

LAMPS  
Internet-Draft  
Intended status: Standards Track  
Expires: 9 August 2026

M. Ounsworth  
J. Gray  
Entrust  
J. Klaussner  
Bundesdruckerei GmbH  
D. Van Geest  
CryptoNext Security  
5 February 2026

Composite ML-DSA for use in Cryptographic Message Syntax (CMS)  
draft-ietf-lamps-cms-composite-sigs-04

## Abstract

Composite ML-DSA defines combinations of ML-DSA 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

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 9 August 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 . . . . .	3
1.1. ASN.1 . . . . .	3
1.2. Conventions and Terminology . . . . .	3
2. Composite ML-DSA Algorithm Identifiers . . . . .	3
3. Signed-Data Conventions . . . . .	4
3.1. Pre-Hashing . . . . .	5
3.2. SignedData digestAlgorithms . . . . .	5
3.3. Signature Generation and Verification . . . . .	5
3.4. SignerInfo Content . . . . .	6
4. ASN.1 Module . . . . .	8
5. IANA Considerations . . . . .	10
6. Security Considerations . . . . .	10
7. References . . . . .	10
7.1. Normative References . . . . .	10
7.2. Informative References . . . . .	12
Appendix A. Examples . . . . .	12
Acknowledgements . . . . .	19
Authors' Addresses . . . . .	19

## 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. ASN.1

CMS values are generated using ASN.1 [X680], using the Basic Encoding Rules (BER) and the Distinguished Encoding Rules (DER) [X690].

