LAMPS
Internet-Draft
Intended status: Standards Track
Expires: 13 August 2026

V. Smyslov
ELVIS-PLUS

9 February 2026

Internet X.509 Public Key Infrastructure - Algorithm Identifiers for FrodoKEM
draft-smyslov-lamps-frodokem-certificates-00

Abstract

FrodoKEM is an unstructured lattice-based Key Encapsulation Mechanism (KEM). Compared to ML-KEM, FrodoKEM is considered as having more conservative design. This document specifies the conventions for using FrodoKEM in X.509 Public Key Infrastructure. The conventions for the subject public keys and private keys are also specified.

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 13 August 2026.

Table of Contents

1. Introduction

FrodoKEM [I-D.longa-cfrg-frodokem] is an unstructured lattice-based Key Encapsulation Mechanism (KEM). At the time of writing this document, FrodoKEM is being standardized in ISO (International Organization for Standardization) as a quantum-resistant key-encapsulation mechanism.

This document specifies the use of FrodoFEM in Public Key Infrastructure X.509 (PKIX) certificates [RFC5280] at two security levels: FrodoKEM-976 and FrodoKEM-1344, using object identifiers assigned by ISO. The private key format is also specified.

1.1. Applicability Statement

FrodoKEM certificates are used in protocols where the public key is used to generate and encapsulate a shared secret used to derive a symmetric key used to encrypt a payload, like in CMS. To be used in TLS, FrodoKEM certificates could only be used as end-entity identity certificates and would require significant updates to the protocol; see, for example, [I-D.celi-wiggers-tls-authkem].

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Algorithm Identifiers

The AlgorithmIdentifier type is defined in [RFC5912] 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 [RFC5912] and is compatible with the 2021 ASN.1 syntax [X680]. See [RFC5280] for the 1988 ASN.1 syntax.

The fields in AlgorithmIdentifier have the following meanings:

* algorithm identifies the cryptographic algorithm with an object identifier.

* parameters, which are optional, are the associated parameters for the algorithm identifier in the algorithm field.

The AlgorithmIdentifier for a FrodoKEM public key MUST use one of the object identifiers (OID) from ISO listed below, based on the security level. The parameters field of the AlgorithmIdentifier for the FrodoKEM public key MUST be absent.

```
  frodokem OBJECT IDENTIFIER ::= { iso(1) standard(0)
  encryption-algorithms(18033) part2(2)
  key-encapsulation-mechanism(2) 7 }

  id-kem-frodokem976-shake OBJECT IDENTIFIER ::= { frodokem 1 }

  id-kem-frodokem1344-shake OBJECT IDENTIFIER ::= { frodokem 2 }

  id-kem-efrodokem976-shake OBJECT IDENTIFIER ::= { frodokem 3 }

  id-kem-efrodokem1344-shake OBJECT IDENTIFIER ::= { frodokem 4 }

  id-kem-frodokem976-aes OBJECT IDENTIFIER ::= { frodokem 5 }

  id-kem-frodokem1344-aes OBJECT IDENTIFIER ::= { frodokem 6 }

  id-kem-efrodokem976-aes OBJECT IDENTIFIER ::= { frodokem 7 }

  id-kem-efrodokem1344-aes OBJECT IDENTIFIER ::= { frodokem 8 }
```

4. Subject Public Key Fields

In the X.509 certificate, the subjectPublicKeyInfo field has the SubjectPublicKeyInfo type, which has the following ASN.1 syntax:

```
SubjectPublicKeyInfo {PUBLIC-KEY: IOSet} ::= SEQUENCE {
    algorithm        AlgorithmIdentifier {PUBLIC-KEY, {IOSet}},
    subjectPublicKey BIT STRING
}
```

The fields in SubjectPublicKeyInfo have the following meaning:

* algorithm is the algorithm identifier and parameters for the public key (see above).

* subjectPublicKey contains the byte stream of the public key.

