
Stream: Internet Engineering Task Force (IETF)
RFC: [9882](#)
Category: Standards Track
Published: October 2025
ISSN: 2070-1721
Authors:
B. Salter
UK National Cyber Security Centre
A. Raine
UK National Cyber Security Centre
D. Van Geest
CryptoNext Security

RFC 9882

Use of the ML-DSA Signature Algorithm in the Cryptographic Message Syntax (CMS)

Abstract

The Module-Lattice-Based Digital Signature Algorithm (ML-DSA), as defined by NIST in FIPS 204, is a post-quantum digital signature scheme that aims to be secure against an adversary in possession of a Cryptographically Relevant Quantum Computer (CRQC). This document specifies the conventions for using the ML-DSA signature algorithm with the Cryptographic Message Syntax (CMS). In addition, the algorithm identifier and public key syntax are provided.

Status of This Memo

This is an Internet Standards Track document.

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

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc9882>.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions

with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Conventions and Definitions	3
2. ML-DSA Algorithm Identifiers	3
3. Signed-Data Conventions	4
3.1. Pure Mode Versus Pre-Hash Mode	4
3.2. Signature Generation and Verification	4
3.3. SignerInfo Content	5
4. Security Considerations	6
5. Operational Considerations	7
6. IANA Considerations	7
7. References	8
7.1. Normative References	8
7.2. Informative References	8
Appendix A. ASN.1 Module	10
Appendix B. Examples	10
Acknowledgments	25
Authors' Addresses	25

1. Introduction

The Module-Lattice-Based Digital Signature Algorithm (ML-DSA) is a post-quantum digital signature algorithm standardised by the US National Institute of Standards and Technology (NIST) as part of their post-quantum cryptography standardisation process. It offers smaller signatures and significantly faster runtimes than SLH-DSA [FIPS205], an alternative post-quantum signature algorithm also standardised by NIST. This document specifies the use of the ML-DSA in the CMS at three security levels: ML-DSA-44, ML-DSA-65, and ML-DSA-87. See [Appendix B](#) of [RFC9881] for more information on the security levels and key sizes of ML-DSA.

Prior to standardisation, ML-DSA was known as Dilithium. ML-DSA and Dilithium are not compatible.

For each of the ML-DSA parameter sets, an algorithm identifier OID has been specified.

[FIPS204] also specifies a pre-hashed variant of ML-DSA, called HashML-DSA. Use of HashML-DSA in the CMS is not specified in this document. See [Section 3.1](#) for more details.

1.1. 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.

2. 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 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 ML-DSA public keys and signature algorithms. [RFC9881] describes the use of 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 ML-DSA are described below.

parameters: The parameters field contains parameter information for the algorithm identified by the OID in the algorithm field. Each 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 an ML-DSA AlgorithmIdentifier.

The object identifiers for ML-DSA are defined in the NIST Computer Security Objects Register [CSOR], and are reproduced here for convenience.

```
sigAlgs OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16)
    us(840) organization(1) gov(101) csor(3) nistAlgorithms(4) 3 }

id-ml-dsa-44 OBJECT IDENTIFIER ::= { sigAlgs 17 }

id-ml-dsa-65 OBJECT IDENTIFIER ::= { sigAlgs 18 }

id-ml-dsa-87 OBJECT IDENTIFIER ::= { sigAlgs 19 }
```

3. Signed-Data Conventions

3.1. Pure Mode Versus Pre-Hash Mode

[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], SLH-DSA [FIPS205] and ML-DSA have also been standardised, and these algorithms support both a "pure" and "pre-hash" mode. In the pre-hash mode, a message digest (the "pre-hash") is calculated separately and supplied to the signature algorithm as described above. In the pure mode, the message to be signed or verified is instead supplied directly to the signature algorithm. When EdDSA [RFC8419] and SLH-DSA [RFC9814] are used with CMS, only the pure mode of those algorithms is specified. This is because in most situations, CMS signatures are computed over a set of signed attributes that contain a hash of the content, rather than being computed over the message content itself. Since signed attributes are typically small, use of pre-hash modes in the CMS wouldn't significantly reduce the size of the data to be signed, and hence offers no benefit. This document follows that convention and does not specify the use of ML-DSA's pre-hash mode ("HashML-DSA") in the CMS.

3.2. 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. Each method produces a different "message digest" to be supplied to the signature algorithm in question, but because the pure mode of ML-DSA is used, the "message digest" is in fact the entire message. 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, ML-DSA (pure mode) 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, ML-DSA (pure mode) 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.

[Section 4](#) of [\[RFC9814\]](#) describes how, when the content of a signed-data is large, performance may be improved by including signed attributes. This is as true for ML-DSA as it is for SLH-DSA, although ML-DSA signature generation and verification is significantly faster than SLH-DSA.

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

3.3. SignerInfo Content

When using 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. Each ML-DSA parameter set has a collision strength parameter, represented by the "λ" (GREEK SMALL LETTER LAMDA, U+03BB) symbol in [\[FIPS204\]](#). When signers utilise signed attributes, their choice of digest algorithm may impact the overall security level of their signature. Selecting a digest algorithm that offers λ bits of security strength against second preimage attacks and collision attacks is sufficient to meet the security level offered by a given parameter set, so long as the digest algorithm produces at least $2 * \lambda$ bits of output. The overall security strength offered by an ML-DSA signature calculated over signed attributes is the floor of the digest algorithm's strength and is the strength of the ML-DSA parameter set. Verifiers **MAY** reject a signature if the signer's choice of digest algorithm does not meet the security requirements of their choice of ML-DSA parameter set. [Table 1](#) shows appropriate SHA-2 and SHA-3 digest algorithms for each parameter set.

SHA-512 [\[FIPS180\]](#) **MUST** be supported for use with the variants of ML-DSA in this document. SHA-512 is suitable for all ML-DSA parameter sets and provides an interoperable option for legacy CMS implementations that wish to migrate to use post-quantum cryptography, but that may not support use of SHA-3 derivatives at the CMS layer. However, other hash functions **MAY** also be supported; in particular, SHAKE256 **SHOULD** be supported, as this is the digest algorithm used internally in ML-DSA. When SHA-512 is used, the id-sha512 [\[RFC5754\]](#) digest algorithm identifier is used and the parameters field **MUST** be omitted. When SHAKE256 is used, the id-shake256 [\[RFC8702\]](#) digest algorithm identifier is used and the parameters field **MUST** be omitted. SHAKE256 produces 512 bits of output when used as a message digest algorithm in the CMS.

When signing using ML-DSA without including signed attributes, the algorithm specified in the `digestAlgorithm` field has no meaning, as ML-DSA computes signatures over entire messages rather than externally computed digests. As such, the considerations above and in [Table 1](#) do not apply. Nonetheless, in this case implementations **MUST** specify SHA-512 as the `digestAlgorithm` in order to minimise the likelihood of an interoperability failure. When processing a `SignerInfo` signed using ML-DSA, if no signed attributes are present, implementations **MUST** ignore the content of the `digestAlgorithm` field.

Signature Algorithm	Digest Algorithms
ML-DSA-44	SHA-256, SHA-384, SHA-512, SHA3-256, SHA3-384, SHA3-512, SHAKE128, SHAKE256
ML-DSA-65	SHA-384, SHA-512, SHA3-384, SHA3-512, SHAKE256
ML-DSA-87	SHA-512, SHA3-512, SHAKE256

Table 1: Suitable Digest Algorithms for ML-DSA

`signatureAlgorithm`: The `signatureAlgorithm` field **MUST** contain one of the ML-DSA signature algorithm OIDs, and the `parameters` field **MUST** be absent. The algorithm OID **MUST** be one of the following OIDs described in [Section 2](#):

Signature Algorithm	Algorithm Identifier OID
ML-DSA-44	id-ml-dsa-44
ML-DSA-65	id-ml-dsa-65
ML-DSA-87	id-ml-dsa-87

Table 2: Signature Algorithm Identifier OIDs for ML-DSA

`signature`: The `signature` field contains the signature value resulting from the use of the ML-DSA signature algorithm identified by the `signatureAlgorithm` field. The ML-DSA (pure mode) signature-generation operation is specified in Section 5.2 of [\[FIPS204\]](#), and the signature-verification operation is specified in Section 5.3 of [\[FIPS204\]](#). Note that [Section 5.6](#) of [\[RFC5652\]](#) places further requirements on the successful verification of a signature.

4. Security Considerations

The security considerations in [\[RFC5652\]](#) and [\[RFC9881\]](#) apply to this specification.

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

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.

By default, ML-DSA signature generation uses randomness from two sources: fresh random data generated during signature generation, and precomputed random data included in the signer's private key. This is referred to as the "hedged" variant of ML-DSA. Inclusion of both sources of random data can help mitigate against faulty random number generators, side-channel attacks, and fault attacks. [FIPS204] also permits creating deterministic signatures using just the precomputed random data in the signer's private key. The same verification algorithm is used to verify both hedged and deterministic signatures, so this choice does not affect interoperability. The signer **SHOULD NOT** use the deterministic variant of ML-DSA on platforms where side-channel attacks or fault attacks are a concern. Side channel attacks and fault attacks against ML-DSA are an active area of research [WNGD2023] [KPLG2024]. Future protection against these styles of attack may involve interoperable changes to the implementation of ML-DSA's internal functions. Implementers **SHOULD** consider implementing such protection measures if it would be beneficial for their particular use cases.

