy1LxsAaMMK2M8ONTIriCgbQnJfEN4WL0ir+iqB
wb5hzXLqXJlxz1Wnfs5ZR9L3OhQ1h9sIeihfFfNWnoLlF0q2pMnEMgLv+Mcbx7xQ
ZFCjOjnWQEKedzt9jM+V+Nql1nQCdZfnfgm2AvnCWLyeqID4HT1wV0mvWyLKFSto
SmjV2pEFxtYGftq12sD68w2VzBzn1SyGSwTva0hFn1KntRoVGXjAi17V9i3lrgCU
wrbiH8QjLdbX5wAab36iusocbo6Wve3zQ2yd0uwGFihITTk8VX3d7/QAAAAAAAAA
AAAAAAAAAAAAAAABQwTGB0kMEUCIQCutiFHcWBuVepcqbpkStOw7ngMGdeqAx4P
```

pWoMuOAdkAIgXpG9PLXn/nqA6YMPvNnIi7zQDA3ucqzgF7pkjMN+CrY=
-----END CMS-----

```
SEQUENCE {
  # signedData
  OBJECT_IDENTIFIER { 1.2.840.113549.1.7.2 }
  [0] {
    SEQUENCE {
      INTEGER { 1 }
      SET {
        SEQUENCE {
          # sha512
          OBJECT_IDENTIFIER { 2.16.840.1.101.3.4.2.3 }
        }
      }
      SEQUENCE {
        # data
        OBJECT_IDENTIFIER { 1.2.840.113549.1.7.1 }
        [0] {
          OCTET_STRING { "id-MLDSA65-ECDSA-P256-SHA512 signed-da
ta example with signed attributes" }
        }
      }
      SET {
        SEQUENCE {
          INTEGER { 1 }
          SEQUENCE {
            SEQUENCE {
              SET {
                SEQUENCE {
                  # organizationName
                  OBJECT_IDENTIFIER { 2.5.4.10 }
                  UTF8String { "IETF" }
                }
              }
              SET {
                SEQUENCE {
                  # organizationUnitName
                  OBJECT_IDENTIFIER { 2.5.4.11 }
                  UTF8String { "LAMPS" }
                }
              }
              SET {
                SEQUENCE {
                  # commonName
                  OBJECT_IDENTIFIER { 2.5.4.3 }
                  UTF8String { "id-MLDSA65-ECDSA-P256-SHA512" }
                }
```

```
              }
            }
            INTEGER { ‘5b43282ced27a7bfc2874f667c3231026f701f70‘
    }
          }
          SEQUENCE {
            # sha512
            OBJECT_IDENTIFIER { 2.16.840.1.101.3.4.2.3 }
          }
          [0] {
            SEQUENCE {
              # contentType
              OBJECT_IDENTIFIER { 1.2.840.113549.1.9.3 }
              SET {
                # data
                OBJECT_IDENTIFIER { 1.2.840.113549.1.7.1 }
              }
            }
            SEQUENCE {
              # signingTime
              OBJECT_IDENTIFIER { 1.2.840.113549.1.9.5 }
              SET {
                UTCTime { "260121203920Z" }
              }
            }
            SEQUENCE {
              # messageDigest
              OBJECT_IDENTIFIER { 1.2.840.113549.1.9.4 }
              SET {
                OCTET_STRING { ‘88d87347f688afe2febdf4f37a2e1115
   e14074925ee9611fcb7a8e22d252530ae846d6e71345b2744b6d523d3d874ebe
   154ba8a6a8a1506b6d5bf18993c85f4d‘ }