For each FrodoKEM security level, see Table 1, we define a PUBLIC-KEY ASN.1 type as follows.

```
pk-frodokem976-shake PUBLIC-KEY ::= {
  IDENTIFIER id-kem-frodokem976-shake
  -- KEY no ASN.1 wrapping; 15632 octets --
  PARAMS ARE absent
  CERT-KEY-USAGE { keyEncipherment }
  PRIVATE-KEY frodokem976-shake-PrivateKey -- defined in Section 6
  }

pk-frodokem1344-shake PUBLIC-KEY ::= {
  IDENTIFIER id-kem-frodokem1344-shake
  -- KEY no ASN.1 wrapping; 21520 octets --
  PARAMS ARE absent
  CERT-KEY-USAGE { keyEncipherment }
  PRIVATE-KEY frodokem1344-shake-PrivateKey -- defined in Section 6
  }

pk-efrodokem976-shake PUBLIC-KEY ::= {
  IDENTIFIER id-kem-efrodokem976-shake
  -- KEY no ASN.1 wrapping; 15632 octets --
  PARAMS ARE absent
  CERT-KEY-USAGE { keyEncipherment }
  PRIVATE-KEY efrodokem976-shake-PrivateKey -- defined in Section 6
  }

pk-efrodokem1344-shake PUBLIC-KEY ::= {
  IDENTIFIER id-kem-efrodokem1344-shake
  -- KEY no ASN.1 wrapping; 21520 octets --
  PARAMS ARE absent
```

```
    CERT-KEY-USAGE { keyEncipherment }
    PRIVATE-KEY efrodokem1344-shake-PrivateKey -- defined in Section 6
    }

  pk-frodokem976-aes PUBLIC-KEY ::= {
    IDENTIFIER id-kem-frodokem976-aes
    -- KEY no ASN.1 wrapping; 15632 octets --
    PARAMS ARE absent
    CERT-KEY-USAGE { keyEncipherment }
    PRIVATE-KEY frodokem976-aes-PrivateKey -- defined in Section 6
    }

  pk-frodokem1344-aes PUBLIC-KEY ::= {
    IDENTIFIER id-kem-frodokem1344-aes
    -- KEY no ASN.1 wrapping; 21520 octets --
    PARAMS ARE absent
    CERT-KEY-USAGE { keyEncipherment }
    PRIVATE-KEY frodokem1344-aes-PrivateKey -- defined in Section 6
    }

  pk-efrodokem976-aes PUBLIC-KEY ::= {
    IDENTIFIER id-kem-efrodokem976-aes
    -- KEY no ASN.1 wrapping; 15632 octets --
    PARAMS ARE absent
    CERT-KEY-USAGE { keyEncipherment }
    PRIVATE-KEY efrodokem976-aes-PrivateKey -- defined in Section 6
    }

  pk-efrodokem1344-aes PUBLIC-KEY ::= {
    IDENTIFIER id-kem-efrodokem1344-aes
    -- KEY no ASN.1 wrapping; 21520 octets --
    PARAMS ARE absent
    CERT-KEY-USAGE { keyEncipherment }
    PRIVATE-KEY efrodokem1344-aes-PrivateKey -- defined in Section 6
    }

  frodokem976-shake-PublicKey ::= OCTET STRING (SIZE (15632))

  frodokem1344-shake-PublicKey ::= OCTET STRING (SIZE (21520))

  efrodokem976-shake-PublicKey ::= OCTET STRING (SIZE (15632))

  efrodokem1344-shake-PublicKey ::= OCTET STRING (SIZE (21520))

  frodokem976-aes-PublicKey ::= OCTET STRING (SIZE (15632))

  frodokem1344-aes-PublicKey ::= OCTET STRING (SIZE (21520))
```

```
efrodokem976-aes-PublicKey ::= OCTET STRING (SIZE (15632))

efrodokem1344-aes-PublicKey ::= OCTET STRING (SIZE (21520))
```

When a FrodoKEM public key appears outside of a SubjectPublicKeyInfo type in an environment that uses ASN.1 encoding, it can be encoded as an OCTET STRING by using the frodokem976-shake-PublicKey, frodokem1344-shake-PublicKey, efrodokem976-shake-PublicKey, efrodokem1344-shake-PublicKey, frodokem976-aes-PublicKey, frodokem1344-aes-PublicKey, efrodokem976-aes-PublicKey, and efrodokem1344-aes-PublicKey types corresponding to the correct key size.

[RFC5958] describes the Asymmetric Key Package's OneAsymmetricKey type for encoding asymmetric keypairs. When a FrodoKEM private key or keypair is encoded as a OneAsymmetricKey, it follows the description in Section 6.