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

5. Operational Considerations

If ML-DSA signing is implemented in a hardware device such as a hardware security module (HSM) or a portable cryptographic token, implementers might want to avoid sending the full content to the device for performance reasons. By including signed attributes, which necessarily includes the message-digest attribute and the content-type attribute as described in Section 5.3 of [RFC5652], the much smaller set of signed attributes are sent to the device for signing.

Additionally, the pure variant of ML-DSA does support a form of pre-hash via external calculation of the "μ" (GREEK SMALL LETTER MU, U+03BC) "message representative" value described in Section 6.2 of [FIPS204]. This value may "optionally be computed in a different cryptographic module" and supplied to the hardware device, rather than requiring the entire message to be transmitted. Appendix D of [RFC9881] describes use of external μ calculations in further detail.

6. IANA Considerations

For the ASN.1 module in Appendix A, IANA has assigned the following object identifier in the "SMI Security for S/MIME Module Identifier (1.2.840.113549.1.9.16.0)" registry:

Decimal	Description	Reference
83	id-mod-ml-dsa-2024	RFC 9882

Table 3: Object Identifier Assignments

7. References

7.1. Normative References

- [CSOR] NIST, "Computer Security Objects Register (CSOR)", 13 June 2025, <<https://csrc.nist.gov/projects/computer-security-objects-register/algorithm-registration>>.
- [FIPS204] NIST, "Module-Lattice-Based Digital Signature Standard", NIST FIPS 204, DOI 10.6028/NIST.FIPS.204, August 2024, <<https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf>>.
- [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>>.
- [RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <<https://www.rfc-editor.org/info/rfc5652>>.
- [RFC5754] Turner, S., "Using SHA2 Algorithms with Cryptographic Message Syntax", RFC 5754, DOI 10.17487/RFC5754, January 2010, <<https://www.rfc-editor.org/info/rfc5754>>.
- [RFC6211] Schaad, J., "Cryptographic Message Syntax (CMS) Algorithm Identifier Protection Attribute", RFC 6211, DOI 10.17487/RFC6211, April 2011, <<https://www.rfc-editor.org/info/rfc6211>>.
- [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>>.
- [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/info/rfc8702>>.
- [RFC9881] Massimo, J., Kampanakis, P., Turner, S., and B. E. Westerbaan, "Internet X.509 Public Key Infrastructure -- Algorithm Identifiers for the Module-Lattice-Based Digital Signature Algorithm (ML-DSA)", RFC 9881, DOI 10.17487/RFC9881, October 2025, <<https://www.rfc-editor.org/info/rfc9881>>.

7.2. Informative References

- [FIPS180] NIST, "Secure Hash Standard", NIST FIPS 180-4, DOI 10.6028/NIST.FIPS.180-4, August 2015, <<https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>>.

-
- [FIPS205]** NIST, "Stateless Hash-Based Digital Signature Standard", NIST FIPS 205, DOI 10.6028/NIST.FIPS.205, August 2024, <<https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.pdf>>.
- [KPLG2024]** Krahmer, E., Pessl, P., Land, G., and T. Güneysu, "Correction Fault Attacks on Randomized CRYSTALS-Dilithium", Cryptology ePrint Archive, Paper 2024/138, 2024, <<https://ia.cr/2024/138>>.
- [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>>.
- [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/info/rfc5911>>.
- [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/info/rfc8032>>.
- [RFC8419]** Housley, R., "Use of Edwards-Curve Digital Signature Algorithm (EdDSA) Signatures in the Cryptographic Message Syntax (CMS)", RFC 8419, DOI 10.17487/RFC8419, August 2018, <<https://www.rfc-editor.org/info/rfc8419>>.
- [RFC9814]** Housley, R., Fluhner, S., Kampanakis, P., and B. Westerbaan, "Use of the SLH-DSA Signature Algorithm in the Cryptographic Message Syntax (CMS)", RFC 9814, DOI 10.17487/RFC9814, July 2025, <<https://www.rfc-editor.org/info/rfc9814>>.
- [WNGD2023]** Wang, R., Ngo, K., Gärtner, J., and E. Dubrova, "Single-Trace Side-Channel Attacks on CRYSTALS-Dilithium: Myth or Reality?", Cryptology ePrint Archive, Paper 2023/1931, 2023, <<https://ia.cr/2023/1931>>.
- [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>>.

Appendix A. ASN.1 Module

```
<CODE BEGINS>
ML-DSA-Module-2024
{ iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs9(9)
  id-smime(16) id-mod(0) id-mod-ml-dsa-2024(83) }

DEFINITIONS IMPLICIT TAGS ::= BEGIN

EXPORTS ALL;

IMPORTS SIGNATURE-ALGORITHM, SMIME-CAPS
  FROM AlgorithmInformation-2009 -- in [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-ml-dsa-44, sa-ml-dsa-65, sa-ml-dsa-87
  FROM X509-ML-DSA-2024 -- From [RFC9881]
  { iso(1) identified-organization(3) dod(6) internet(1)
    security(5) mechanisms(5) pkix(7) id-mod(0)
    id-mod-x509-ml-dsa-2024(119) } ;

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

SignatureAlgorithmSet SIGNATURE-ALGORITHM ::= {
  sa-ml-dsa-44 |
  sa-ml-dsa-65 |
  sa-ml-dsa-87,
  ... }

SMimeCaps SMIME-CAPS ::= {
  sa-ml-dsa-44.&smimeCaps |
  sa-ml-dsa-65.&smimeCaps |
  sa-ml-dsa-87.&smimeCaps,
  ... }

END
<CODE ENDS>
```

Appendix B. Examples

This appendix contains example signed-data encodings. They can be verified using the example public keys and certificates specified in [Appendix C](#) of [\[RFC9881\]](#).

The following is an example of a signed-data with a single ML-DSA-44 signer, with signed attributes included:

-----BEGIN CMS-----

MIIKsAYJKoZIhvcNAQcCoIIKoTCCCP0CAQExDTALBg1ghkgBZQMEAgMwQwYJKoZI
hvcNAQcBoDYENE1MLURTQS00NCBzaWduZWQ0tZGF0YSBleGFtcGx1IHdpdGggc2ln
bmVkiGF0dHJpYnV0ZXMXggpCMIIKPGIBATA6MCIXDTALBgNVBAoTBELFVEYxETAP
BgNVBAMTCExBTVBTIFdHahQVn/5vIv1cxCxSTfb9XiJQ3jjzTjALBg1ghkgBZQME
Ag0gazAYBgkqhkiG9w0BCQMxCwYJKoZIhvcNAQcBME8GCSqGSIb3DQEJBDFCBEAL
v5NoEkfe30kMRW4rKXw97hdFLivtQ/OVU4Pc/DrfWm3d7P0pIXNQ4WCwyGDTWKwi
dWwCHZ9E3CT0Twj2gI/UMAsGCWCsAF1AwQDEQSCCXTzX9ZSUYiiAjJ2USF/0b1K
fyTnaJTCFymSXY/ZOE0++0F6BZ9HUQweqTlrfXUmp0L1YK+8Hd/zCmyjboKZZmCA
KY4rPlbi4W9ndcowgSgawGixVs0vOBimudg4B5Tbo43c0RwIPW6FdDrCa9eKgcGh
bMIFTYFF7f9J3suzYmcj7H99nDjd3d9P0qPW0J2NWz64UoxZP8iH0u78gd46yIwB
Rz9VYerD0BS0kZiU2kQUXGhCKm0og0ES8Vg1TfV3esn7xeLb0hn4uyrpS0Bx5bdc
3BLRxvWdic+ha0SFQns5uSrduRjXTaLi88tnVWknzfidCzKubzIxJ/7CMcEcXxu+
L+dUOVXZvATV3Fiddk9re8x54Z7gb0kHEyemJnf9uq+084pGB/LrIH5x+ZyYdzlZ
Ys1a7XqEONK/VIuwD2E7UHcYDSROZAYRMFGoyqGKdwVD6/W1E1DYND6eX7Vqss4H
jDuDi7qsha2j4oHet5JQWYecSxSUsmwp+5E9S6p3g/30w4iA1EGQLGZV1H76m+4+
JYWnHapiFFPQ4nxly+C6c6+hDaX+KONZdM/lt0eaJnxq9Nzrprw/ieIqX8A70v9t
1MLVwd7W8Gc4auZec/8WrnDI/f7qaSU0Kt+kNN0ok2maZvLYbDyaDS1UyK4IXvqA
FR5fbSgFmy7SY2TDc4k8JJ/KdBqSg8k0/tRemBiXE/YfltddyZqsD+vhoz5RXhl0
DvyZbQwxW67bdgr6TgRKexRuWQOTR9CAWNitmPzmZDRqIXIhtbg3jtoXujTg4003
/tjhr+ZxCv5zsgcbUiJBicSHRuc1W1er0CRu+fknwXZBgF73WtFhDfDq8u9a00e
jBTW4xMAXVfv3coIaknsDP+Di9LtvSxXhLsMaRr9bFZnfhcfU4/00w+rGwbZ8114
y8Ech//OPjYQxmFvXaqV9r2Fz6KkslwlerMq/MjFUjt6vNcxHaGEID/m+xzSJAB
5/BzW0qkIBFoWIDHTkYo9wie7QI6cbgM7qbpTxJAbauPU0VYf2VUTTuGxVtb4aNQ
zMDYSBjHVDjZ3/o+kmkjrlBxl+Jvx7Qe10G0VNHKMP70wMIXj50txvWqRV1TXIvm
p5Qv/NFJWQTJWDv608Mt5/4lbGqJB07v9T7gfvxd1LWXmmd1X/T8oPg9rFI6rGNP
Nz7xoxs8xkAa+sBcoPmNQyk9q9srER8Fwi3eBGNuFuAq8nKfn+2LXh/Iuhxk6BFc
a1wC4Qa5PV4uiKjsUrKyWwux12Z3dAbtLI f9HNStu1157KaiJ/XLKcsUsDVAcq8L
GJHpuT000Y/2Ai/JkE6CjJH9nEXQLGxWHadD0gJrQA8rnwV0ccex7RjX7xkhh/0d
b3HxL2f0Ft6lyWgFK1uZKpLrp1fk6+U1hXk+EuUfdayrT0t5poNo1RXaohINP7m
ZZj1yqGhW1bq0xkZt7xantZ5FB1QuT9hT5FiY4TFoB1Z5LJlXvLpM/QFB/4n9ZJi
fqjKA6wMCWxBpsu4+Z0faQkwvRZ+9+08QIM1QaRqyMoZeSVh622QmUjuAw7EyYY
KRR/sPkLe1SFxWfg6mcqrnABRGy2kHs2a63j4MIpev1DonKNWPbbBSzkqncPYpb6
MHXQTiL1/uqbl/vUE1NucQxvzsaCIDP0ULQiZLS5PU018rjWa3BbE0ner4MyAT2s
QXj5fxHYmuT69JppafV9omZa30d2mUDDtz9Wy2xGRE8MvSrawsRNE5Hucc/tXZu1
Bz0GPARTzKB3lgrXuQU9CyYSM3T387tm1o1AXm0J0/H4bhAbAqFeFnL1Wm/gFWFr
ocpVPNWAWRQj7NdteRMX/qE8nWMjG11ax7w13BPa8pDwC+61pnVfGDzBNlWbZThz
oXtjGTTRuFi1Zpy6BgVAPuVZcxXC6Pg8Eeod01XH4pPKtPJ+tkCWLrnXzMur7oAP
i5P3UZ/AEXrLiMw/f6o1tVVDWvGD9T50eemgB4fRzSG/0Sxu1WpMBm1va1v56Gym
U0u59MHb6jR2NpsGBRu1J/5FVoxghvitSA4ggAhkLmIndoNcW0ThHJx67WBJH78h
gVHhjQBuAXwR1focyqdrNw4B9iVAEx/sxldvF9pIvlsnRXKore8RF9p40fYz7GGc
2+cbtdgCVyfpnt2u2reyvPg0AzW/Moms+AXs+LaxzHt6mrWIJOsuNtLwrwTEJu1t
GkQibWzWd1G+wb885YvMxAoAXU9s88jSWzEyfUS4ksMgG2CVrmfewHeFuLIFR9D1
LZkFSmQTgWLKwdJw73XUgF0qHxzMTBkLoTAIQasTZKjC160zCbWzV5e/PT7hqvQk
ic07PJLIjA41uhGnSyaN2ELYQYKQFcTakY5eHYaDHDJgMZTTKMn+k1SHYHCBYkzH
ToSoodOW7ezgzjzKJMAp3A/egYFrChpOdmiCkE6ot20CW8Ju9vxKQMWAXXe1FoA7
j3tVsqIUdvTjzyAGINsVU8ihKaSSt08khn0ftb/aUj7eN36FHMwMeNH2LhXbwSJI
++u4GWW3woD8ZUyo1mpH7xLmBrci7Phs7gFpHtJeIZpPBeG5MuEDpvzCHHBBrvUA
Ek8zuLLGYdlbb2PWGM6A3M+efSnjaY6JQS3GURQLA9BWMtuS5L3+ytm0F00w0VCA
hq2BN+vNwXm1XWqlEG1sbpAUbnGwKpyipUT3GBBvjp+Ak3RIlciLQGCZ1I1Xeg1E
W9K8YhhLo490h3GDuf4CZgPULSHXqKcCr91VDpff/kcxwVeXITQiFVykwjfe11XT
gnxR3zQRP61P3aisQxwsaKgHKGzD5idGAzGQuwVgAs95xA/ka1ccMe8a5da+bKP/
9QqnAFFtArVZpso0Xcy2D/iusW2bcBjiSANM4GnZwsyphF0WIK89aq/411WIz3zc
Xf1JIW80fAy47VF8W340bSgc24A0rQ1z38TEGLIcvqPvSMTQRVud12S9PgGo8cpP
J5+lm7FzJftRSTwYsaSwtOUM1hvvXbvcWf03g8XMJbof8cWH7QeEPcan+ygxbtt
ArQ5Dk+BE4Rv/MBJUvi5E30IBHxWXx60TWsljFDjBwt8bPVk7YMaBWMMy4KZw5jU
nRakavONHDQDizfy7U0IRAEjKTxKTFaRk56+y839PF2T1p63w00UFzAyQVvKZ2uR

```

zs/Q7xYbHEBpepGfq7C0w9Tp7fgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
DhYkNA==
-----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 { "ML-DSA-44 signed-data example with sig
ned attributes" }
        }
      }
      SET {
        SEQUENCE {
          INTEGER { 1 }
          SEQUENCE {
            SEQUENCE {
              SET {
                SEQUENCE {
                  # organizationName
                  OBJECT_IDENTIFIER { 2.5.4.10 }
                  PrintableString { "IETF" }
                }
              }
              SET {
                SEQUENCE {
                  # commonName
                  OBJECT_IDENTIFIER { 2.5.4.3 }
                  PrintableString { "LAMPS WG" }
                }
              }
            }
          }
          INTEGER { `159ffe6f22fd5cc42c524df6fd5e28d0de38f34e` }
        }
      }
      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 {
  # messageDigest
  OBJECT_IDENTIFIER { 1.2.840.113549.1.9.4 }
  SET {
    OCTET_STRING { `0bbf93681247c4dce90c456e2b297c3d
ee17452e2bed43f3955383dcfc3adf5a6dddecf3a9231350e160b0c860d358ac
22756c1c1d9f44dc24f44f08f6808fd4` }
  }
}
SEQUENCE {
  OBJECT_IDENTIFIER { 2.16.840.1.101.3.4.3.17 }
}
OCTET_STRING { `f35fd6525188a202327651217fd1bd4a7f24e7
6894c21729925d8fd9384d3efb417a059f47510c1ea9396b7d7526a4e2e560af
bc1ddff30a6ca36e8299666080298e2b3e56c8e16f6775ca3081281ac068b156
c3af3818a6b9d8380794dba38ddc391c083d6e85743ac26bd78a81c1a16cc205
4d8145edff49decbb3626723ec7f7d9c325ddddf4f3aa3d6d09d8d5b3eb8528c
593fc8873aeefc81de3ac88c01473f5561eac338148e919894da44145c68422a
63a880e112f158354df5777ac9fbc5e2db3a19f8bb2ae948e071e5b742dc12d1
c6f59d89cfa168e485427b39b92adbb918d74da2e2f3cb67556927cdf89d0b32
ae6f323127fec231c11c5f1bbe2fe7543955d9bc04d5dc521d764f6b7bcc79e1
9ee06f49071327a62677fdbaa4f38a4607f2eb207e71f99c9877395962cd5a
ed7a8438d2bf548bb00f613b5077180d244e6406113051a8caa18a770543ebf5
b51250d8343e9e5fb56ab2ce078c3b838bbaac85ada3e281deb792505987824b
1494b26c29fb913d4baa7783fdf4c388809441902c6655d47efa9bee3e2585a7
1daa621453d0e27c65cbe0ba73afa10da5fe28e37374cfe5b7479a267c6af4dc
eba6bc3f89e22a5fc03b3aff6dd4c2d5c1ded6f067386ae65e73ff16ae70c8fd
feea6925342adfa434dd282b699a66f2d86c3c9a0d2954c8ae085efa80151e5f
6d28059b2ed26364c373893c249fca741a9283c934fed45e98189713f61f96d7
5dc99aac0febe1a33e515e19740efc996d0c315baedb760afa4e044a7b146e58
e41347d08058d8ad98fce664346a231221b5b8378eda17b894e0e0e3b7fed8e1
afe6710afe73b2071b522241882b07461b9cd56d5eace091bbe7e49f05d90601
7bdd6b458437c3abcbdb6b4d1e8c14d6e313005d57efddca086a49ec0cff838b
d2edbec5f184bb0c691afd6c56677e171f538fced30fab1966d9f25d78cbc102
87ffce3e3610c6616f5daa95f6bd85cfa2a4b25cf095eaccabf3231548edeaf3
5cc476861080ff9bec73489001e7f0735b4aa42011685880c74e4628f7089eed
023a71b80ceea6e94f12406dab8f5345587f65544d3b86c55b5be1a350ccc0d8
4818c75438d9dffa3e926923ae507197e26fc7b41e94e18e54d84a30fecec0c2
178f9d2dc6f5aa4559535c8be6a7942ffcd1495904c9583bfad3c32de7fe256c
6a8904eeeff53ee07f1bddd4b5979a6775ff4fca0f83dac523aac634f373ef1
a31b3cc6401afac05ca0f98d43293dabdb2b111f05c22dde0469d416e02af272
9f9fed8b5e1fc8ba1c64e8115c6b5c02e106b93d5e2e88a8ec52b2b25b0bb1d7
66777406ed2c87fd1cd4adbb5979eca6a227f5cb902b14b0354072af0b1891e9
b93d0e398ff6022fc9904e828c91fd9c45d02e0c561da743d2026b400f2b9f05
4e71c7b1ed18d7ef192187fd1d6f71f12dfd9f385b7a9725a014ad6e64aa4bae
9d5f93af94d61c64f84b947dd6b2ad33ade69a0da254576a884834fee66598f5
caa1a15a56ead31919b7bc5a9ed679141d50b93f614f91626384c5a01d59e4b2
655ef2e933f40507fe27f592627eaaa3280eb03025b1069b2ee3e64e7da424c2
f459fbd8ef1020c950691ab232865e49587adb6426523b80c3b13261829147f
b0f90b7b54855f0160ea672aae7001446cb6907b366bade3e0c2297afd43a272
8d58f6db052ce4aa770f6296fa3075d04e22f5feea9b97fbd412536e710c6fce
c6822033f450b42264b4b93d43b5f2b8d66b705b10e9deaf8332013dac4178f9
7f11d89ae4faf49a6969f57da2665adf47769940c3b73f56cb6c46444f0cbd2a
```

