EUROPEAN UNION AGENCY
FOR CYBERSECURITY

Hybridization of traditional cryptographic mechanisms with PQC

Standardisation Status

30/04/2026

DISCLAIMER

This report provides an analysis of standardisation efforts related to hybridization of traditional cryptographic mechanisms with PQC. It does not establish a recommendation from ENISA and the ECCG subgroup on cryptography despite providing a mapping with ACM version 2.0 (April 2025).

It can serve as an inspiration for the implementation of hybridization and may later be used by the ECCG subgroup on cryptography to establish recommendations for preferred hybridization modes.

Outline

- ECCG Agreed Cryptographic Mechanisms (ACMs)
- IETF Introduction
- Hybrid Key Agreement Frameworks
- Concrete Hybrid Key Agreement Mechanisms
- Hybrid Digital Signature Frameworks
- Concrete Hybrid Digital Signature Mechanisms
- Hybrid Standardisation Status Summary

Section 1

ECCG Agreed Cryptographic Mechanisms

AGREED CRYPTOGRAPHIC MECHANISMS

ECCG Agreed Cryptographic Mechanisms - version 2

For the purposes of hybridization, recommends:

- Asymmetric Constructions
  - Digital Signature
  - Key Establishment
  - Key Encapsulation
  - Asymmetric Encryption
- Key Combiners (Key agreement hybridization)
- Does not specify Digital Signature hybridization

https://certification.enisa.europa.eu/publications/eucc-guidelines-cryptography_en

ASYMMETRIC CONSTRUCTIONS

Non-Hybridized Schemes

Recommended standalone schemes:

- XMSS
- LMS
- SLH-DSA

Legacy:

- FF-DLOG
- RSA-PKCS#1.5 Asymmetric Encryption

ASYMMETRIC CONSTRUCTIONS

Agreed Digital Signature Schemes

Algorithm	Parameters
RSA-PSS	n ≥ 3000, log2 (e) > 16
RSA-PKCS#1.5	n ≥ 3000, log2 (e) > 16
EC-DSA	[1]
ML-DSA	ML-DSA-65 ML-DSA-87

Agreed Key Agreement Schemes

Algorithm	Parameters
RSA-OAEP[2]	n ≥ 3000, log2 (e) > 16
EC-DH	[1]
ML-KEM	ML-KEM-768 ML-KEM-1024
FrodoKEM	FrodoKEM-976 FrodoKEM-1344

[1] EC Parameters: BrainpoolP256r1, BrainpoolP384r1, BrainpoolP512r1, NIST P-256, NIST P-384, NIST P-521
[2] Hybridized as a Key Agreement Scheme

KEY COMBINERS

ETSI TS 103 744 CatKDF & CasKDF

Hybrid Key Establishment Frameworks

- IND-CPA in ROM as long as one component is OW-CPA
- CatKDF is IND-CCA in ROM as long as one component is OW-CCA
- Security for generic component algorithms relies on including the public keys and/or ciphertexts

AGREED CRYPTOGRAPHIC MECHANISMS

Colour Coding

The following colour coding is used throughout these slides to indicate recommended, legacy, and non-recommended ACM Mechanisms as hybrid components.

Recommended Mechanisms
Legacy Mechanisms
Non-Recommended Mechanisms

e.g.

MLDSA44-RSA2048-PKCS15-SHA256
MLDSA44-ECDSA-P256-SHA256
MLDSA65-RSA3072-PSS-SHA512
MLDSA65-RSA3072-PKCS15-SHA512

Section 2

IETF Introduction

IETF DOCUMENT MATURITY

Document State	Notes	Next Step	Maturity
Individual Draft e.g. draft-stebila-tls-hybrid-design	Anyone can publish an Individual Draft	Working group call for adoption	No standing
Working Group (WG) Draft e.g. draft-ietf-tls-hybrid-design	Control transfers to the working group	Working group last call	In progress, may change
IESG Review	Sent for wider IETF review	Submit to RFC Editor	Nearing completion, likely technically stable
RFC Editor	Final editorial changes	Publication	Technically stable
RFC	Common types: Standards Track, Experimental, Informational		Published

Colour coding for document state is used throughout this document

HYBRID KEY AGREEMENT STANDARDISATION STATUS

Protocol	Framework	ML-KEM	FrodoKEM	ECDH	BP-ECDH	RSA-OAEP	X-ECDH
[CFRG]	WG	WG		WG			WG
HPKE		WG		WG			WG
PKIX/X.509		IESG		WG	WG	WG	WG
TLS	RFC Ed	RFC Ed		RFC Ed			RFC Ed
IKEv2	RFC	IESG	WG	RFC	RFC		RFC
CMS		IESG		IESG	IESG	IESG	IESG
S/MIME							
SSH		RFC Ed		RFC Ed			RFC Ed
OpenPGP							WG
COSE	IESG	Individual		Individual			Individual
JOSE	IESG	Individual		Individual			Individual
MLS		WG		WG			WG
EAP-AKA’		WG		WG			WG