              }
            }
          }
          SEQUENCE {
            OBJECT_IDENTIFIER { 1.3.6.1.5.5.7.6.45 }
          }
          OCTET_STRING { ‘12441c14b2fd180879fbaccd044403af8c4f26
   34d98c483081fd04e645e234bb733c9d06333acbfdc509cfc2cc844126d4732d
   2c359cdc93b0487e6ce56c8544d4abf6385b0dcc331c50dd7593b798014189b4
   2985f79cfb25381390ad0f004a475c8bd2ce058c04363ee5d5a91a31072db3ed
   fc421ab92c63a0bb6697acc72eb9bea9e44ca99cd8517c0fded8f95b3249e002
   63edd9cd30d9d631080294f3aaa4e22b2e662e91de274d366f8b5d4ed55d5dae
   c9653d9b2a5430b52c0e5673c43df56209b081925687d0af28a3e7c6b2de97ab
   ebdde3557e5eb2e9acfc2779df24f5c93ccc966add87d55f36314f06e9063322
   33268c721d1fb3eba6c5ed48b60159ddf279a48ed2862c3e8f679ced6ef0d5f9
   ac5f317496d936c03c3121917c834a80887dd23b1ec6b3685f4276f457dce16e
```

33edd8d1e24cceeb4134f0bdd3e1c4341697cb8171ab0fc4e2031e78eb95cbde
821a575a0b52240566e300932b9e4f5c1b211368c5b5b702c705de9c2da976ce
1e1fc27eccdc1478bf554d983e97c5a45c7bda8bb68532569605264ad1f71e04
cfeb23000e3f580ba416aeac2afc9d1c5c8d5ece2950e8453fa61053c08e093a
120cb3f8dd4f9fcae3b18f3d6958b303bc886663d4f2a6c606404256825ba966
1a1c5635c12faa141f5d21d3972baaefab1d4f3b7c9a027f18cb54a0dd6a2089
ee66db786acac07e8fe1bf545c130557380d94c13da1c8eec3cb1200891955f8
b1a1a15a363bb946cae955d36715ab0665d44b79e46bb3956444606885e52013
580c43c0102963ea5e3f59785939c0351e59f1130bebfbfe17c935ae3aff6315
92f95bba9beec130ba0d7bc7bc4555de58647bd0b56922cc68002a3423ee8323
745d7bbdb9eb4e0a3008e5294a79428c9ad6cf9aacb77afc3278ebd26170bf56
2ff52d3a11c098eb606692cd1e1257f0c1902ff097b6215b5389d3dfb7b02d74
2835a39163cf84ec9f815b27faae71c6397b48371478d244c5dea56da74a8bf5
c90b80f4cf8a362d1f96524b403320db263783b8f941178d4b5b9eebf0368b4c
2c14b6912ce5ac2f217495955fd5574bc48d946b1c1040e4441dbf445c8e2681
5297efabf2aefb75b563bac9896c34f790e7e7f8422176292628eff1b2b1be9f
8ceb8b7d3d6cda6cac79f34d5522d7b1f18ce591a84a1d65daf7a242c2c4186a
90db66f5d480b3af39336c288cd522c50b8d7fe03afc77447f71208404090e1b
49b5ea51db92dd7a275ebd9b2b79981963cb7b39427458109a801729a29967ab
4ac8304bf37c3217a6ef541b33fd0264b1c85f1d47f7574e756f59a8b51f92ec
cdd51227a2e18d28f1bbb16f387925aaaf0735243e00292b3795943ea54805d8
30e5d382dce1937eaa3339953b2e6b1e46df2c217d3bd955a2892fbf6eedfff6
5adf6165136a73c35bb31908a59e070eb91c95227a8ff7f80bc7db84be667952
c7678765d99a15c5cbd018d140d8d40cc25737765684c11583af8333f2f4de04
436dd0fcdde5a5bab0b52e1b72967fde4e349163d2668ee7444c3c245ef209d2
fc0da0477b55a74f8156e25e0f40d5f2a1ad69949caed5eb795467bdb7927c4e
65c663b18e2d5467aacac7142ecd78ccf1b638d9f36a3a4b444573c23dec06fa
60587e8d0bf8cd6f766791a3860a7f27ba1ea795068541eee2330784e51a0152