### 1.2. 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

The same AlgorithmIdentifier is used to identify a Composite ML-DSA public key and signature algorithm. The object identifiers for Composite ML-DSA algorithms are defined in [I-D.ietf-lamps-pq-composite-sigs], and are reproduced here for convenience. The parameters field of the AlgorithmIdentifier for the Composite ML-DSA public key and signature algorithm MUST be absent.

```
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, e.g., in the case of id-MLDSA87-ECDSA-P384-SHA512 the traditional component uses SHA-384. These internal 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. When signing with a Composite ML-DSA algorithm, this list of identifiers SHOULD include the corresponding digest algorithm from Table 1.

### 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 use 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<sup>テ</sup>瘍<sup>er</sup>, 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>>.
- [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>>.
- [X690] ITU-T, "Information technology - Abstract Syntax Notation One (ASN.1): ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ITU-T Recommendation X.690, ISO/IEC 8825-1:2021, February 2021, <<https://www.itu.int/rec/T-REC-X.690>>.
- [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>>.
- [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-----

MIIIOxQYJKoZIhvcNAQcCoIIOTjCCDrICAQExDTALBglghkgBZQMEAgMwVgYJKoZI  
hvcNAQcBoEker2lkLU1MRFNBNjUtrUNEU0EtUDI1Ni1TSEE1MTIgc2lnbmVklWRh  
dGEgZXhhbXBsZSB3aXR0IHNPZ25lZCBhdHRyaWJldGVzMYIORDCCDkACAQEwXjBG  
MQ0wCwYDVQQKDARJRVRGMQ4wDAYDVQQQLDAVMQU1QUzElMCMGA1UEAwucaWQtTUxE  
U0E2NS1FQ0RTQS1QMjU2LVNIQTUxMGIUW0MoLO0np7/Ch09mfDIxAm9wH3AwCwYJ  
YIZIAWUDBAIDoIGJMBGCSqGSIb3DQEJAzELBgkqhkiG9w0BBwEwHAYJKoZIhvcN  
AQkFMQ8XDTE2MDEyMTIwMzkyMFowTwYJKoZIhvcNAQkEMUIEQIjYc0f2iK/i/r30  
83ouERXhQHSSXulhH8t6jiLSULMK6EbW5xNFSnRLbVI9PYdOvhVLqKaooVBrbVvx  
iZPIX00wCgYIKwYBBQUHBI0Egg00EkQcFLL9GAh5+6zNBEQDr4xPJjTZjEgwgf0E  
5kXiNLtzPJ0GMzrL/cUJz8LMhEEm1HMTLDWc3JOwSH5s5WyFRNSr9jhbDcwzHFDd  
dZO3mAFBibQphfec+yU4E5CtDwBKRlyL0s4FjAQ2PuXVqRoxBy2z7fxCGrksY6C7  
Zpesxy65vqnkTKmc2FF8D97Y+VsySeACY+3ZzTDZ1jEIApTzqqTiKy5mLpHeJ002  
b4tdTtVdXa7JZT2bKlQwtSwOVnPEPfiCbCBklaH0K8oo+fGst6Xq+vd41V+XrLp  
rPwned8k9ck8zJZq3YfVXzYxTwbpBjMiMyaMch0fs+umxe1ItgFZ3fJ5pI7Shiw+  