5. Key Usage Bits

The intended application for the key is indicated in the keyUsage certificate extension; see Section 4.2.1.3 of [RFC5280]. If the keyUsage extension is present in certificates, then keyEncipherement MUST be the only key usage set for certificates that indicate id-kem-frodokem976-shake, id-kem-frodokem1344-shake, id-kem-efrodokem976-shake, id-kem-efrodokem1344-shake, id-kem-frodokem976-aes, id-kem-frodokem1344-aes, id-kem-efrodokem976-aes, id-kem-efrodokem1344-aes in SubjectPublicKeyInfo.

6. Private Key Format

"Asymmetric Key Packages" [RFC5958] describes how to encode a private key in a structure that both identifies which algorithm the private key is for and allows for the public key and additional attributes about the key to be included as well. For illustration, the ASN.1 structure OneAsymmetricKey is replicated below.

```
OneAsymmetricKey ::= SEQUENCE {
  version                   Version,
  privateKeyAlgorithm       SEQUENCE {
  algorithm                 PUBLIC-KEY.&id({PublicKeySet}),
  parameters                PUBLIC-KEY.&Params({PublicKeySet}
                              {@privateKeyAlgorithm.algorithm})
                                 OPTIONAL}
  privateKey                OCTET STRING (CONTAINING
                              PUBLIC-KEY.&PrivateKey({PublicKeySet}
                                {@privateKeyAlgorithm.algorithm})),
  attributes           [0] Attributes OPTIONAL,
  ...,
  [[2: publicKey       [1] BIT STRING (CONTAINING
                              PUBLIC-KEY.&Params({PublicKeySet}
                                {@privateKeyAlgorithm.algorithm})
                                OPTIONAL ]],
  ...
}

...

PrivateKey ::= OCTET STRING
                    -- Content varies based on type of key. The
                    -- algorithm identifier dictates the format of
                    -- the key.
```

For FrodoKEM private keys, the privateKey field in OneAsymmetricKey contains the OCTET STRING representation of the FrodoKEM private key.

The privateKeyAlgorithm field uses the AlgorithmIdentifier structure with the appropriate OID as defined in Section 3.

The publicKey field contains the byte stream of the public key. If present, the publicKey field will hold the encoded public key as defined in Section 4.

7. Security Considerations

The Security Considerations section of [RFC5280] applies to this specification as well.

Protection of the private-key information, i.e., the seed, is vital to public-key cryptography. Disclosure of the private-key material to another entity can lead to masquerades.

The generation of private keys relies on random numbers. The use of inadequate pseudo-random number generators (PRNGs) to generate these values can result in little or no security. An attacker may find it

much easier to reproduce the PRNG environment that produced the keys, searching the resulting small set of possibilities, rather than brute force searching the whole key space. The generation of quality random numbers is difficult.

For more detailed FrodoKEM specific security considerations refer to [I-D.longa-cfrg-frodokem].

8. IANA Considerations

For the ASN.1 Module in Appendix A, IANA is requested to assign an object identifier (OID) for the module identifier (TBD) with a Description of "id-mod-frodokem-kem-2026". The OID for the module should be allocated in the "SMI Security for PKIX Module Identifier" registry (1.3.6.1.5.5.7.0).

9. References

9.1. Normative References

[I-D.longa-cfrg-frodokem]
Longa, P., Bos, J. W., Ehlen, S., and D. Stebila, "FrodoKEM: key encapsulation from learning with errors", Work in Progress, Internet-Draft, draft-longa-cfrg-frodokem-01, 16 September 2025, <https://datatracker.ietf.org/doc/html/draft-longa-cfrg-frodokem-01>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

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

[RFC5912] Hoffman, P. and J. Schaad, "New ASN.1 Modules for the Public Key Infrastructure Using X.509 (PKIX)", RFC 5912, DOI 10.17487/RFC5912, June 2010, <https://www.rfc-editor.org/info/rfc5912>.

[RFC5958] Turner, S., "Asymmetric Key Packages", RFC 5958, DOI 10.17487/RFC5958, August 2010, <https://www.rfc-editor.org/info/rfc5958>.

[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>.