308f9a2c4614f7e96f5f16e7f57c642aa8ff37192ea38aa2aaa9aa3bde45a708
5bf3a3a67c9434c69475774a37d1e1b2b1a8f4222eeee554f99e2da0242bdb96
02f6e904fc29f6cba6f11865babb45bb01fdc6ddf6576403056e4cd9115a6cb1
78bafc496b8eed4937a2402424a4c59f8cd753052b8072ce947d3191fb164ef6
411aa4cd38c91ce612de667b627237780209fad1d320c9b59667434e979d3183
04dbb59961e5d57dc6bebe09566662101c075b2b212fb81c5dfb2879b43be73b
6d40d7bf318f372b760cecf551c9eb24170884353588bbcb62ef0530074a9677
2a31170ed82dca829aa944c290ace9f04ae3ea3c77e214022e07681d5039f5a1
bffe5ea6785003519d91e65df9fe9f66c7f9e4fc11c119d578de2f215ec891d1
0d1d1a67213474a272540f5ae7af58c074184e92953043e9f51b6896434194c6
58a50c2fc21960d5172bcb8bfa59a99729a6a06ce737410e3c9cd7592325054e
21408463d178005ebb0553521145366d21ab3732ec02a8860e993e160b771ae4
7d1dd0d1b8cc66fb3aaea6b5ed5bf2cf142dd070e064224ef47fd7d41b0e0676
25a63727a14f162cf3ca80e13f35fbed8a9c7ca42d95e48ab9bfaba2ee31d9e8
6d53a1b235c7b6ab3889b9fa7474812e51ee05b1d541f87ac11a5460f190b3a4
4d5d64c70ab70734679762d4aee968775b0b56ed72fe654ffa0fee06af86de54
d68f8f3aad3542bea5a3f8fee265782c112a1bc7b90c365bc24de0da50f7407f
8c3450701ad2d047784907d0f8b3d3cfe26961c3d2e67ab644353a96371b1996
72faa8979e355f7117279a5d400ec3533fb6f8ac2abf370dc14efdf613a31e7c
4e5d6735f8158d556e0ce13fcbd98d3bf825bb29a2af1fa4eae37e1a59e0f456

17ee602562a16ec55cedb24bd6a6b02b8271ac3f9f149b326753105b7ae5c78b
a36daae7b4edba06d0092e6010264aa80daf73210ab3e81d0f1d47192ffd3b75
dff55a58bc468d5b935989bd38e00de05d165bab79dd376719c94a37145fd7c2
c7cf5eb6ab059ba0c70eef9db76674a90b6fd220e674f465765057d70ae5eea2
51f6d6477b3af8bbc63ea49b880d4340b339034b5dc9b083c8e06fbaf4d1b012
6a51ed770fdca9167a77aeeb752d675ff194b15d50b04155468f24f54bffc0b0
756f18d18543c583534edf977ed073d083fc72b5da674146ffd0430de23000f9
8711e6cfbe1b1882a6b8d9ac3cfb5cb3f22d7e6c68becba370f8301ec6a1f0a8
1d490bf6b2898af6dc7f05ae620bd9cbb66f0357704fa54c79e6cbf6de5bd53b
94fee4f1e5f9700a4d879d09506c11fb102b846531bb91c779051cef45286ba7
fef03312ac664c6afa6a55a602da322c9ed90c45eb4d92be10b76ea82bf02076
791e887a1f358b3f30e66be24c2b95ff0415692a806ed13bd097cc79f449f5f8
d61f1c515c9753228e44031c6941e7a2c1317b70c890e7ea21b9dac9094f374d
b058775780a02520bc1d7ae679f3f0b35506b74b9754d04210c5264994c60480
34530f2d2c15bdac16e495b044deafa753a7ca01213862adadfbd794e6b2e731
3e75c5fb975d5c4328f054d8cd173d824a545c1422a6b9297e1c26230d9dfba9
2151d9035257af3e6d5cd16652141c6ed325f6c58174f13ebcac009d3a010f72
5823af025112866fc9e4c492d3d3c3ec9a3cfd82a616252271954c8bb1a09bad
8cfcbd68af485067706c3684e629c2b9afc0dacaf897a195c7e544edc43dc509
608ecd447b7cb28495e09e8b86b0165978d024abbb0f5d8fe0a012b48ba5047b
320db566fd69ba9586d89c63925a287bcf132174d0d4d2dbe5b35c21a598377e