WORKING GROUPS

IETF Working Groups

- 130+ Working Groups (WGs) across 7 areas
- Security Area (SEC): 31 WGs
- 11 SEC WGs with hybridization work
- +CFRG IRTF Research Group

Hybridization-Relevant Working Groups	
COSE	EMU
HPKE	IPSECME
JOSE	LAMPS
MLS	OPENPGP
PQUIP*	SSHM
TLS	CFRG**

* Informational PQ support/guidance group
** IRTF research group

POST-QUANTUM USE IN PROTOCOLS (PQUIP)

RFC 9794: Terminology for Post-Quantum Traditional Hybrid Schemes

PQ/T hybrid Key Encapsulation Mechanism (KEM):

A multi-algorithm KEM made up of two or more component algorithms where at least one is a post-quantum algorithm and at least one is a traditional algorithm.

PQ/T hybrid digital signature:

A multi-algorithm digital signature scheme made up of two or more component digital signature algorithms where at least one is a post-quantum algorithm and at least one is a traditional algorithm.

Section 3

Hybrid Key Agreement Frameworks

Hybrid Key Agreement Frameworks

Hybrid key agreement frameworks approved or undergoing standardisation:

- ETSI: TS 103 744 v1.2.2 - Hybrid Key Establishment Schemes
- NIST: SP 800-227 - Approved Key Combiners
- CFRG: Hybrid PQ/T Key Encapsulation Mechanisms
- TLS: Hybrid key exchange in TLS 1.3

Hybrid Key Agreement Frameworks

ETSI TS 103 744

Concatenate-based (CatKDF) and cascade-based (CasKDF) combiners.

36 allowed (ML-KEM, ECDH, KDF) parameter sets.

Allowed traditional components:

- NIST P-384, NIST P-256, brainpoolP384r1, brainpoolP256r1, Curve448, Curve25519

Allowed KDFs:

- HKDF/SHA256, HKDF/SHA384, HMAC/SHA256, HMAC/SHA384, KMAC128, KMAC256

Allowed CasKDF PRFs:

- HMAC/SHA256, HMAC/SHA384, KMAC128, KMAC256

ETSI TS 103 744

CatKDF	$context = f(info, pk_1, pk_2, ct_1, ct_2)$ $KDF(k_1 \| k_2, label, context, length)$

CatKDF

1) Form $secret = psk \| k_1 \| k_2$.
2) Set $context = f(info, MA, MB)$, where f is a context formatting function.
3) $key\_material = KDF(secret, label, context, length)$.
4) Return $key\_material$.

3 allowed KDF forms:

CatKDF	$context = cahb\_f(info, pk_1, pk_2, ct_1, ct_2)$ $HMAC(label, [01]_{32} \| k_1 \| k_2 \| context))$
CatKDF	$context = cb\_f(info, pk_1, pk_2, ct_1, ct_2)$ $KMAC(label, [01]_{32} \| k_1 \| k_2 \| context, len, \text{“KDF”}))$
CatKDF	$context = cahb\_f(info, pk_1, pk_2, ct_1, ct_2)$ $kdk = HKDF\text{-}Extract(label, k_1 \| k_2)$ $HKDF\text{-}Expand(kdk, length, context)$

CasKDF	$ss_1 = KDF(PRF(\emptyset, k_1, pk_1, ct_1), label_1, info_1, length_1)$ $ss = KDF(PRF(ss_1, k_2, pk_2, ct_2), label_2, info_2, length_2)$

CasKDF

1) $chain\_secret_0 = psk$, a k_len value or the empty string, $\emptyset$.
2) For $i = 1, 2$.
   a) $round\_secret_i = PRF(chain\_secret_{i-1}, k_i, MA_i, MB_i)$.
   b) $chain\_secret_i \| key\_material_i = KDF(round\_secret_i, label_i, info_i, k\_len + length_i)$.
3) Return $chain\_secret_1$, $chain\_secret_2$, and $key\_material_1$, $key\_material_2$.

Similar HMAC, KMAC, and HKDF KDF forms as CatKDF.