[RFC9629] Housley, R., Gray, J., and T. Okubo, "Using Key Encapsulation Mechanism (KEM) Algorithms in the Cryptographic Message Syntax (CMS)", RFC 9629, DOI 10.17487/RFC9629, August 2024, <https://www.rfc-editor.org/info/rfc9629>.

[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>.

9.2. Informative References

[I-D.celi-wiggers-tls-authkem] Wiggers, T., Celi, S., Schwabe, P., Stebila, D., and N. Sullivan, "KEM-based Authentication for TLS 1.3", Work in Progress, Internet-Draft, draft-celi-wiggers-tls-authkem-06, 4 November 2025, <https://datatracker.ietf.org/doc/html/draft-celi-wiggers-tls-authkem-06>.

Appendix A. ASN.1 Module

This appendix includes the ASN.1 module [X680] for FrodoKEM. Note that as per [RFC5280], certificates use the Distinguished Encoding Rules; see [X690]. This module imports objects from [RFC5912] and [RFC9629].

```
<CODE BEGINS>
X509-FRODOKEM-2026
{ iso(1) identified-organization(3) dod(6)
  internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
  id-mod-frodokem-kem-2026(TBD) }

DEFINITIONS IMPLICIT TAGS ::= BEGIN

EXPORTS ALL;

IMPORTS

 PUBLIC-KEY
   FROM AlgorithmInformation-2009  -- [RFC 5912]
     { iso(1) identified-organization(3) dod(6) internet(1)
       security(5) mechanisms(5) pkix(7) id-mod(0)
```

```
          id-mod-algorithmInformation-02(58) }

    KEM-ALGORITHM
      FROM KEMAlgorithmInformation-2023  -- [RFC 9629]
        { iso(1) identified-organization(3) dod(6) internet(1)
          security(5) mechanisms(5) pkix(7) id-mod(0)
          id-mod-kemAlgorithmInformation-2023(109) };

   --
   -- FrodoKEM Identifiers
   --

   frodokem OBJECT IDENTIFIER ::= { iso(1) standard(0)
   encryption-algorithms(18033) part2(2)
   key-encapsulation-mechanism(2) 7 }

   id-kem-frodokem976-shake OBJECT IDENTIFIER ::= { frodokem 1 }

   id-kem-frodokem1344-shake OBJECT IDENTIFIER ::= { frodokem 2 }

   id-kem-efrodokem976-shake OBJECT IDENTIFIER ::= { frodokem 3 }

   id-kem-efrodokem1344-shake OBJECT IDENTIFIER ::= { frodokem 4 }

   id-kem-frodokem976-aes OBJECT IDENTIFIER ::= { frodokem 5 }

   id-kem-frodokem1344-aes OBJECT IDENTIFIER ::= { frodokem 6 }

   id-kem-efrodokem976-aes OBJECT IDENTIFIER ::= { frodokem 7 }

   id-kem-efrodokem1344-aes OBJECT IDENTIFIER ::= { frodokem 8 }

   --
   -- Public Key Algorithms
   --

   PublicKeys PUBLIC-KEY ::= {
     -- This expands PublicKeys from [RFC 5912]
     pk-frodokem976-shake |
     pk-frodokem1344-shake |
     pk-efrodokem976-shake |
     pk-efrodokem1344-shake |
     pk-frodokem976-aes |
     pk-frodokem1344-aes |
     pk-efrodokem976-aes |
     pk-efrodokem1344-aes,
     ...
     }
```

```
   --
   -- FrodoKEM Public Keys
   --

   pk-frodokem976-shake PUBLIC-KEY ::= {
     IDENTIFIER id-kem-frodokem976-shake
     -- KEY no ASN.1 wrapping; 15632 octets --
     PARAMS ARE absent
     CERT-KEY-USAGE { keyEncipherment }
     PRIVATE-KEY frodokem976-shake-PrivateKey -- defined in Section 6
     }

   pk-frodokem1344-shake PUBLIC-KEY ::= {
     IDENTIFIER id-kem-frodokem1344-shake
     -- KEY no ASN.1 wrapping; 21520 octets --
     PARAMS ARE absent
     CERT-KEY-USAGE { keyEncipherment }
     PRIVATE-KEY frodokem1344-shake-PrivateKey -- defined in Section 6
     }

   pk-efrodokem976-shake PUBLIC-KEY ::= {