j2ec7W7w1fmsXzF0ltk2wDwxIZF8g0qAiH3S0x7Gs2hfQnb0V9zhbjPt2NHiTM7r  
QTTwvdPhxDQWl8uBcasPxOIDHnjrlcveghpXWgtSJAVm4wCTK55PXBshE2jftbcC  
xwXenC2pds4eH8J+zNwUeL9VTZg+l8WkXHvai7aFmlaWBSZK0fceBM/rIwAOPlgL  
pBaurCr8nRxcjV7OKVDoRT+mEFPAjgk6Egyz+N1Pn8rjsY89aVizA7yIZmPU8qbG  
BkBCVoJbqWYaHFYlwS+qFB9dIdOXK6rvqx1PO3yaAn8Yy1Sg3Wogie5m23hqysB+  
j+G/VFwTBVc4DZTBPahi7sPLEgCJGVX4saGhWjY7uUbK6VXTZxWrBmXUS3nka7OV  
ZERgaIXlIBNYDEPAEC1j6l4/WXhZ0cAlHlnxEwvr+/4XyTWuOv9jFZL5W7qb7sEw  
ugl7x7xFVd5YZHvQtWkizGgAKjQj7oMjdf17vbnrTgowCOUpSn1CjJrWz5qst3r8  
Mnjr0mFwv1Yv9S06EcCY62Bmks0eElfwZAv8Je2IVtTidPft7AtdCg1o5Fjz4Ts  
n4FbJ/quccY5e0g3FHjSRMXepW2nSovlyQuA9M+KNi0fllJLQDMg2yY3g7j5QReN  
Slue6/A2i0wsFLaRLOWsLyF0lZVf1VdLxI2UaxwQQOREHb9EXI4mgVKX76vyrvt1  
tWO6yYlsNPeQ5+f4QiF2KSYo7/Gysb6fjOuLft1s2mysefNNVSLXsfGM5ZGoSh11  
2veiQsLEGGQq22b11ICzrzkbzCim1SLFC41/4Dr8d0R/cSCEBAKOG0m16lHbkt16  
Jl69myt5mBljy3s5QnRYEJqAFymimWerSsgwS/N8Mhem71QbM/0CZLHIXx1H91dO  
dW9ZqLUfkuzN1RinouGNKPG7sW84eSwwqrwclJD4AKSs3lZQ+pUgF2DD104Lc4ZN+  
qjM5lTsuax5G3ywhfTvZVaKJL79u7f/2Wt9hZRNqc8NbsxkIpZ4HDrkclSJ6j/f4  
C8fbbL5meVLHZ4dl2ZoVxcvQGNFA2NQMWlc3dlaEwRWDr4Mz8vTeBENT0Pzd5aW6  
sLUuG3KWf95ONJFj0ma050RMPCRe8gnS/A2gR3tVp0+BVuJeD0DV8qGtaZScrtXr  
eVRnvbeSfE5lxmOxji1UZ6rKxxQuzXjM8bY42fNqOktERXPCPewG+mBYfo0L+Mlv  
dmeRo4YKfye6HqeVBoVB7uIzB4TlGgFSMI+aLEYU9+lvXxbn9XxkKqj/Nxkuo4qi

qqmq095Fpwhb86OmfJQ0xpRld0o30eGysaj0Ii7u5VT5ni2gJCvblgL26QT8KfbL  
pvEYZbq7RbsB/cbd9ldkAwVuTNkRWmyxeLr8SWu07Uk3okAkJKTFn4zXUwUrgHLO  
lH0xkfsWTvZBGqTNOMkc5hLeZnticjd4Ag60dMgybWWZ0N0150xgwTbtZlh5dV9  
xr6+CVZmYhAcBlsrIS+4HF37KHm00+c7bUDXvzGPNyt2D0z1UcnrJBcIhDUliLvL  
Yu8FMAdKlncqMRcO2C3KgpqRMKQRonwSuPqPHfiFAIuB2gdUDnlob/+XqZ4UANR  
nZHmXfn+n2bH+eT8EcEZlXjeLyFeyJHRDR0aZyE0dKJyVA9a569YwHQYTpkVMEpp  
9RtolKNBlMZYPQwvwhlg1Rcry4v6WamXKaagbOc3QQ48nNdZiYUFTiFAhGPREABe  
uwVTUhFFNm0hgzcy7AKohg6ZPhYLdxrkfr3Q0bjMZvs6rqal7VvyzxQt0HDgZCJO  
9H/XlBsOBnYlpcnoU8WLPKGOE/Nfvtipx8pC2V5Iq5v6ui7jHZ6G1TobIlx7ar  
OIm5+nR0gS5R7gWxlUH4esEaVGDXkLOkTVlKxwq3BzRnl2LUrulodlsLVuly/mVP  
+g/uBq+G3lTWj486rTVCvqWj+P7iZXgsESobx7kMnLvCTeDaUPdAf4w0UHAa0tBH  
eEKH0Piz08/iaWHD0uZ6tkQ1OpY3GxmWcvqol541X3EXJ5pdQA7DUz+2+KwqvzcN  
wU799h0jHnxOXWcl+BWNVW4M4T/L2Y07+CW7KaKvH6Tq434aWeD0VhfuYCVioW7F  
XO2yS9amsCuCaw/nxSbMmdTEFT65ceLo22q57TtugbQCS5gECZKqA2vcyEKs+gd  