ETSI TS 103 744

CatKDF + CasKDF Key Takeaways

- Exactly 2 keys (k_1, k_2) + optional PSK
- All public keys and ciphertexts mixed into KDF (*MA*, *MB*)
- Context formatting function, *f*, ensures injective encoding of context data
- CasKDF PRF uses context formatting function + MAC:
  - *HMAC(secret, cahb_f(val1, val2, …))*
    - *cahb_f* is a length-encoding concatenate-and-hash formatting function
  - *KMAC (secret, cb_f(val1, val2, …)*
    - *cb_f* is a length-encoding concatenating formatting function
    - If the PSK is not used in CasKDF, *secret* is a string of zeroes

Hybrid Key Agreement Frameworks

NIST SP800-227 Approved Key Combiners

References SP 800-56C for one-step and two-step key derivation

- $K \leftarrow KDM((S_1, S_2, \cdots, S_t), OtherInput)$
- Allows combiners with multiple secrets if at least one is approved

References SP 800-133

- All secrets must be approved, so we ignore SP 800-133

Unlike TS 103 744:

- Does not require public keys and ciphertexts in *OtherInput*.
- Does not require a context formatting function

NIST SP 800-227

SP 800-56C One-Step Key Derivation

1) For *i = 1 to reps = ceil(L / H_outputBits)*:
   a) Increment counter by 1
   b) Compute *K(i) = H(counter || Z || FixedInfo)*
   c) Set *Result(i) = Result(i – 1) ||K(i)*

H: approved Hash, HMAC, or KMAC

Simplified:

1Step	$H([01]_{32}(optional) \| k_1 \| k_2 \| FixedInfo)$

- Fixed output size based on *H* (*reps* = 1).
- $[01]_{32}$ counter is optional for reps = 1.

CatKDF	$context = f(info, pk_1, pk_2, ct_1, ct_2)$ $KDF(k_1 \| k_2, label, context, length)$

Compared to CatKDF

- CatKDF can't be used with a Hash
- CatKDF with HMAC or KMAC is a NIST-approved one-step KDF
- CatKDF with HMAC, *f()* does length-encoded concat-and-hash

CatKDF	$context = cahb\_f(info, pk_1, pk_2, ct_1, ct_2)$ $HMAC(label, [01]_{32} \| k_1 \| k_2 \| context))$

- CatKDF with KMAC, *f()* does length-encoded concat

CatKDF	$context = cb\_f(info, pk_1, pk_2, ct_1, ct_2)$ $KMAC(label, [01]_{32} \| k_1 \| k_2 \| context, len, "KDF"))$

- CatKDF requires $[01]_{32}$ counter

NIST SP 800-227

SP 800-56C Two-Step Key Derivation

1) *KDK = MAC(salt, Z, …)*
2) *Result = KDF(KDK, L, {IV,} FixedInfo)*

HKDF is an approved two-step key derivation

2Step	***kdk = Extract(salt, k_1 || k_2)*** ***Expand(kdk, length, FixedInfo)***

CatKDF	***context = f(info, pk_1, pk_2, ct_1, ct_2)*** ***KDF(k_1 || k_2, label, context, length)***

Compared to CatKDF

- When used with HKDF, CatKDF is an approved two-step key derivation function
- CatKDF with HKDF, *f()* does length-encoded concat-and-hash

CatKDF	***context = cahb_f(info, pk_1, pk_2, ct_1, ct_2)*** ***kdk = HKDF-Extract(label, k_1 || k_2)*** ***HKDF-Expand(kdk, length, context)***

Hybrid Key Agreement Frameworks

IETF Hybrid Key Agreement Frameworks

WG	Title	Document	Status
CFRG	Hybrid PQ/T Key Encapsulation Mechanisms	draft-irtf-cfrg-hybrid-kems	WG Draft
TLS	Hybrid key exchange in TLS 1.3	draft-ietf-tls-hybrid-design	RFC Ed
IPSECME	Multiple Key Exchanges in the Internet Key Exchange Protocol Version 2 (IKEv2)	RFC 9370	RFC
MLS	Amortized PQ MLS Combiner	draft-ietf-mls-combiner	WG Draft (WGLC)
CFRG	Chempat: Generic Instantiated PQ/T Hybrid Key Encapsulation Mechanisms	draft-josefsson-chempat	Individual (Recently Expired)

IETF HYBRID KEY AGREEMENT FRAMEWORKS

CFRG - Hybrid PQ/T Key Encapsulation Mechanisms

Universal Combiner with a Nominal Group

UG	$KDF(k_{pq} \| k_t \| c_{pq} \| c_t \| p_{pq} \| p_t \| label)$

C2PRI Combiner with a Nominal Group

CG	$KDF(k_{pq} \| k_t \| c_t \| p_t \| label)$

- PQ must have Ciphertext Second Preimage Resistance (C2PRI)

KDF indifferentiable from a random oracle.