The following is an example of a signed-data with a single ML-DSA-65 signer, with signed attributes included:

Page 14

Q5Ngzx6rbNyCUbuUxx+ut+IgCAqfbdynWxROD01vW3nbZ72ZZcnejvvvMSWyLQIE
/3aszLIkJ8GDsRt2UxyDc/o0DP04ULboC8B4AQq2qH1+MWILU+QTUm/+Jwg7tVjJ
5r+7kcpQT0J/kGexd86GwsuWQcNjNRZvstyMozrbz5jLahT+XLPBJH4lZWIKTi4
41RC5JRQajZ/Eh9+UYxtsp1wWnNZwXhp4BvMouKB/GtT7CfYB12b4yGGeyxjA7kR
Jip6PiPJUP03MX580kqFkoiDJs1/HpINHdLEIGip83xbEley/KaV2j0u0njyUMdI
FMFebivD0hSEVW6biU7FKFCgNeFxsG3Ls6qabp/kqakZnolfpVU8jTeFpapilZoL
0a/wp/xUiuUTJfARjjq0Z5A+HxVhkhLwykt14KC3v/jcp8URzDxw7/h8LNzEeo1P
C6eT3psEzPN0L3TqJRNCGsDYtrtl0NoT0Zpj7Vj//8cAg4rj1aZiYkIuytJwLvxx
dkLaq2MbJoiCq/OwnRFeARSdwt2viAf+MyI/GU3n1A4mEwM4NsYVJxRZzbUisekJ
L+6cb4T5pnw1wZHySECw3YiHLYHRYHpi9Moi6ldy7HZBNT3z7G00+Z0yA0HSKek1
HD7K6K7L0GL6s9gy/hd779s4DxhLFg2is5xfJ6wcvYDg+wgy8vCoQc/D9SchL98M
DjQlh+x0Z8iqoTJ+z0mYB4fCKxqtiq3ufkrRGKHvkWDEyeTXAWV1/k3sZtEGkmX6
nan2U/Gfqv7ilYel083kb1CRLXeUbEXhBoqBuIAIaTbDwbTRJk38mNAF/14QwPle
IaQ0hwDZ/EAb7IICi64+RKdDGQvYid4jIjy3wuhdz6iCM5vwmVT/K81o67Q0MZj
aCT22unxJK0Se9nwb8T0uEzqRpHtTQftBK+0/nYPZMx3AGjuU6wabb7eR1ux9DVk
QFz0ykykN7gle89bcEjNr6wZ6GtY9qkmkY861+PWVTj4380aSZxNgJibnKhQ3jH5
tr93/r+Jcs0I8a2Vj94y/ufTDAE3uEX9Z3MarceQ9FDcGq5CWQYXR5Cf3oWh0Rii
PC0/qZ6LGmiXV0d8bYYQ1XFxgUpdsLlnb7IyVEt7QJ2CrQfyT1e12bz1c0iCeImt
bQbhWaf550uvkyRpDS/eqHFV/yFMqMurdCvxuKmfEWNgZayG+LhwgPHK5xDfAHwi
ItT2e+G0mVUNecsMutvc5DrP9MTQkU8RUhPx0kiuQi3/Nc5vWIULR1a/MeV1lwuB
l4ZCKyowZ2KW51M3StHgAngy0gbFfil2X9y0P+fGwGvNZTILiQLCnWgZ39Bpm05u
fcQH19aN/Arjnxpgaysx8TilzpfK06Id40aTH5Pt18vMvhnVa/WzXGIy8YkuzAb
lt2IXcZhD3g41s1Cjmr0r20bUfxH/AvFQp60FssB+A411tSp/whzqdanvofjFdz7
yhS1ZTXBHgwJAv0eLEzZ+0B6Q8jiVbzHfOX5g50QRPuGj7pQLiSxPV3GeYHsNqn3
wdiW6gNnEEM8ST9VGIihSVZQ1H86d1S//wNMNLs1957JdQECUgdqpDT+8fya3P4G
/nVz7FU+Go5Zc7IK5FrNhK57JiTuU5INHN8Z1bm+w0oglCk0aZFU0Sf9Qxrhaus+
nYQofSG0zEoB0LyEzjVccbgA5bw75ZsaaMjRIGRotWTXtrMfBoMLNxBmVGAkqluL
7Wm3U1bKG43gcg7sIS2zdh069HD6aUqt+VKDtd2WG7FGMGc6MADwIbVN14E5AcBJ
19kKKQK08f+vrsxpSNY8XRKk5ShnT0ig0vRIoWIAgkN4YJu46YjZ2WorSfuaKNx/
+olnWjhlCRsf3o0l0TpwYlhp7Clok9/t7kCZS8L8Kv0UZ8K36VL0E+4LeKycAZk3
Y4ziBJMW8wDG3tU10QQZfZSKyBEgyCiugr8tXsJAKPLy8U38YtxDtwAgwcXTkDiN
85YXK5AreJR8sr33LZZI3Y0qiCIJVMQWfcSnrcwdSUXDuqXyG979qJr7aRiwt5iH
X2GJqubN0XdpC6Y4KSSTZx4sYs2Ts9/HWFbixXgAgshyz2zLC/0FTR1fiBZF2Zf
7tgoJcF0FqKxJUq4BW0JNk4C/RwpSV5cMiU/rpkwojMJ7HnxV6k+18ZqIUQJ7hWU
cGQm1BP3kd4dueatyC2rvw3UrLfcttiLbAqYTHVo7UHYhpKX1vLZ5p1tPKKz5mb1
zxhnenB3BRKj31+Fq0UE3luHur63W1cLSnqvGFhUcyz47pjZ7VntZrjMu3QyQbeg
bNv/PROC0wp3EYo+C5/AS2H03quY6oW+0Ix1iWw16EzUDCVdnXT3bmnmqNJEN1Hgs
eyiKcmbTX+1378KIYjVY5DE6eYDTyzpc0lxcg8Vb4eM7q2cdmts+jZLTH6Xq/xLQ
Kq93FkNvx8bkC83F8zXor8MbEptzjQcjZI+adJrTTdUDrIDAF3s0ddlgK5Lr15cR
np5p1napwi/VXweRqXTkYqjmZsfCKAe5AaleTfSBnPSCsczIXAVTTQC1CoQfxoM
8jjfzhPzHr/kHaktGQ0mS66L8/Gw/eVDxfgRj876exDl+J5Hp1+2+pHafw8jH00/
EkPn9R/78P70H2P8XVrystIEGM0Bq32jJNgDCT6YARq1HkrUBi1LKGHYniLWFsXw
2mp5Lx/61WSJ3jH0NQ1enyWVwb0iZo2jQxVjccaC+2hKgQgJZNVUr4zBPxcequ5V
rEl29BcXNgEWL5lywVIxYijFULcxyw9g/Z1LTJbBofZ38zqhCxFtKjfraCp+pZaM
jP1+PgZ4CD/Q2uqt2d+0cThjvrru9C1PFk6ssAuGN6DXQnnL3MoFKwL4eCw0UdVR
a9C8ZW7D+ax16gQBmD3hQB/K/4bdQFD3tQRsLoG1DR4Mi10GIvMxj5wdbglrNAeS
1rKMN5M3bJ/Zv4mXE+nfWehBFw4A+gDP3LR21579/WJy3TWG0FIK7Gc23BxhAuJY
hWE80C/NmuHhZp7n2u0mydFpkiGA4HcQaJti3Cw9bwMCoJmKQdvUZG+bJYNBLW/3
v/lo4Ireg30JE18wi0TXsqvtqoAfVoERh4ZQMYMz4PDooxG0KqDgHyDfY3AEU506
KAVCjquMuCazq/B8CTMSqg2HrufMBVg0S4mzfwiCK6CdZsHbzMWy7yy28Bn5/Vfa
r/tBXMESqvfz2RZmYk2mgoaxHxYwbDT/tH01EBkSuXG243J5VUbd0DGcny16s43a
GQ2mLRz7KqCAK/QXgy7yU/quguVy6bUsSZxxwnpCv09fCg8VZkThuME19DKe68bt
b1xrzc4jXKLpa5C6LGIy4+BYVRV9NsZL0Z6RDcIIKYA7wnjutMNdYRBg86ukvdC
q4CKWpGVH9851yS+PP0Yhvo0cfMpKVg1EoPuCX4qFEX9Qt8Rs1vxEpUE3djYykuE
WKvzh+yS1h0TnNnHIGNVGS0ZVVt4rV+Rn2Sh3DZbR6U5tFcCK6FziH/wwQ7FL4YU
v4uCF1xLZtMkulYE9a7SRvUYqeX88CEQQ57zQasJa+a/puljswL7UV/QBnmnM44g
NmRyyHSD0bZp1X2hKr6cbQ6IDACM0YLBqveN0x478tW65D/e3EdQip4LKPf3TB/2
NabF50gr/XPeh9eMKJzCEFA2NBzy20yjr6uHGprkd4Yd7iMzBz/DD9P/4dE61AXGA


```