Dx1HGS/903Xf9VpYvEaNW5NZib044A3gXRZbq3ndN2cZyUo3FF/XwsfPXrarBZug  
xw7vnbddmKkLb9Ig5nT0ZXZQV9cK5e6iUfbWR3s6+LvGPqSbiA1DQLM5A0tdybCD  
yOBvuvTRsBJqUel3D9ypFnp3rut1LWdf8ZSxXVCwQVVGjyTlS//AshVvGNGFQ8WD  
U07f137Qc9CD/HK12mdBRv/QQw3iMAD5hxHmz74bGIKmuNmsPPTcs/Itfmxovsu  
cPgWhsah8KgdSQv2somK9tx/Ba5iC9nLtm8DV3BPpUx55sv23lvVO5T+5PH1+XAK  
TYedCVBsEfsQK4RlMbuR3kFHO9FKGun/vAzEqxmTGr6alWmAtoyLj7ZDEXrTZK+  
ELduqCvwiHZ5Hoh6HzWLPzDma+JMK5X/BBVpKoBu0TvQl8x59En1+NYfHFFcl1Mi  
jkQDHGLB56LBMXtwyJDn6iG52skJTzdNsFh3V4CgJSC8HXrmefPws1UGt0uXVNBC  
EMUmSZTGBIA0Uw8tLBW9rBbklbBE3q+nU6fKASE4Yq2t+9eU5rLnMT5lxfuXXVxD  
KPBU2M0XPYJKVfWUIqa5KX4cJiMnNfupIVHZA1JXrz5tXNFmUhQcbtMl9sWBdPE+  
vKwAnToBD3JYI68CURKGB8nkxJLT08Psmjz9gqYWJSJxlUyLsaCbrYz8vWivSFBn  
cGw2hOYpwrvmvNrK+Jehlcf1RO3EPcUJYI7NRHt8soSV4J6LhrAWWXjQJKu7D12P  
4KASTiulBHsyDbVm/Wm6lYbYnGOSwih7zxMhdNDU0tvls1whpZg3fod6mJ6crNcP  
tKUjmlSf/MGrfsstW7LiMwvkerA5HdhTLA9tZp+WmqJ9LSRhT0kXf1QwnZFln3R  
UVUIysYdH8qi/OjNdvdh0xp7KanflbIquMTPbg5sdJNKCVUjYLZ21aP3D2vHDDW9  
CrN3KcFgd+uQUaEvWdr0VgJaK8VNfTH+5jZODPyQjJGy2Gp4t53FtELeaQtNZmbC  
mL07uESIO78AoDXdVN+G3q56OoaYvhAkOjUXnqTx1IyRpRrZycxg3yqYwKRRmKWs  
EoqWdmqh8zsInNTGYSAdo+Dol7pY9Smk2nMVPRZHTDCmoNCdezrzFN5K0RydBwMK  
ozvfSkBEKw7UqahNU3BhtpkBX72YYCHMYWMZNI77rYWRDjL99c9rMQ9LRuynYGF5  
KD2jMf9NLkelelJ8g93JTRYlAdKslviOvQ3j/1Ljuen8B420Pb0ueLJSiQr+dego  
C+jN4qM4auC4W+zwxxbnTviSUDPPxKGSIEFrLrQqVfxqUrXI/KYB1KgLL8Se4p50  
Ho/58ZYcr3kKuAJ8QVJAX0cQCyqio9jyJUZDb9Cc85GNp61B8UghTqfgOREuFAd8  
dvXz20ZLftXxghlJkrv/bWRSjVF2X7hLihOiaNQJAgx+pFXaehh51TdFckUMDzLU  
1J8ktL6m6VhcdRh9enogOLkvr4DM4o/1j48fIgPIurtYX+kKuxWRL088y7wsU4np  
awjk+6c8Uy7q4NdrhuwME1fBeRylLxsAaMMK2M8ONTIriCgbQnJfEN4WL0ir+iqB  
wb5hzXLqXJlxzlWnfs5ZR9L3OhQ1h9sIeihfFFNWnoLlF0q2pMnEMgLv+Mcbx7xQ  
ZFCjOjnWQEKedzt9jM+V+Nql1nQCdZfnfgm2AvnCWLyeqID4HT1wV0mvWyLKFSto  
SmjV2pEFxtYGftq12sD68w2VzBzn1SyGSwTva0hFn1KntRoVGXjAi17V9i3lrgCU  
wrbiH8QjLdbX5wAab36iusocbo6Wve3zQ2yd0uwGFihITTk8VX3d7/QAAAAAAAAA  
AAAAAAAAAAAAAAAAABQWtGB0kMEUCIQCutifHcWBUvepcqbKStOw7ngMGdeqAx4P  
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 { '88d87347f688afe2febd4f37a2e1115
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
33edd8d1e24ccee4134f0bdd3e1c4341697cb8171ab0fc4e2031e78eb95cbde
821a575a0b52240566e300932b9e4f5c1b211368c5b5b702c705de9c2da976ce
1e1fc27eccdc1478bf554d983e97c5a45c7bda8bb68532569605264ad1f71e04

