lamps
Internet-Draft
Intended status: Standards Track
Expires: 15 August 2026

M. Chen
L. Su
China Mobile
G. Wang
Huawei Int. Pte Ltd

11 February 2026

Use of FrodoKEM in the Cryptographic Message Syntax
draft-chen-lamps-cms-frodokem-00

Abstract

FrodoKEM is a quantum-resistant key encapsulation mechanism (KEM) based on the standard Learning With Errors (LWE) problem. It is one of three KEMs in the process of ISO standardization. This document specifies the conventions for using eight variants of FrodoKEM in the Cryptographic Message Syntax (CMS), by employing the KEMRecipientInfo structure defined in "Use of Key Encapsulation Mechanism (KEM) Algorithms in the Cryptographic Message Syntax (CMS)" [RFC9629]. These eight variants of FrodoKEM are (e)FrodoKEM-976-AES and (e)FrodoKEM-976-SHAKE for security level 3, and (e)FrodoKEM-1344-AES and (e)FrodoKEM-1344-SHAKE for security level 5.

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 15 August 2026.

Copyright Notice

Table of Contents

1. Introduction

FrodoKEM is one of three KEMs in the process of ISO standardization [FrodoKEM]. Its security is based on a well-studied hard problem in unstructured lattices, called the learning with errors problem. The algorithm details of FrodoKEM are specified in [I-D.LBES25] . FrodoKEM has both AES and SHAKE variants to offer optimized performance across different hardware platforms. AES variants are highly suitable for devices with hardware acceleration for AES (like AES-NI on Intel processors). SHAKE variants provide competitive or better performance on platforms lacking AES hardware acceleration (such as many embedded systems and general-purpose CPUs). To cover both scenarios, this specification include both variants.

"Using Key Encapsulation Mechanism (KEM) Algorithms in the Cryptographic Message Syntax (CMS)" [RFC9629] defines the KEMRecipientInfo structure for the use of KEM algorithms in the CMS enveloped-data, authenticated-data, and authenticated-enveloped-data content types. This document specifies the conventions for the direct use of eight variants of FrodoKEM in the CMS with the

KEMRecipientInfo structure: (e)FrodoKEM-976-AES and (e)FrodoKEM-976-SHAKE for security level 3, and (e)FrodoKEM-1344-AES and (e)FrodoKEM-1344-SHAKE for security level 5. Note that these are all variants of FrodoKEM to be standardized in ISO.

1.1. 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].

1.2. FrodoKEM

In total, FrodoKEM [I-D.LBES25] [FrodoKEM] has 12 variants. Namely, it offers 3 NIST security levels 1, 3, and 5; the pseudorandom generator (PRG) uses AES128 or SHAKE 128; and the KEM public key can be a long-term key (standard mode) or a short-term key (ephemeral mode).

According to the current standardization progress in ISO, FrodoKEM will be standardized for the 8 variants for NIST security levels 3 and 5. Namely, there are (e)FrodoKEM-976 and (e)FrodoKEM-1344, but not (e)FrodoKEM-640 for security level 1. To align with ISO, this specification specifies the use of (e)FrodoKEM varaints for security levels 3 and 5 only, not variants for security level 1.

Based on the above, this document specifies only eight variants of (e)FrodoKEM for Cryptographic Message Syntax. Namely, (e)FrodoKEM-976-AES and (e)FrodoKEM-976-SHAKE for security level 3, and (e)FrodoKEM-1344-AES and (e)FrodoKEM-1344-SHAKE for security level 5.

Key encapsulation mechanism (KEM) is a kind of key exchange, which allows one entity to encapsulate a secret under a (long-term or ephemeral) public key of another entity. A KEM consists of three algorithms:

* KeyGen(k) -> (pk, sk): A probabilistic key generation algorithm, which generates a public encapsulation key pk and a secret decapsulation key sk, when a security parameter k is given.

* Encaps(pk) -> (ct, ss): A probabilistic encapsulation algorithm, which takes as input a public encapsulation key pk and outputs a ciphertext ct and a shared secret ss.

* Decaps(sk, ct) -> ss: A decapsulation algorithm, which takes as input a secret decapsulation key sk and ciphertext ct and outputs a shared secret ss.

Table 1 summarizes the sizes of keys, ciphertext, and shared secret for all variants of FrodoKEM with security levels 3 and 5. Note that using either AES128 or SHAKE 128 does not affect these sizes.

Level	Algorithms	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: Size (in bytes) of keys and ciphertexts of FrodoKEM

2. Use of the FrodoKEM Algorithm in the CMS

The FrodoKEM algorithm MAY be used for one or more recipients in the CMS enveloped-data content type [RFC5652], the CMS authenticated-data content type [RFC5652], or the CMS authenticated-enveloped-data content type [RFC5083]. In each case, the KEMRecipientInfo [RFC9629] structure is used with the FrodoKEM algorithm to securely transport a content-encryption key from an originator to one ore multiple recipients.