877a989e9cacd70fb4a5239a549ffcc1ab7ecb2dc16ecb88cc2f91eac0e47761
4cb03db59a7e5a6a89f4b4918533a45c5950c2764594ddd1515508cac61d1fca
a2fce8cd76f0e1d31a7b29a9dfd5b22ab8c4cf6e0e6c74934a09552360b676d5
a3f70f6bc70c35bd0ab37729c16077eb9051a12f59daf456025a2bc54d16d1fe
e6364e74fc908c91b2d86a78b79dc5b442de690b4d6666c298bd3bb844883bbf
00a035dd54df86deae7a3a8698be10243a35179ea4f1948c91a51ad9c9cc60df
2a98c0a46b3245ac128a96766aa1f33b089cd4c662c01da3e0e897ba58f529a4
da73153d16474c30a6a0d09d7b3af314de4ad11c9d07030aa33bdf4a4044916e
d4a9a84d537061b699015fbd9861c1cc616319348efbad85910e32fdf5cf6b31
0f4b46eca7606179283da331ff4d2e47a57a527c83ddc94d163501d2acd6f88e
bd0de3fe52e3b9e9fc078db43dbd2e78b252890afe75e8280be8cde2a3386ae0
b85becf0c716e74ef8925033cfc4a19220416b2eb42a55fc6a52b5c8fca601d4
a80b97c49ee29e741e8ff9f1961caf790ab8027c4152405f47100b2aa2a3d8f2
2546436fd09cf3918da7ad41f148214ea7e03917ae14077c76f5f3db464b7d3c
578219632abbff6d64528d51765fb84b8a13a268d409020c7ea455da7a187995
374572450c0f32d4d49f24b4bea6e9585c75187d7a7a2038b92faf80cce28ff5
8f8f1f2203c8babb585fe90abb15912f4f3ccbbc2c5389e96b08e4fba73c532e
eae0d76b86ec0c1357c1791cb52f1b0068c30ad8cf0e35322b88281b42725f10
de162f48abfa2a81c1be61cd72ea5c9971cf55a77ece5947d2f73a143587db08
7a285f15f3569e82e5174ab6a4c9c43202eff8c71bc7bc506450a33a39d64042
9e773b7d8ccf95f8daa5d674027597e77e09b602f9c258bc9ea880f81d3d7057
49af5b22ca152b684a68d5da9105c6d6067edab5dac0faf30d95cc1ce7d52c86
4b04ef6b48459f52a7b51a151978c08b5ed5f62de5ae0094c2b6e21fc4232dd6
d7e7001a6f7ea2baca1c6e8e96bdedf3436c9dd2ec061628484d393c557dddef
f400000000000000000000000000000000000000050c13181d243045022100ae
b6214771606e55ea5ca9ba644ad3b0ee780c19d7aa031e0fa56a0cb8e01d9002
205e91bd3cb5e7fe7a80e9830fbcd9c88bbcd00c0dee72ace017ba648cc37e0a

```
   b6‘ }
           }
         }
       }
     }
   }
```

Acknowledgements

The authors wish to thank Piotr Popis for his valuable feedback on this document.

Thanks to the co-authors of [RFC9882], Ben Salter and Adam Raine, this document borrows heavily from that one. "Copying always makes things easier and less error prone" - [RFC8411].

Authors’ Addresses

Mike Ounsworth
Entrust Limited
2500 Solandt Road 褰Suite 100
Ottawa, Ontario K2K 3G5
Canada
Email: mike.ounsworth@entrust.com

John Gray
Entrust Limited
2500 Solandt Road Suite 100
Ottawa, Ontario K2K 3G5
Canada
Email: john.gray@entrust.com

Jan Klaussner
Bundesdruckerei GmbH
Kommandantenstr. 18
10969 Berlin
Germany
Email: jan.klaussner@bdr.de

Daniel Van Geest
CryptoNext Security
16, Boulevard Saint-Germain
75007 Paris
France
Email: daniel.vangeest@cryptonext-security.com