     IDENTIFIER id-kem-efrodokem976-shake
     -- KEY no ASN.1 wrapping; 15632 octets --
     PARAMS ARE absent
     CERT-KEY-USAGE { keyEncipherment }
     PRIVATE-KEY efrodokem976-shake-PrivateKey -- defined in Section 6
     }

   pk-efrodokem1344-shake PUBLIC-KEY ::= {
     IDENTIFIER id-kem-efrodokem1344-shake
     -- KEY no ASN.1 wrapping; 21520 octets --
     PARAMS ARE absent
     CERT-KEY-USAGE { keyEncipherment }
     PRIVATE-KEY efrodokem1344-shake-PrivateKey -- defined in Section 6
     }

   pk-frodokem976-aes PUBLIC-KEY ::= {
     IDENTIFIER id-kem-frodokem976-aes
     -- KEY no ASN.1 wrapping; 15632 octets --
     PARAMS ARE absent
     CERT-KEY-USAGE { keyEncipherment }
     PRIVATE-KEY frodokem976-aes-PrivateKey -- defined in Section 6
     }

   pk-frodokem1344-aes PUBLIC-KEY ::= {
     IDENTIFIER id-kem-frodokem1344-aes
     -- KEY no ASN.1 wrapping; 21520 octets --
     PARAMS ARE absent
```

```
    CERT-KEY-USAGE { keyEncipherment }
    PRIVATE-KEY frodokem1344-aes-PrivateKey -- defined in Section 6
    }

  pk-efrodokem976-aes PUBLIC-KEY ::= {
    IDENTIFIER id-kem-efrodokem976-aes
    -- KEY no ASN.1 wrapping; 15632 octets --
    PARAMS ARE absent
    CERT-KEY-USAGE { keyEncipherment }
    PRIVATE-KEY efrodokem976-aes-PrivateKey -- defined in Section 6
    }

  pk-efrodokem1344-aes PUBLIC-KEY ::= {
    IDENTIFIER id-kem-efrodokem1344-aes
    -- KEY no ASN.1 wrapping; 21520 octets --
    PARAMS ARE absent
    CERT-KEY-USAGE { keyEncipherment }
    PRIVATE-KEY efrodokem1344-aes-PrivateKey -- defined in Section 6
    }

  frodokem976-shake-PublicKey ::= OCTET STRING (SIZE (15632))

  frodokem1344-shake-PublicKey ::= OCTET STRING (SIZE (21520))

  efrodokem976-shake-PublicKey ::= OCTET STRING (SIZE (15632))

  efrodokem1344-shake-PublicKey ::= OCTET STRING (SIZE (21520))

  frodokem976-aes-PublicKey ::= OCTET STRING (SIZE (15632))

  frodokem1344-aes-PublicKey ::= OCTET STRING (SIZE (21520))

  efrodokem976-aes-PublicKey ::= OCTET STRING (SIZE (15632))

  efrodokem1344-aes-PublicKey ::= OCTET STRING (SIZE (21520))

  END
  <CODE ENDS>
```

Appendix B. Parameter Set Security and Sizes

Instead of defining the strength of a quantum algorithm in a traditional manner using the imprecise notion of bits of security, NIST has defined security levels by picking a reference scheme, which NIST expects to offer notable levels of resistance to both quantum and classical attack. To wit, a KEM algorithm that achieves NIST PQC security must require computational resources to break IND-CCA security comparable or greater than that required for key search on

AES-128, AES-192, and AES-256 for Levels 1, 3, and 5, respectively. Levels 2 and 4 use collision search for SHA-256 and SHA-384 as reference.

Level	Parameter Set	Public Key pk	Secret Key sk	Ciphertext ct	Shared Secret ss
3	FrodoKEM-976	15,632	31,296	15,792	24
3	eFrodoKEM-976	15,632	31,296	15,744	24
5	FrodoKEM-1344	21,520	43,088	21,696	32
5	eFrodoKEM-1344	21,520	43,088	21,632	32

Table 1: Mapping between NIST Security Level, FrodoKEM parameter set, and sizes in bytes

Acknowledgments

Most of the text was impudently stolen from draft-ietf-lamps-kyber-certificates.

Author’s Address

Valery Smyslov
ELVIS-PLUS
Russian Federation
Email: svan@elvis.ru