vALm0S8mrv8p6S1ln2lrYjYptdELG6FbAm5ZFRWD9XDQUCmbDp8qQkw4q7nFSLTx
lzu6lQIiB7weAoJ0/WyhrD75GTcp7W9e0pcmqQL6YMYTIlvRSoq0aK4l4nz+7eUY
tCuJjGDmj/+2kHV0ZUF/p8fzZmsWBcgpMUJnPo0hTUZ3oQqxsNYFiXZDStVtyA7b
hS80X6kE08652tGQop6jIx3WEUs/vqSa/h1BHVW3a0d29Rqw0Tf1o6BoIoDdccpi
4NlIgwVFxFhzqxy9QvQF0nuaPIaCZFF8vTxaMSVD7JVmvAG2QJXQXfseyttHnaut
i3iV/dQfCk6q5AF3FfLWmpbv7xGzgAqEQLJbWGTgzkWhrUd4XSxMuz3Fdr2miYqZ
bKeW7WTYzheWIByiulhuxh9UYf0GDxAYY4m5EGV5pek6xgwhMj1YYmVobHng4g8n
YK0x3QAAAAAAAAAAAAAAAAAAAAAAAAAAECxASHiQ=
-----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 { "ML-DSA-65 signed-data example with sig
ned attributes" }
        }
      }
      SET {
        SEQUENCE {
          INTEGER { 1 }
          SEQUENCE {
            SEQUENCE {
              SET {
                SEQUENCE {
                  # organizationName
                  OBJECT_IDENTIFIER { 2.5.4.10 }
                  PrintableString { "IETF" }
                }
              }
              SET {
                SEQUENCE {
                  # commonName
                  OBJECT_IDENTIFIER { 2.5.4.3 }
                  PrintableString { "LAMPS WG" }
                }
              }
            }
          }
          INTEGER { `159ffe6f22fd5cc42c524df6fd5e28d0de38f34e` }
        }
      }
      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 {
        # messageDigest
        OBJECT_IDENTIFIER { 1.2.840.113549.1.9.4 }
        SET {
          OCTET_STRING { `d5740888352a0e92a69df3eb1a1ce555
60ac3f2d2f8281ce3f06a56d3a8285cb24ee6404757129a17aef477cdf1a443a
12220e30cfde2308f7b88142ce9e3aa8` }
        }
      }
    }
    SEQUENCE {
      OBJECT_IDENTIFIER { 2.16.840.1.101.3.4.3.18 }
    }
    OCTET_STRING { `529c9039cce0a4fc9d267e4967892860063cc4
db31f7a1a7c91ed2d3008b49c0eee7880dace2daa653d2df555389fe6d700b9
0b357398634e7badd2b2fd5e16bbeb5ba865963542b0052580670ff886d257ac
df6d6fb4fe7d24e7f40f577cca6cd8631a8b5e67f6fa7872389ae3d61216848d
ba6c06acad16550dbfa5d16ac942ddcaf93de955199e38682a6460e1bc4d2cfb
7d6f5e2025903be45a7322d205439360cf1eab6cdc8251bb94c71faeb7e22008
0a9f6ddca75b144e0f4d6f5b79db67bd9965c9de8efbef3125b22d0204ff76ac
ccb22427c183b11b76531c8373fa340cfd3850b6e80bc078010ab6a87d7e3162
0b53e413526ffe27083bb558c9e6bfbb91ca504f427f9067b177ce86c2cb9641
c36335166fb324f2328ceb6f3e632da853f972e90491f89735882938b8e35442
e494506a367f121f7e518c6db29d705a7359c17869e01bcca2e281fc6b53ec27
d8075d9be321867b2c6303b911262a7a3e23c950fd37317e7cd24a8592888326
c97f1e920d1dd2c42068a9f37c5b1257b2fca695da3d2ed278f250c74814c15e
6e2bc33a1484556e9b894ec528572035e1714a0dcb3aa9a6e9fe4a9a199e89
5fa5553c8d3785a5aa62959a0bd1aff0a7fc548ae51325f0118e3a8e67903e1f
15619212f0ca4b75e0a0b7bffdca7c511cc3c70eff87c2cdcc47a8d4f0ba793
de9b04ccf3742f74ea2513421ac0d8b6bb65d0da13399a63ed58fffc700838a
e3d5a648ca422ecad2702efc717642daab631b268882abf3b09d115e01149dc2
ddaf8807fe33223f194de7d40e2613033836c615271459cdb522b1e9092fee9c
6f84f9a67c35c191f24840b0dd88872d81d1607a62f4ca22ea5772ec7641353d
f3ec638ef993b200e1d229e9351c3ecae8aecbd062fab3d832fe177befdb380f
184b160da2b39c5f27ac1cbd80e0fb0832f2f0a841cfc3f527212fdf0c0e3425
87ec7467c8aaa1327ecf49980787c22b1aad8aadee7e4ad118a1ef9160c4c9e4
d7016575fe4dec66d1069265fa9da9f653f19fa95ee29587a53bcde46f50912d
77946c45e1068a81b880086936c3c1b4d1264dfc98d005fe5e10c0f95e21a434
8700d9fc401bec82028bae3e44a743190bd889de23209cb7c2e85dcfa882339b
f03154ff2bcd68ebb40638c6636824f6dae9f12643927bd9f06fc4ceb84cea46
91ed4d07ed04afb4fe760f64cc770068ee53ac1a6dbede475bb1f43564405cf4
ca4ca437b8257bcf5b7048cdafac19e86b58f6a926918f3ad7e3d65538f8dfcd
1a499c4d80989b9ca850de31f9b51f77feb78972c388f1ad958fde32fee7d30c
0137b845fd677300adc790f450dc1aae4259061747909fde85a13918a23c23bf
a99e8b1a689757477c6d8610d57171814a5db252e76fb232544b7b409d82ad07
f24f57b5d9bcf57348827889ad6d06e159a179e74baf9324690d2fdea87155ff
214ca8cbab742bf1b8a99f11636065ac86f8b87080f1cae710df007c2222d4f6
7be18e99550d79cb0cbadbdce43acff4c4d0914f115213f13a48ae422dff35ce