CatKDF	$KDF(k_1 \| k_2, label, f(info, pk_1, pk_2, ct_1, ct_2), length)$

Compared to CatKDF

- C2PRI Combiner does not include PQ public key and ciphertext
- Inputs are assumed to be fixed length
- No context formatting function
- Assumes a simple single-parameter KDF
- No obvious mapping between the two

IETF HYBRID KEY AGREEMENT FRAMEWORKS

TLS - Hybrid key exchange in TLS 1.3

- TLS 1.3 ONLY
- Concatenation of fixed key combinations
- TLS 1.3 key schedule mixes in hash of ClientHello and ServerHello, containing public keys and ciphertexts

```
          |
          v
k1||k2 -> HKDF-Extract = Handshake Secret
          |
          +-----> Derive-Secret(., "c hs traffic",
          |                     ClientHello...ServerHello)
          |                     = client_handshake_traffic_secret
          |
          +-----> Derive-Secret(., "s hs traffic",
          |                     ClientHello...ServerHello)
          |                     = server_handshake_traffic_secret
          v
```

CatKDF	$KDF(k_1 \| \| k_2, label, f(info, pk_1, pk_2, ct_1, ct_2), length)$

Compared to CatKDF

Generally, maps well to CatKDF:

- TLS 1.3 key schedule uses HKDF
- The transcript hash doesn't map exactly to *cahb_f*, but TLS messages are a combination of fixed-length and length-encoded fields, so it is very similar
- With no PSK, the label to the Extract

CatKDF	$context = cahb\_f(info, pk_1, pk_2, ct_1, ct_2)$ $kdk = Extract(label, k_1 \| \| k_2)$ $Expand(kdk, length, context)$

IETF HYBRID KEY AGREEMENT FRAMEWORKS

IPSECME - RFC 9370: Multiple Key Exchanges in the Internet Key Exchange Protocol Version 2 (IKEv2)

Framework allowing up to 8 registered key agreement algorithms to be hybridized

- Avoids IP fragmentation issues in IKEv2 initial exchange
- IKE SA (initial IKEv2 establishment) behaves similarly to CasKDF
- Rekey and Child SA behaves similarly to CatKDF

IETF HYBRID KEY AGREEMENT FRAMEWORKS

IPSECME - RFC 9370 IKE SA Establishment

- Multiple rounds of key agreement, each updating the symmetric keys
- *prf()* followed by *prf+()* is equivalent to an Extract-Expand KDF
  - When HMAC is used, this is HKDF

Round 1:

```
SKEYSEED = prf(Ni | Nr, g^ir)
KEYMAT = prf+ (SKEYSEED, Ni | Nr | SPIi | SPIr)
```

Round 2..N:

```
SKEYSEED(n) = prf(SK_d(n-1), SK(n) | Ni | Nr)
KEYMAT(n) = prf+ (SKEYSEED(n), Ni | Nr | SPIi | SPIr)
```

CasKDF	$ss_1 = KDF(PRF(\emptyset, k_1, pk_1, ct_1), label_1, info_1, length_1)$ $ss = KDF(PRF(ss_1, k_2, pk_2, ct_2), label_2, info_2, length_2)$

Compared to CasKDF

- Public keys and ciphertexts aren't included in the KDF
- Variable-length nonces may be used as the KDF salt, CasKDF requires them to be fixed-length
- CasKDF does a PRF of the secrets, public keys and ciphertexts before passing them to the KDF

IETF HYBRID KEY AGREEMENT FRAMEWORKS

IPSECME - RFC 9370 Rekey & Child SA Establishment

- Multiple rounds of key agreement with one single key derivation at the end
  - Deals with simultaneous rekeying
- IKE SA Rekey
  - KDF Extract-Expand:

```
SKEYSEED = prf(SK_d, SK(0) | Ni | Nr | SK(1) | ... | SK(n))
{SK_d | SK_ai | SK_ar | SK_ei | SK_er | SK_pi | SK_pr}
              = prf+ (SKEYSEED, Ni | Nr | SPIi | SPIr)
```

- Child SA
  - KDF Expand with previous secret:

```
KEYMAT = prf+ (SK_d, SK(0) | Ni | Nr | SK(1) | ... | SK(n))
```

CatKDF	$KDF(k_1 \| k_2, label, f(info, pk_1, pk_2, ct_1, ct_2), length)$

Compared to CatKDF

- Public keys and ciphertexts aren't included in the KDF
- Child SA:
  - Shared secrets are passed to KDF-Expand, not KDF-Extract

IETF HYBRID KEY AGREEMENT FRAMEWORKS

MLS - Amortized PQ MLS Combiner

Runs a traditional session with a PQ session in parallel with synchronization

- PQ session is updated less frequently
- Secret is exported from PQ session and imported to traditional session as PSK
- Hybridizes key agreement, authentication, or both

Compared to CasKDF/CatKDF

- Maps to neither CasKDF nor CatKDF.
- Conceptual similarities to CasKDF where hybridization occurs in stages.