The steps for processing FrodoKEM with KEMRecipientInfo follow Section 2 of [RFC9629]. To support the FrodoKEM algorithm, a CMS originator MUST implement the Encapsulate() function, and a CMS recipient MUST implement the Decapsulate() function.

2.1. RecipientInfo Conventions

When the FrodoKEM algorithm is used for a recipient, the RecipientInfo choice for that recipient MUST be the OtherRecipientInfo choice using the KEMRecipientInfo structure defined in [RFC9629]. The fields of KEMRecipientInfo have the following meanings:

* version is the syntax version number; it MUST be 0.

* rid identifies the recipient’s certificate or public key.

* kem identifies the KEM algorithm; it MUST contain one of 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 or id-kem-efrodokem1344-aes. These identifiers are reproduced in Section 3.

* kemct is the ciphertext generated for this recipient.

* kdf identifies the key derivation algorithm. Implementations MUST support the HKDF [RFC5869] with SHA-256 [FIPS180] using the id-alg-hkdf-with-sha256 KDF object identifier [RFC8619]. As specified in [RFC8619], when this object identifier appears in an ASN.1 type AlgorithmIdentifier, the parameters field MUST be absent. Implementations MAY support other KDFs.

* kekLength is the size of the key-encryption key in octets.

* ukm is an optional input to the key derivation function. The secure use of FrodoKEM in the CMS does not depend upon the use of the ukm value, and therefore this document has no requirements for this value. See Section 3 of [RFC9629] for more information on the ukm parameter.

* wrap identifies the key-encryption algorithm used to encrypt the content-encryption key.

  - Implementations supporting FrodoKEM-976-AES or FrodoKEM-976-SHAKE MUST support the AES-Wrap-192 Key Wrap algorithm [RFC3394], using the id-aes192-wrap key-encryption algorithm object identifier [RFC3565].

  - Implementations supporting FrodoKEM-1344-AES or FrodoKEM-1344-SHAKE MUST support the AES-Wrap-256 Key Wrap algorithm [RFC3394], using the id-aes256-wrap key-encryption algorithm object identifier [RFC3565].

  - Implementations MAY support other key-encryption algorithms.

2.2. Underlying Components

When FrodoKEM is used in CMS, the underlying components used in the KEMRecipientInfo structure SHOULD be consistent with the desired minimum security level. To meet the requirements for the KDF and key-wrap algorithm from Section 7 of [RFC9629], the table below provides the minimum requirements for the components used with FrodoKEM.

Security Strength	Algorithm	KDF preimage strength	Symmetric key encryption strength
192-bit	(e)FrodoKEM-976	192-bit	192-bit
256-bit	(e)FrodoKEM-1344	256-bit	256-bit

Table 2: FrodoKEM KEMRecipientInfo component security levels

2.3. Use of the HKDF-based Key Derivation Function

The HKDF function is a composition of the HKDF-Extract and HKDF-Expand functions.

HKDF(salt, IKM, info, L) = HKDF-Expand(HKDF-Extract(salt, IKM), info, L)

When used with KEMRecipientInfo, the salt parameter is unused, that is it is the zero-length string "". The IKM, info and L parameters correspond to the same KDF inputs from Section 5 of [RFC9629]. The info parameter is independently generated by the originator and recipient. Implementations MUST confirm that L is consistent with the key size of the key-encryption algorithm.

2.4. Certificate Conventions

[RFC5280] specifies the profile for X.509 certificates used in Internet applications. FrodoKEM requires a static public key for the recipient, which the originator obtains from the recipient’s certificate. The conventions for carrying a FrodoKEM public key are specified in [I-D.S26].

2.5. SMIME Capabilities Attribute Conventions

Section 2.5.2 of [RFC8551] defines the SMIMECapabilities attribute for advertising a partial list of algorithms that an S/MIME implementation can support. When constructing a CMS SignedData content type [RFC5652], a compliant implementation MAY include the SMIMECapabilities attribute to announce support for one or more of the FrodoKEM algorithm identifiers.

The SMIMECapability SEQUENCE representing a FrodoKEM algorithm MUST contain one of the FrodoKEM object identifiers in the capabilityID field. When one of the FrodoKEM object identifiers appears in the capabilityID field, the parameters MUST NOT be present.