```

6f58850b4756bf31e575970b81978642932a16cf6296e753374ad1e0027832d2
06c57e29765fdbc43fe7c6c06bcd65320b22a2c29d6819dfd0699b4e6e7dc407
d7d68dfc0ae39f1a606b2b31f13225ce92052b4e88778d1a4c7e4fb65f2f32f8
6755afd6cd7188cbc624bb301b96dd885dc6610f7838d6cd428e6ae8af6d1b51
fc47fc0bc5429eb416cb01f80e35d6d4a9ff0873a9d6a7be87e315dcfbca14b5
6535c11e0c0902f39e2c4cd9fb407a43c8e255bcc71685f983939044fb868fba
502e24b13d5dc67981ec36a9f7c1d896ea036710433c493f551888a1495650d4
7f3a7754bfff034c34bb35f79ec975010252076aa434fef1fc9adcfe06fe7573
ec553e1a8e5973b20ae45acd84ae7b2624d4bb920d1cdf1995b9bec0ea209429
34699154d127fd431ae16aeb3e9d84287d21b4cc4a0138bc84ce355c71b800e5
bc3be59b1a68c8d1206468b564d7b6b31f06830b37106654600aaa5b8bed69b7
5256ca1b8de0720eec212db3761d3af470fa694aadf952834ddd961bb1463200
ba3000f021b54dd7813901c6c9d7d90a2902b4f1ffafacecc6948d63c5d12a4e5
28674f48a0d2f448a162001a4378609bb8e988d9d96a2b49fb9a28dc7ffa8967
5a386571149fde83a5d13a7060b869ec296893dfedee40994bc2fc2af39467c2
b7e952f413ee0b78ac9c019937638ce2049316f300c6ded525d104197d948ac8
1120c828ae82bf2d5ec24090f2f2f14dfc62dc43b70020c1c5d390388df39617
2b902b78947cb2bdf72d9648dd8d2a88220954c4167dc4a7ac2c1d4945c3baa5
f21bdefda89afb6918b0b798875f6189aae6cdd177690ba638292493671e2c62
cd93b1ff7f1d615b8b35e0020b07cb3db32c2ff41534757e205917665feed828
25c17416a2b1254ab8056389364e02fd1c29495e5c32253fae9930a23309ec79
f157a93e97c66a214409ee15947064269413f791de1db9e6adc82dabbf0dd4ac
b7dcb6d88b6c0a984c7568ed41d8869297d6f2d9e69d6d3ca2b3e666e5cf1867
7a70770512a3df5f85ab4504de5b87babeb75a570b4a7aaf185854732cf8ee98
d9ed59ed66b8ccbb743241b7a06cbbff3d1382d30a77118a3e0b9fc04b61f4de
ab98ea85bed08c75896c35e84cd40c255d9d74f76e69ea34910dd4782c7b288a
0a66d35fe977efc288623558e4313a7980d3cb3a5cd2573183c55be1e33bab67
1d9adb3e8d92d31fa5eaff12d02aaf7716436fc7c6e40bcd5f335e8afc31b10
fb738d0723648f9a749ad34dd503ac80c0177b0e75d9602b92ebd797119e9e69
9676a9c22df55f0791a915d3918aa3999b1f08a01ee406a57937d20673d20ac7
332170154d3402d42a107f1a0cf238dfce13f31ebfe41da92d190d264bae8bf3
f1b0fde543c458118fcefafa7b10e5f89e47a75fb6fa91da7f0f231ced3f1243e7
f51ffbf0fef41f63fc5d5af2b1d21e18cd01ab7da324d803093e98011aa51e4a
d406288b2861f23622d616c5f0da6a792f1ffa956489de31f4350d5e9f2595c1
b3a2668da343156371c682fb684a81080964d554af8cc13f171eaaee55ac4976
f417173601162f9972c152316228c550b731cb0f60fd9d4b4c96c1a1f677f33a
a10b116d2a37eb682a7ea5968c8cfd7e3e0cf8083fd0daeaadd9dfb4713863be
baeef4294f164eacbb0b8637a0d74279cbdccca052b02f8782c0e51d5516bd0bc
656ec3f9ac75ea0401983de1401fcafff86dd4050f7b5046c2e81b50d1e0c8a53
8622f3318f9c1d6e096b340792d6b28c3793376c9fd9bf899713e9df59e84117
0e00fa00cfdcb476d79efdfd6272dd3586d0520aec6736dc1c6102e8d885613c
d02fcd32e1e1ce9ee7dae3a6c9d169922180e07710689b62dc2c3d6f0302a093
2441dbd4646f9b2583412d6ff7bfff968e08ade837d09135f308b44d7b2abedaa
801f568111878650318333e0f0e8a311b42aa0e01f20df637004539d3a280542
8ea50cb826b3abf07c093312aa0d87aee7cc0558344b89b37f08822ba09d66c1
dbccc5b2ef2cb6f019f9fd57daaffb415cc12caaf7f3d91666624da68286b11f
16306c34ffb473b5101912b971b6e372795546ddd0319c9f297ab38dda190da6
2d1cfb2aa0802bf417832ef253faae82e572e9b52c499c71c27a42bcef5f0a0f
156644e1b8c125f4329eebc6ed6f5c6bcdce235ca2e96b90ba2c6232e3e05855
157d36ccd92ce67a44370820a600ef09e3bad30d75844183ceae92f742ab808a
5a91951fdf399724be3cf39886fa3471f3292958351283ee097e2a1445fd42df
11b25bf1129504ddd8d8ca4b8458abf31fec92d613939cd361206355192a1955
5b78ad5f919f64a1dc365b47a539b457022ba173887ff0c10ec52f8614bf8b82
175c4b66d324ba5604f5aed246f518a9e5fcf02110439ef341ab096be6bfa6e9
63b302fb515fd00679a7338e20366472c8748339b669957da12abe9c6d0e880c
008cd182dbaaf78dd31e3bf2d5bae43fdedc47508a9e0b28f7f74c1ff635a6c5
e7482bfd73de87d78c289cc2105036341cb6d328ebeae1c6a6b91de1877b88cc
c1cff0c3f4ffff8744ea5017180bc02e6d12f26aeff29e92d659f696b623629b5