```



cfeb23000e3f580ba416aeac2afc9d1c5c8d5ece2950e8453fa61053c08e093a  
120cb3f8dd4f9fcae3b18f3d6958b303bc886663d4f2a6c606404256825ba966  
1alc5635c12faa141f5d21d3972baaefab1d4f3b7c9a027f18cb54a0dd6a2089  
ee66db786acac07e8felbf545c130557380d94c13da1c8eec3cb1200891955f8  
b1ala15a363bb946cae955d36715ab0665d44b79e46bb3956444606885e52013  
580c43c0102963ea5e3f59785939c0351e59f1130bebfbfel17c935ae3aff6315  
92f95bba9beec130ba0d7bc7bc4555de58647bd0b56922cc68002a3423ee8323  
745d7bbdb9eb4e0a3008e5294a79428c9ad6cf9aacb77afc3278ebd26170bf56  
2ff52d3a11c098eb606692cd1e1257f0c1902ff097b6215b5389d3dfb7b02d74  
2835a39163cf84ec9f815b27faae71c6397b48371478d244c5dea56da74a8bf5  
c90b80f4cf8a362d1f96524b403320db263783b8f941178d4b5b9eebf0368b4c  
2c14b6912ce5ac2f217495955fd5574bc48d946b1c1040e4441dbf445c8e2681  
5297efabf2aefb75b563bac9896c34f790e7e7f8422176292628eff1b2b1be9f  
8ceb8b7d3d6cda6cac79f34d5522d7b1f18ce591a84ald65daf7a242c2c4186a  
90db66f5d480b3af39336c288cd522c50b8d7fe03afc77447f71208404090elb  
49b5ea51db92dd7a275ebd9b2b79981963cb7b39427458109a801729a29967ab  
4ac8304bf37c3217a6ef541b33fd0264b1c85f1d47f7574e756f59a8b51f92ec  
cdd51227a2e18d28f1bbb16f387925aaaf0735243e00292b3795943ea54805d8  
30e5d382dcel937eaa3339953b2e6b1e46df2c217d3bd955a2892fbf6eedfff6  
5adf6165136a73c35bb31908a59e070eb91c95227a8ff7f80bc7db84be667952  
c7678765d99a15c5cbd018d140d8d40cc25737765684c11583af8333f2f4de04  
436dd0fcdde5a5bab0b52elb72967fde4e349163d2668ee7444c3c245ef209d2  
fc0da0477b55a74f8156e25e0f40d5f2alad69949caed5eb795467bdb7927c4e  
65c663b18e2d5467aacac7142ecd78ccf1b638d9f36a3a4b444573c23dec06fa  
60587e8d0bf8cd6f766791a3860a7f27ba1ea795068541eee2330784e51a0152  
308f9a2c4614f7e96f5f16e7f57c642aa8ff37192ea38aa2aaa9aa3bde45a708  
5bf3a3a67c9434c69475774a37d1elb2b1a8f4222eeee554f99e2da0242bdb96  
02f6e904fc29f6cba6f11865babb45bb01fdc6ddf6576403056e4cd9115a6cb1  
78bafc496b8eed4937a2402424a4c59f8cd753052b8072ce947d3191fb164ef6  
411aa4cd38c91ce612de667b627237780209fad1d320c9b59667434e979d3183  
04dbb59961e5d57dc6bebe09566662101c075b2b212fb81c5dfb2879b43be73b  
6d40d7bf318f372b760cecf551c9eb24170884353588bbcb62ef0530074a9677  
2a31170ed82dca829aa944c290ace9f04ae3ea3c77e214022e07681d5039f5a1  
bffe5ea6785003519d91e65df9fe9f66c7f9e4fc11c119d578de2f215ec891d1  
0d1d1a67213474a272540f5ae7af58c074184e92953043e9f51b6896434194c6  
58a50c2fc21960d5172bcb8bfa59a99729a6a06ce737410e3c9cd7592325054e  
21408463d178005ebb0553521145366d21ab3732ec02a8860e993e160b771ae4  
7d1dd0d1b8cc66fb3aaea6b5ed5bf2cf142dd070e064224ef47fd7d41b0e0676  
25a63727a14f162cf3ca80e13f35fbed8a9c7ca42d95e48ab9bfaba2ee31d9e8  
6d53a1b235c7b6ab3889b9fa7474812e51ee05b1d541f87ac11a5460f190b3a4  
4d5d64c70ab70734679762d4aee968775b0b56ed72fe654ffa0fee06af86de54  
d68f8f3aad3542bea5a3f8fee265782c112a1bc7b90c365bc24de0da50f7407f  
8c3450701ad2d047784907d0f8b3d3cfe26961c3d2e67ab644353a96371b1996  
72faa8979e355f7117279a5d400ec3533fb6f8ac2abf370dc14efdf613a31e7c  
4e5d6735f8158d556e0cel3fcbd98d3bf825bb29a2af1fa4eae37e1a59e0f456  
17ee602562a16ec55cedb24bd6a6b02b8271ac3f9f149b326753105b7ae5c78b  
a36daae7b4edba06d0092e6010264aa80daf73210ab3e81d0f1d47192ffd3b75  
dff55a58bc468d5b935989bd38e00de05d165bab79dd376719c94a37145fd7c2



```
}  
}  
}
```

## Acknowledgements

The authors wish to thank Piotr Popis (Enigma) 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 Suite 100  
Ottawa, Ontario K2K 3G5  
Canada  
Email: [mike.ounsworth@entrust.com](mailto:mike.ounsworth@entrust.com)

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

Jan Klaussner  
Bundesdruckerei GmbH  
Kommandantenstr. 18  
10969 Berlin  
Germany  
Email: [jan.klaussner@bdr.de](mailto:jan.klaussner@bdr.de)

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