IETF HYBRID KEY AGREEMENT FRAMEWORKS

CFRG - Chempat

More generic than Hybrid PQ/T Key Encapsulation Mechanisms or Hybrid key exchange in TLS 1.3

- Designed for generic security independent of component algorithms
- More complicated/less performant than other frameworks

Compared to CatKDF

- KDF is simple 1-parameter KDF (SHA3-256)
- Different content formatting function, non-length delimited
  - Implicitly assumes inputs are fixed-length
- No obvious mapping between the two

Section 4

Concrete Hybrid Key Agreement Mechanisms

HYBRID KEY AGREEMENT MECHANISMS

CFRG - Concrete Hybrid PQ/T Key Encapsulation Mechanisms

Status: **WG Draft**

- Concrete instantiations of Hybrid PQ/T Key Encapsulation Mechanisms
- Compared to CatKDF:
  - Uses SHA3-256 as KDF: not allowed
  - Does not use CatKDF's required $[01]_{32}$ counter.

Hybrid Algorithms

MLKEM768-P256 (SHA3-256 KDF)

MLKEM768-X25519 (SHA3-256 KDF)

MLKEM1024-P384 (SHA3-256 KDF)

MLKEM768-X25519 is also known as X-Wing

HYBRID KEY AGREEMENT MECHANISMS

LAMPS - Composite ML-KEM for use in X.509 Public Key Infrastructure

Status: **IESG Review**

- 3 algorithms align with Concrete Hybrid PQ/T Key Encapsulation Mechanisms
- 9 additional concrete instantiations of Hybrid PQ/T Key Encapsulation Mechanisms
- Different private key encoding than CFRG
- All use C2PRICombiner since ML-KEM is C2PRI

Hybrid Algorithms

MLKEM768-RSA2048-SHA3-256
MLKEM768-RSA3072-SHA3-256
MLKEM768-RSA4096-SHA3-256
MLKEM768-X25519-SHA3-256
MLKEM768-ECDH-P256-SHA3-256
MLKEM768-ECDH-P384-SHA3-256
MLKEM768-ECDH-brainpoolP256r1-SHA3-256
MLKEM1024-RSA3072-SHA3-256
MLKEM1024-ECDH-P384-SHA3-256
MLKEM1024-ECDH-brainpoolP384r1-SHA3-256
MLKEM1024-X448-SHA3-256
MLKEM1024-ECDH-P521-SHA3-256

HYBRID KEY AGREEMENT MECHANISMS

HPKE - Post-Quantum and Post-Quantum/Traditional Hybrid Algorithms for HPKE

Status: **WG Draft**

- "H" in HPKE is a different "Hybrid"
  - Key agreement + symmetric encryption
- Adds algorithms from Concrete Hybrid PQ/T Key Encapsulation Mechanisms to HPKE
- Uses SHA3-256 as KDF

HPKE Hybrid Algorithms

MLKEM768-P256 (SHA3-256 KDF)

MLKEM768-X25519 (SHA3-256 KDF)

MLKEM1024-P384 (SHA3-256 KDF)

HYBRID KEY AGREEMENT MECHANISMS

TLS - Post-quantum hybrid ECDHE-MLKEM Key Agreement for TLSv1.3

Status: **RFC Editor**

Concrete instantiations of Hybrid key exchange in TLS 1.3

Hybrid Algorithms

X25519MLKEM768

SecP256r1MLKEM768

SecP384r1MLKEM1024

HYBRID KEY AGREEMENT MECHANISMS

TLS – RFC 9849: TLS Encrypted Client Hello

Status: **RFC**

- Hybridization ready - uses HPKE for public key encryption
- Once Post-Quantum and Post-Quantum/Traditional Hybrid Algorithms for HPKE is standardised, any of those algorithms could be offered in the ECHConfig
- Uses SHA3-256 as KDF

Hybrid Algorithms

MLKEM768-P256 (SHA3-256 KDF)

MLKEM768-X25519 (SHA3-256 KDF)

MLKEM1024-P384 (SHA3-256 KDF)

HYBRID KEY AGREEMENT MECHANISMS

IPSECME – Post-quantum Key Exchange with ML-KEM in the Internet Key Exchange Protocol Version 2 (IKEv2)

Status: **IESG Review**

Specifies use of ML-KEM standalone or using RFC 9370: Multiple Key Exchanges in the Internet Key Exchange Protocol Version 2 (IKEv2).