C4UhcBVX2PVtT2knDqnWe73vimTlTiMM79Yno6EK2QQ7wCU/dt2Qzfwb4GbpP2qB
Mh8fnfJfK7fY0VUVn2bJttzyQYqh83DpgJJ6W1AFNZj sm/JJ8Pq74qy+6uIXKVGa
7mtv0vvwZuVP6nVVBmJGY4Brx1ZIg7I2I0yaTK+Lm0F1JGTyoktZgS08/AWwFlvf
qSLcX2WV0s0wic9MLOj3yZNeVQhEmKaQ1TQ0gtaw6NYoa0f+mGT9w/OtC0ltTWfy
ohM4Lb0GEyupuosv0K4ZiEU740Ir4y39zUugVHY09oHTzG5iSYbvRviewctNWKq3
LYXwtq0byov7SfV/YbQSZxo9azdQtasSqdcN7LdoheoK/Tfs4pYAt0s3yE5Dd/0
lZBdk+M/mpkQnwre15FE1ahDGrQoyTw0iyJ6JWXsILMyEB1NvBYU7iawHe1+R7hn
MKamavo1V9EYtTzFmXn5fupDIitjwHIYWo+J3NZoP8uPu50S/IdJCavge+KYi8pjQ
3F/QGbr5+kMCMns71UdqTRY6oYWtzXIZRtYWBjFphowPUS+OV69SEMDYdJBF+83Q
Vyojyj1l3gP4l0pJwF1gIajPxbqphaqTTqAhDYZxIvxESpd2ZARD+afL6wLPRfRI
sHJl/1z00/xHF+40og0MFGao9zZl/yf8h6Tt8rDzQvzva9ftHWr0wLvengvKia2i
+TvSrHrQwxwv3C/tSH205qadjJifrBQQGvL4lGI1TK54/9qJZYVDRoKCF7HybtAY
NW7jgdrEXim3B4Q2zZbCzAj536080Gpw6p18wg84zqMpsPMse0WEBLOSDEamu+u0
9WSBct42059gwLR8togJjRrme1dlc4DbgtvqFpt3jvUSrxhFoAmF+bF0gUNXKYdD
17YuuDSQX0vBsZwwA/HRsldeU2Ui9EaaYAsB1RvQxajfHZ+89h1/ciHgOfqDNGUo
Ys1Dm5IDI7KzG+CVDHsVcaHq4Z3xZ5qWwYdVG3go0Jw6b20Q/KQjFR9ewjzuEkOn
GD14vYRRoraGc5m/PPz0etJHbzXqgoc4zt1kfZ1c/ecjgyfzD+7a9f/X2HCc05hU
ZO/P49aysUZWSxNqY3r02J80F+9am6ooySLBTmC0z2W75o0h09eSzwk+MUTQW2f
VfgaisIoQzpchXma675Vnu3ikH3VU1qse2CDMXZtmLcJMxTofWogekIvF07bxeEt
3eBHAUg1Ll63PgByQ1TXMCFyWLu2tP9MngNGeM/mckXFg7LQsyQBL06/09oga+C
1UAA12onrz4VpwbAAWmjYHgaizJ/4P3bFREmQ+66Inb5x5m9mZoUG5t5XjKze0W
JbaANsnwz72+qPd9LFkj2W/qaRiLR6N6aYDF5vtk1PXRjfh7GzwGQ/tPy88SROGN
aWlyWdI9Q2zvT0xAwk9005fxQMUS3CVwa6L7DaZYFPNmJ89RnPG+HPd7wSH8/Bo1
KVjJVtnyx3D+2E5viLnLE/+0it7JXF77BARNrsybJLIEHXfjXl9XBFj/BibL2ovG
8xrPzpt1N81qyDrm0AL1uYNYonsvK1uEKBa9qwYLTpgDTTp6KctJlXtmt7PR7op1
ntj5CsWZxpLC6AT6xH2knUGoDoRbE3F1iHKB2x0P77X1zGFp3Lc7UTnzBmwipTpW
5VPXVAC5vgZt/N5/z97dNuEmwkXXyYWW2SbL31EabBagv3cEP5N8swxTxpgrJaTs
4vu3teTneSSR77I2fc+YDeTBqw3uewpl0nfm66XsLW1KBSsAI/6iFBl4w1t8h/WH
rE/2/8Y49UobrppdoMFDVZf5ZD1sXnfD8fHUmYNFb+NsCYV+MaBukqZzujLw78C9
znZHLQzbGrzIK+xmPysgudCGJxpBlZ1kiD3S+ACwdqLW1UZrZ2c+Vcch000ueGVN
uT1e7eUZs1IkkggzIZjpEikrLuJzkVqkTiIS/aA4oW9qLYe/8xFJ8co/qU9SI04F
LygK4+bj6F4bzYtz2xnEGR4xYKgtV5J6MrRn7PbJUfmaUdMHwynAud6Npo5P071L
EugZH6HL1Wa+ep4YRrxgVmp6SWTWq7Rn6f6FAh1f+iIYcy9T/Sk3kfKVM0kA4cmb
5f1BE5hqxDswyI8dLBBczSgr0MUMNuP9WipzNmrLbvs4ZypB5zQH2xopPel1ZdkW
9iJZkiv4y21n5BjVbAayqdBJwexlkhwb2Ns26nY/kgGKZcdKSoERxvyRABYUTYq
Cj+CI32x7mjof77CjY10vMvMhDRFvX930zfWVngFRNFURlhtI7Q1Wq9FLqNgjSb5
Tza00aJbD60rqIfFLLhXTlqKY9qGs3fAqF0LwFgPyGGut9t2m9uD/YD//5ZZj/MR
w0VojznVJ8kuPVuKbiG+jHFUGxKUJQ97p6JCwnND0ZDA0rrQiBm/X5nxS2qA8rmT
p+b7brWoLEJlM5gUDJ02AYh81spKKThTUEXH1RT7+GTP03MWFOf4VDy5jbAwPMU
bHcEBpRbv8589a17YsS9u4BjGGohTGBtKEHtK7FhMmUd26sqc31HfzHsy5570dvA
P6y4dn+nmMI1C5M0vHfPSeuDNL0rD47MNHM2cJLWpRL09Q0KuqEGG7/kSnwFB76m
ruMDfzfEbBSRzSeA/uNzEBCjdzqZU3vwn0KEhQ1tG2vcmpq3P8g1Dh48LNJiBY3x
0TFe4bh36rIwB1L/fqMrVIUsv+DuuEyBqEX7LNBtWwXZ+vr0IK+De2n0H5d0pY3d
Vg3LSXSF65YF3uqe33aBoEOy9SIzjshngSEEjVCRvvWn0xAJ67aYk0ZFfzm5hTuU
rMiTYDT42sDA8Q02+pixdIrpC0tDERa8usQHP0msd/n5VsBaqu0YRKJw6k/gNWU1
oDjGuGgUJ41G2VjvreV7x3zj0ITNtLaXj0NzIVZI0Lurvn0F99FmMM8tS05wnUih
E2NpRqCs+LpUuN/JOpwmEenfGaFJ1jv6BXb+dHz728NHRU5Lezw+QBGVJR6i99Qz
quWH1yr6p+6Ykkcmjy/idyb5LZLDhQW3Yc5EYK4UdeJDXjYr1LNV64ncXbzmCEAF
Y5TD59BIFfl0E130yDniY0WbqJl6I7uPpmu1tfoTxUhbM7Hda2cHqQ5caJKYk0tk
lZFE4QKxuCoqI2cgn6vszkUrLPD/Yo+unFKQ5tBTNceqM0+YW6SNH75uRjVyT0sB
9GofTeyIxftebq5hof9+XRdPn8C6zQ0jnLv4D5KibJrart11XbNC5JWq1+u13/52
FudfRv5dUQcqqSXPJRTV+s330BYuDUfXnNxJk8y8V1bDbfTfgGwyWh3FopRcpd/K
s7PntnKET792spvx9RaHL15D3iWIC/xCbPSeMPsSDCc/VlDiZOYIwMT/GNvL4c4
blE6Ahq1BNg5S1bFuXh05I0Ma9ITqptkImZreHWAKg1RI2GWVHIrmPqpYNVzrTSS
05EarQa7Bd9dTDdJbsBX6jvvrq0zu/BdhySK/TNGEr3hE2u0+++M4nfjRqZnUqTcd
zyiXmW36jyWJxdF9FjrJpnkaRq2fB6+7a5hnBzIvIIQ0Cm+91uWU1i1z24vGM3FSB
a3fpLFX1p9ckiQG10FhpdfZoGMOacb3LpsAgxld46zBwhc7Rk00KR9N9jRRgCbAi
nlhHsZ7Gc1AVnnwLYYAq8BnXRerrkTIPvE4FbXzcJCL/IcTBQzyPM8sTDJnaDvcw

```

2aUopkGXDl9Cm8nreEnSxTAh0T9qRcWA9XDivGHDR0C171T1uEcL4ErM06YZReJN
9xPtsg3x2VouYo6V/VoG4c3Ia/chA56181yCGTrmgxIdJ5nSHUZrNMvx8vjdLu2a
qCKew79jYIyzRIoX0SM37lehkJuMRU7hfziMrC4fhVSjp16MX9fv7r51RLfJo8n/
n6hgrjDXmpSqzGRRatsCLjbYy/Bij7UljieM4uyst1Tb3bJvE0xrQRTQqcjEfEbx
oAnZkqiDy0qMU9EK5v1EnpAH4XEoaPut3Lezocj2CouAJFo9q71aM0FJ6HMAb9hM
jKpXuCG/h8xe9uPRXT5/cJCnz60aK1m4BGT6HBg++idJiH+dS4FBUm06CN/AubuZ
Kw0Fj0Rtohmmt+9RhBrxg8JrWFFp973R/W0NP1oA+TK6lJ9q56125ILHJ+saMwA0
93kz15TLPWIfGj/wvbnkmvPCAKCvxcaAUt7iiKRZBHGc1ZZ4KoNapkiIwJdGb9eh
N546WTMQ0vspzgjx6zkZWgAOGIaNmrCy07Ln+QEiaq0+wyBRYYG0mK6xvczS2U02
1+UJO20/xN4BEiktT2yN0NzsGjJET15vjpnE/wAAAAAAAAAAAAAAAAAAAAAakMEh4i
KDI8
-----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 { "ML-DSA-87 signed-data example with sig
ned attributes" }
        }
      }
      SET {
        SEQUENCE {
          INTEGER { 1 }
          SEQUENCE {
            SEQUENCE {
              SET {
                SEQUENCE {
                  # organizationName
                  OBJECT_IDENTIFIER { 2.5.4.10 }
                  PrintableString { "IETF" }
                }
              }
            }
            SET {
              SEQUENCE {
                # commonName
                OBJECT_IDENTIFIER { 2.5.4.3 }
                PrintableString { "LAMPS WG" }
              }
            }
          }
          INTEGER { `159ffe6f22fd5cc42c524df6fd5e28d0de38f34e` }
        }
      }
    }
  }
}