3. Identifiers

The identifiers for indicating the use of FrodoKEM in the CMS are defined in [CSOR] and [RFC8619]. For convenience, they are reproduced here.

```
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 }

id-alg-hkdf-with-sha256 OBJECT IDENTIFIER ::= { iso(1)
   member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9)
   smime(16) alg(3) 28 }

id-alg-hkdf-with-sha256 OBJECT IDENTIFIER ::= { iso(1)
   member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9)
   smime(16) alg(3) 28 }

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

id-aes192-wrap OBJECT IDENTIFIER ::= { aes 25 }

id-aes256-wrap OBJECT IDENTIFIER ::= { aes 45 }

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

id-shake256 OBJECT IDENTIFIER ::= { hashAlgs 12 }
```

4. Security Considerations

The security considerations sections of [I-D.draft-smyslov-lamps-frodokem-certificates] and [RFC9629] apply to this specification as well.

Implementations MUST protect the FrodoKEM private key, the key-encryption key, the content-encryption key, the message-authentication key, and the content-authentication-encryption key. Of these keys, all but the private key are ephemeral and MUST be erased after use. Compromise of the FrodoKEM private key can lead to the compromise of all messages protected with that key.

The generation of the private key and the FrodoKEM encapsulation function depend on random numbers. The use of inadequate pseudo-random number generators (PRNGs) to generate these values can result in little or no security. Generation of high-quality random numbers is difficult; see [FRODO-SPEC] for more information.

The encapsulation and decapsulation of FrodoKEM only output the shared secret and the ciphertext. Implementations MUST NOT use the intermediate values directly for any purpose. Implementations SHOULD NOT leak information about the intermediate values or computations through timing or other "side channels", as an attacker may be able to determine information about keying data and/or the recipient’s private key.

In general, it is good cryptographic practice to use a given FrodoKEM key pair in only one scheme. This practice avoids the risk that a vulnerability in one scheme could compromise the security of the other.

5. IANA Considerations

TBD

6. Acknowledgements

This document borrows heavily from [W-D.POV26], [RFC9629], and the FrodoKEM specification [I-D.LBES25]. Thanks go to the authors of those documents.

7. Normative References

[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>.

[RFC3394] Schaad, J. and R. Housley, "Advanced Encryption Standard (AES) Key Wrap Algorithm", RFC 3394, DOI 10.17487/RFC3394, September 2002, <https://www.rfc-editor.org/rfc/rfc3394>.

[RFC3565] Schaad, J., "Use of the Advanced Encryption Standard (AES) Encryption Algorithm in Cryptographic Message Syntax (CMS)", RFC 3565, DOI 10.17487/RFC3565, July 2003, <https://www.rfc-editor.org/rfc/rfc3565>.

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

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

[RFC5869] Krawczyk, H. and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", RFC 5869, DOI 10.17487/RFC5869, May 2010, <https://www.rfc-editor.org/rfc/rfc5869>.

[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>.

[RFC8551] Schaad, J., Ramsdell, B., and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification", RFC 8551, DOI 10.17487/RFC8551, April 2019, <https://www.rfc-editor.org/rfc/rfc8551>.

[RFC8619] Housley, R., "Algorithm Identifiers for the HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", RFC 8619, DOI 10.17487/RFC8619, June 2019, <https://www.rfc-editor.org/rfc/rfc8619>.

[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/rfc/rfc9629>.

[CSOR] NIST, "Computer Security Objects Register", , August 2024, <https://csrc.nist.gov/projects/computer-security-objects-register/algorithm-registration>.

[I-D.LBES25]
Longa, P., Bos, J. W., Ehlen, S., and D. Stebila, "FrodoKEM: key encapsulation from learning with errors", Work in Progress (v01), Internet-Draft, September 2025, <https://datatracker.ietf.org/doc/draft-longa-cfrg-frodokem/>.

8. Informative References

[FrodoKEM] Alkim, E., Bos, J. W., Ducas, L., Longa, P., Mironov, I., Naehrig, N., Nikolaenko, V., Peikert, C., Raghunathan, A., and D. Stebila, "FrodoKEM: Learning With Errors Key Encapsulation", Standardization Proposal submitted to ISO , September 2025, <https://frodokem.org/files/FrodoKEM_standard_proposal_20250929.pdf>.

[W-D.POV26]
Prat, J., Ounsworth, M., and D. Van Geest, "Use of ML-KEM in the Cryptographic Message Syntax (CMS)", Work in Progress (v13), Internet-Draft (Group Documet of LAMPS), IETF, February 2026, <https://datatracker.ietf.org/doc/draft-ietf-lamps-cms-kyber/>.

[I-D.S26] Smyslov, V., "Internet X.509 Public Key Infrastructure - Algorithm Identifiers for FrodoKEM", Work in Progress (v00), Internet-Draft, IETF, February 2026, <https://datatracker.ietf.org/doc/draft-smyslov-lamps-frodokem-certificates/>.

Authors' Addresses

Meiling Chen
China Mobile
BeiJing
China
Email: chenmeiling@chinamobile.com

Li Su
China Mobile
BeiJing
China
Email: suli@chinamobile.com

Guilin Wang
Huawei Int. Pte Ltd
Singapore

Email: wang.guilin@huawei.com