Hybrid Algorithms

Can be hybridized with up to 7 other algorithms

HYBRID KEY AGREEMENT MECHANISMS

IPSECME – Post-quantum Hybrid Key Exchange in IKEv2 with FrodoKEM

Status: **WG Draft**

Specifies use of FrodoKEM standalone or using RFC 9370: Multiple Key Exchanges in the Internet Key Exchange Protocol Version 2 (IKEv2).

Notes

Blocked waiting for a publicly available FrodoKEM standard which can be normatively referenced

Hybrid Algorithms:

Can be hybridized with up to 7 other algorithms

HYBRID KEY AGREEMENT MECHANISMS

IPSECME – PQ/T Hybrid Composite Key Exchange and Reliable Transport for IKEv2

Status: **Individual Draft**

Hybridization without RFC 9370

- Requires a reliable transport because of large keys
- Uses CFRG's UniversalCombiner

Notes

REMINDER: Individual Drafts have no standing at IETF

- Should really use one of the CFRG concrete hybrid KEMs
- KDF is undefined

Hybrid Algorithms:

ecp256-mlkem768

ecp384-mlkem1024

curve25519-mlkem768

HYBRID KEY AGREEMENT MECHANISMS

CMS – Composite ML-KEM for use in Cryptographic Message Syntax (CMS)

Status: **IESG Review**

Applies the Composite ML-KEM algorithms from Composite ML-KEM for use in X.509 Public Key Infrastructure to CMS

Hybrid Algorithms

MLKEM768-RSA2048-SHA3-256
MLKEM768-RSA3072-SHA3-256
MLKEM768-RSA4096-SHA3-256
MLKEM768-X25519-SHA3-256
MLKEM768-ECDH-P256-SHA3-256
MLKEM768-ECDH-P384-SHA3-256
MLKEM768-ECDH-brainpoolP256r1-SHA3-256
MLKEM1024-RSA3072-SHA3-256
MLKEM1024-ECDH-P384-SHA3-256
MLKEM1024-ECDH-brainpoolP384r1-SHA3-256
MLKEM1024-X448-SHA3-256
MLKEM1024-ECDH-P521-SHA3-256

HYBRID KEY AGREEMENT MECHANISMS

S/MIME

S/MIME uses CMS

- CMS has ongoing hybrid key agreement work
- No ongoing work to make hybrid key agreement mandatory in S/MIME

HYBRID KEY AGREEMENT MECHANISMS

SSH - PQ/T Hybrid Key Exchange with ML-KEM in SSH

Status: **RFC Editor**

Generally maps well to CatKDF:

- SSH includes entire handshake as key schedule input
- The transcript hash doesn't map exactly to *cahb_f*, but SSH messages are a combination of fixed-length and length-encoded fields, so it is very similar
- SHA-256 or SHA-384 are used as KDF
  - Not allowed in CatKDF

Hybrid Algorithms

mlkem768nistp256-sha256

mlkem1024nistp384-sha384

mlkem768x25519-sha256

HYBRID KEY AGREEMENT MECHANISMS

OpenPGP - Post-Quantum Cryptography in OpenPGP

Status: **RFC Editor**

- Uses CFRG's C2PRICombiner
- KDF = SHA3-256
- Does not map to CFRG's concrete hybrid KEMs, a different label is used

Hybrid Algorithms

ML-KEM-768+X25519

ML-KEM-1024+X448

HYBRID KEY AGREEMENT MECHANISMS

COSE - Use of Hybrid Public-Key Encryption (HPKE) with CBOR Object Signing and Encryption (COSE)

Status: **IESG Review**

Adding HPKE to COSE, will be able to adopt hybrid KEMs through this

HYBRID KEY AGREEMENT MECHANISMS

JOSE - Use of Hybrid Public Key Encryption (HPKE) with JSON Web Encryption (JWE)

Status: **IESG Review**

Adding HPKE to JOSE, will be able to adopt hybrid KEMs through this

HYBRID KEY AGREEMENT MECHANISMS

COSE/JOSE - Post-Quantum and Hybrid KEMs for HPKE with JOSE and COSE

Status: **Individual Draft**

Uses HPKE to add KEM algorithms to COSE and JOSE

Uses SHA3-256 as KDF

Notes

REMINDER: Individual Drafts have no standing at IETF

Hybrid Algorithms:

MLKEM768 + P-256 (SHA3-256 KDF)

MLKEM768 + X25519 (SHA3-256 KDF)

MLKEM1024 + P-384 (SHA3-256 KDF)

HYBRID KEY AGREEMENT MECHANISMS

MLS - ML-KEM and Hybrid Cipher Suites for Messaging Layer Security

Status: **WG Draft**

Uses HPKE to add KEM algorithms to MLS

Uses SHA3-256 as KDF

Hybrid Algorithm Cipher Suites

MLS_128_MLKEM768X25519_AES128GCM_SHA256_Ed25519 (SHA3-256 KDF)

MLS_128_MLKEM768X25519_AES256GCM_SHA384_Ed25519 (SHA3-256 KDF)

MLS_128_MLKEM768P256_AES128GCM_SHA256_P256 (SHA3-256 KDF)

MLS_128_MLKEM768P256_AES256GCM_SHA384_P256 (SHA3-256 KDF)

MLS_192_MLKEM1024P384_AES256GCM_SHA384_P384 (SHA3-256 KDF)

HYBRID KEY AGREEMENT MECHANISMS

EAP-AKA’ - Enhancing Security in EAP-AKA' with Hybrid Post-Quantum Cryptography

Status: **WG Draft**

Adds PQ/T Hybrid Key Exchange to the Forward Secrecy Extension to the Improved Extensible Authentication Protocol Method for Authentication and Key Agreement (EAP-AKA' FS).

Uses HPKE to add KEM algorithms to EAP-AKA’

Hybrid Algorithms

ML-KEM-768,P-256 (SHA3-256 KDF)

ML-KEM-768,X25519 (HKDF-SHA256 KDF)

SUMMARY

Protocol	Title	Document	Status
[CFRG]	Concrete Hybrid PQ/T Key Encapsulation Mechanisms	draft-irtf-cfrg-concrete-hybrid-kems	WG Draft
PKIX / X.509	Composite ML-KEM for use in X.509 Public Key Infrastructure	draft-ietf-lamps-pq-composite-kem	IESG Review
HPKE	Post-Quantum and Post-Quantum / Traditional Hybrid Algorithms for HPKE	draft-ietf-hpke-pq	WG Draft
TLS	Post-quantum hybrid ECDHE-MLKEM Key Agreement for TLSv1.3	draft-ietf-tls-ecdhe-mlkem	RFC Editor
TLS	TLS Encrypted Client Hello	RFC 9849	RFC
IKEv2	Post-quantum Key Exchange with ML-KEM in the Internet Key Exchange Protocol Version 2 (IKEv2)	draft-ietf-ipsecme-ikev2-mlkem	IESG Review

SUMMARY

Protocol	Title	Document	Status
IKEv2	Post-quantum Hybrid Key Exchange in IKEv2 with FrodoKEM	draft-ietf-ipsecme-hybrid-kem-ikev2-Frodo	WG Draft
IKEv2	PQ/T Hybrid Composite Key Exchange and Reliable Transport for IKEv2	draft-reddy-ipsecme-ikev2-hybrid-reliable	Individual Draft
CMS	Composite ML-KEM for use in Cryptographic Message Syntax (CMS)	draft-ietf-lamps-cms-composite-sigs	IESG Review
SSH	PQ/T Hybrid Key Exchange with ML-KEM in SSH	draft-ietf-sshm-mlkem-hybrid-kex	RFC Editor
OpenPGP	Post-Quantum Cryptography in OpenPGP	draft-ietf-openpgp-pqc	RFC Editor

SUMMARY

Protocol	Title	Document	Status
COSE	Use of Hybrid Public-Key Encryption (HPKE) with CBOR Object Signing and Encryption (COSE)	draft-ietf-cose-hpke	IESG Review
JOSE	Use of Hybrid Public Key Encryption (HPKE) with JSON Web Encryption (JWE)	draft-ietf-jose-hpke-encrypt	IESG Review
COSE & JOSE	Post-Quantum and Hybrid KEMs for HPKE with JOSE and COSE	draft-reddy-cose-jose-pqc-hybrid-hpke	Individual Draft
MLS	ML-KEM and Hybrid Cipher Suites for Messaging Layer Security	draft-ietf-mls-pq-ciphersuites	WG Draft
EAP-AKA’	Enhancing Security in EAP-AKA' with Hybrid Post-Quantum Cryptography	draft-ietf-emu-hybrid-pqc-eapaka	WG Draft

Section 5

Hybrid Digital Signature Frameworks

Hybrid Digital Signature Frameworks

Unlike for Key Agreement, there are no regulatory frameworks for specific hybrid digital signature frameworks.