```

```

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 {
    # messageDigest
    OBJECT_IDENTIFIER { 1.2.840.113549.1.9.4 }
    SET {
      OCTET_STRING { `024f5ef2846bda2220e542208acfd715
ddd3b8e111e8390d62864b1dc128c0a2c9b74567b0b955c617f002204d27d887
95699e065f016ae31c6d0a4b42662264` }
    }
  }
}
SEQUENCE {
  OBJECT_IDENTIFIER { 2.16.840.1.101.3.4.3.19 }
}
OCTET_STRING { `9863de9a87725f55d7963b509e9a5496df4646
97c42d6b93d355de27d9c70f3188c57aa479288cb5b8aa993a728f9e75ec12ca
fdc25be154dc691f1580ab1a43f2692a526453d82c9dd1aeaa35c2116c0de25e
d2e34f2ea594c7b2d409db4911e546e0e2953f0ea7301b0f2f4e111398215521
5833e49dc5c67bb74a846e7685d477be4d32734fc0c4f7bbb42d78a18467aee1
a7a3fbfcac476ce116e51ef4cbb131d5656a27fa1836badcc390da857d87ad63
10f5b78e85ce066ff004d6f5043fc21bb13c8382765e2395e5b4c02229e3de86
cfb090e6618db2fc8303cae6f3e2d4e7a23f0e70d4514f70eaaa82f7d50c6dc9
67df4455a15708ff15750be0a12481be1e51fe7380f9fc8b76c271eb5221c857
5c1ef0eee3a04b8746700f99f05392de9a69faa302825cc66f2665e9b6c7ae65
ac12a4fcc742a2d623e9e0948b73073291dc4172f90b4db03a6d96f5e5877153
4730b6e139bf8fec94ff85c4eaf8a6782a94b2b8f27cda2f1d84f007915eb9d5
9c873db5b5b1f2c572aa624f9f66b3c547bb7dc9dada5c8ac4961e30811cb93a
ee0b2cd4f61792cf83201aefe55e308d0b180ddc45daa367eff566daafbd6fcd
4d403ec7b21c563c38c5a48a187e5e6ea430a1fcd177f3ae976a6bb939588e7a
7a71d2ccca3607c829065e67b9125fd6056d662cb08b852578c9d88fe87333ce
ff61827fa71af02159786f4c837d11934f0808d8f5365471d6d3b1f1e2b158eb
341d89a5b3d5909c8dd22f68aba5d605de1cc3480388f80fe1f778a9ee1832ba
a6fbe3126a54199becf424e47a456ac3bda64da698365b54cc4eb5cf2a842773
7409af003522b35188fe55a34811bc8bac06015add438a5d15b83180d3ded89c
40be7f313349e9edde655651274ef6eb4ffd679cfb42b6094f2b92ecdd6ac10d
a70a24ccd187fba3d727dbb1a46ceebc01c4c9a07f54df806dac77317072be6b
ff5346c618e1cb0e4eb944097815720266e0b09c501e8de0b05ccfabf2ddfbec
0de3fccd5e0dc562a95cfc21ecc0ed60a238023db2982f99a0949b96230b2623
b88213a8a4ae54ae9715e9fdb2e67d40a549e795e3cf8ecc049abff9d848f0bd
31ca652dd292e9fc69c8bb2fead7e38a9e8357513d5e15c235a2b35c07e2d326
23e4613a82c124fc7f9cb8124e8ab57078bdf4c3ed2998c22a331e3dddffdc52
5a8a9851e9deb3e51b4f3a558722296f08529ff657e238162fc974e9735371a2
0c1526acc7210138055e46f4428776b8e3ea3224cf78039090edf4f4ebfea570
fec7f83c298d44733797e657e963fe72ded410f88889c50230e0c2e4790e335
e0ac07ed4c3baeb90d7ba76191926eeac965ab2b6ae8af4b8a501e7819d8308
57b86a019dc5cb60619c7cb616796ea59242823387f1b4a57406c7d8d151f880

```

2d877a3dc09c8059f4a5de5c031fa162b2ead125eee7de53fb585d83b5c2cc56
ce8e36cc0a5f339c9b713a37432408b782884c2df141c0bb5137ce9572b86c7a
f5d73352f99cd7aafb0311caf6e1944d983b99eb8b4c332d931df0af9189f412
37a75458b57797b7186bebb5f7aa4373d8db5406df2aab789799d5b0404c3ef4
e086b85c57158ef471824519f93891d81087a9136aa35aec56b6fa91bb2865cf
1fab12259c17025d558fa28e210bb09a4c7c46697e0d67b9c3958f6004aa0b3f
dbeb412cb55ea94fe0444922769921653fd28935696765536e3cc2fd1f7b792b
9aaa01b50a3ae6ec6162ea97cb00c554fb4bd5a45e62b944a6abb6b4a4fefa17
b9255e7d3f1ebf8844af7dee8e7067394a6da71fb3d5e449e79e7920c5911480
429312e90b497568e1fdb58e03f586e6062f8eb3719ff7a7e94d59de2a6557f
b57105d247d94d4d127375169a1c4c5d7b69d2dd0ab15835f5cb55770b0b8521
701557d8f56d4f69270ea9d67bbdef8a64e54e230cefcd627a3a10ad9043bc025
3f76dd90cdfc01e066e93f6a81321f1f9df25f2bb7d8d1552f3766c9b6dcb341
8aa1f370e980927a5b50053598ec9bf249f0fabbe2acbeeae21729519aee6b6f
3afbfb066e54fea755504c8c663806bc7564883b236234c9a4caf8b98e1652464
f2a24b738123bcfc05b0165bdfa922dc5f65953acd3089cf4c2ce8f7c9935e55
084498a6aad5343482d6b0e8d6286b47fe9864fdc3f3ad0b496d4d67f2a21338
2db386132ba9ba8b2fd0ae1988453be3422be32dfdc4ba0547634f681d3cc6e
624986ef46f89ec1cb4d58aab72d85f0b6a39bca8bfb49f57f61b412671a3d6b
3750b5ab12a9da9c37b2dda217a82bf4dfb38a5802dd2cdf21390ddfce95905d
93e33f9a99109f0ade979144d5a8431ab428c93c0e8b227a2565ec20b3321019
4dbc1614ee26b01ded7e47b86730a6a66afa2557d118b53cc59979f97eea4322
d8f01c8616a3e277359a0ff2e3eee4e4bf21d2426af81ef8a622f298d0dc5fd0
19b479fa430298db3b95476a4d1cbaa185adcf123346d616049169868c0f512f
8e57af5210c0d8749045fbcdd0572a23ca3d65de03f894ea49c0596021a8cfc5
baa985aa934ea0210d867122fc44a977664045df9a7cbeb02cf45f448b07265
ff5cf4d3fc4717ee34a2038c1466a8f73665ff27fc87a4edf2b0f342fcef6bd7
ed1d6af4c0bbde9e0bca21ada2f93bd2ac7ad0c31c2fdc2fed487db4e6a69d8c
989fac14101af2f89462354cae78ffda8965854346828217b1f26ed018356ee3
81dac45e29b7078436cd96c2cc08f9dfad3ca06a70ea997cc20f38cea329b0f3
2c7b458404b3920c46a6bbbebb4f5648172de363b9f60c0b47cb688098d1ae67b
57657380db82dbea169b778ef512af1845a00985f9b14e8143572b274397b62e
b834905f4bc1b19c3003f1d1b25744536522f4469a600b01d51bd0c5a8df1d9f
bcf61d7f7221e039fa8334652862cd439b920323b2b31be0950c7b1571a1eae1
9df1679a96c187551b7828389c3a6f6390fca423151f5ec23cee1243a7183978
bd8451a2b6867399bf3cfcce7ad2476f35ea828738ced9647d995cfde7238327
f30feedaf5ffd7d8709c3b985464efcfe3d6b2b146564b136a637aced89f3417
ef5a9baa28c922c14e608ecf65bbe68d213bd792cebc0af8c52d416d9f55f81a
8ac228433a5c85799aebbe559eede2907dd5525aac7b608331766d98b7093314
e87d6a207a422f14eedbc5e12ddde0470148252edeb73e00724254d73027f2c0
baeedad3fd32780d19e33f99c917160ecb42cc9004bd3afcecf6881af82d54000
2f6a27af3e15a706c001632360781a8b327fe0fddb7d112643eeba2276f9c45e
66f66668506e6de578cacded1625b68036c9f0cfdbbea8f77d2c5923d96fea69
18a547a37a6980c5e6fb64d4f5d18df87b1b3c0643fb4fcbcf1244e18d696972
59d23d436cef4cec40c24f4e3b97f140c512dc25706ba2fb0da65814f36627cf
519cf1be1cf77bc121fcfc1a352958c956d9f2c770fed84e6f88b9cb13ffb48a
dec95c5efb04044daecc9b24b2041d77e35e5f570458ff0626cbda8bc6f31acf
ce9b7537cd6ac83ae63802f5b98358a27b2f2b5b842816bdab060b4cf8034d3a
7a29cb49957b66b7b3d1ee8a659ed8f90ac599c692c2e804fac47da49d41a80e
845b137175887281db138fefb5f5cc6169dcb73b5139f3066c22a53a56e553d7
5400b9be066dfcde7fcfdedd36e126c245d7c98595d926cbdf511a6c16a0bf77
043f937cb30c53c6982b25a4ece2fbb7b5e4e7792491efb2367dcf980de4c1ab
0dee7b0a653a77e6eba5ec2d6d4a052b0023fea2141978c35b7c87f587ac4ff6
ffc638f54a1baeba5da0c1435597f964396cc4d7c3f1f1d49983456fe36c0985
7e31a06e92a673ba32f0efc0bdce7647950cdb1abcc82bec663f2b20b9d08625
7a41959d64883dd2f800b076a2d6d5466b67673e55c721d0e3ae78654db93d5e
ede519b352249068332198e910892b2ee273915aa44c8892fda038a16f6a2d87
bff31149f1ca3fa94f52234e052f280ae3e6e3e85e1bcd8b73db19c4191e3160

} }

```
}  
}  
}
```

Acknowledgments

The authors would like to thank the following people for their contributions and reviews that helped shape this document: Viktor Dukhovni, Russ Housley, Panos Kampanakis, Mike Ounsworth, Falko Strenzke, Sean Turner, and Wei-Jun Wang.

This document was heavily influenced by [RFC8419], [RFC9814], and [RFC9881]. Thanks go to the authors of those documents.

Authors' Addresses

Ben Salter

UK National Cyber Security Centre
Email: ben.s3@ncsc.gov.uk

Adam Raine

UK National Cyber Security Centre
Email: adam.r@ncsc.gov.uk

Daniel Van Geest

CryptoNext Security
Email: daniel.vangeest@cryptonext-security.com