- ETSI TR 103 966: Mentions "and mode" and "or mode" where two signatures are presented and one or both are verified. It also gives some hybrid signature examples.
- NIST'S Post-Quantum Cryptography FAQ allows dual signatures and requires them both to be successfully verified.

Hybrid Digital Signature Frameworks

Standardised Hybrid Digital Signature frameworks:

- ITU-T: Recommendation X.509 (10/19) – Alternative algorithm extensions
- IETF: RFC 9763 - Related Certificates for Use in Multiple Authentications within a Protocol
- IETF: RFC 4739 - Multiple Authentication Exchanges in the Internet Key Exchange (IKEv2) Protocol

MLS - Amortized PQ MLS Combiner could also be considered hybrid authentication when used with PQ signature ciphersuites.

Hybrid Digital Signature undergoing standardisation:

- ASC X9: X9.146 Quantum TLS – Work to add TLS extensions to make use of X.509's alternative algorithm extensions for hybridization

CFRG - Hybrid Digital Signatures with Strong Unforgeability (Individual Draft)

Hybrid Digital Signature Frameworks

ITU-T X.509 (10/19) Alternative Algorithm Extensions

- Defines non-critical alternative subject public key and alternative signature extensions for X.509 certificates and CRLs
- Intended to be used for migration, i.e. “or mode”
- Could just as easily be used in “and mode” with a protocol profile

Hybrid Digital Signature Frameworks

RFC 9763: Related Certificates for Use in Multiple Authentications within a Protocol

- Specifies a mechanism to provide assurance that two X.509 certificates are owned by the same entity
- Another mechanism would be needed to use those certificates in a hybrid manner
- No protocols currently make use of this mechanism to provide hybrid digital signatures

Hybrid Digital Signature Frameworks

RFC 4739: Multiple Authentication Exchanges in the Internet Key Exchange (IKEv2) Protocol

- Experimental RFC which allows a peer to send an additional authentication if it wishes
- Does not mandate or commit to hybrid authentication
- Does not specify or commit to particular algorithm combinations
- No guarantees that both authentications are from the same entity
  - Guarantees could be added by using RFC 9763: Related Certificates for Use in Multiple Authentications within a Protocol

Hybrid Digital Signature Frameworks

X9.146 Quantum TLS

- Draft standard by ASC X9F5 Financial PKI workgroup
- Uses ITU-T X.509 (10/19) Alternative Algorithm Extension certificates within TLS
- Adds TLS Certificate Key Selection extensions to negotiate the use of native, alternative, or both keys when signing the TLS handshake

Hybrid Digital Signature Frameworks

CFRG - Hybrid Digital Signatures with Strong Unforgeability

- Individual Draft presented to CFRG
- SUF-CMA if the PQ algorithm is SUF-CMA
  - Compared to other proposals where the hybrid is only SUF-CMA if both components are SUF-CMA and one is deterministic

Section 6

Concrete Hybrid Digital Signature Mechanisms

HYBRID DIGITAL SIGNATURE MECHANISMS

PKIX/X.509 - Composite ML-DSA for use in X.509 Public Key Infrastructure

Status: **RFC Editor**

- Explicit combinations or ML-DSA with RSA, ECDSA, and EdDSA
- EUF-CMA secure against quantum adversaries if the traditional component is broken
- EUF-CMA secure against traditional adversaries if ML-DSA is broken
- Not SUF-CMA secure

Hybrid Algorithms

MLDSA44-RSA2048-PSS-SHA256
MLDSA44-RSA2048-PKCS15-SHA256
MLDSA44-Ed25519-SHA512
MLDSA44-ECDSA-P256-SHA256
MLDSA65-RSA3072-PSS-SHA512
MLDSA65-RSA3072-PKCS15-SHA512
MLDSA65-RSA4096-PSS-SHA512
MLDSA65-RSA4096-PKCS15-SHA512
MLDSA65-ECDSA-P256-SHA512
MLDSA65-ECDSA-P384-SHA512
MLDSA65-ECDSA-brainpoolP256r1-SHA512
MLDSA65-Ed25519-SHA512
MLDSA87-ECDSA-P384-SHA512
MLDSA87-ECDSA-brainpoolP384r1-SHA512
MLDSA87-Ed448-SHAKE256
MLDSA87-RSA3072-PSS-SHA512
MLDSA87-RSA4096-PSS-SHA512
MLDSA87-ECDSA-P521-SHA512

HYBRID DIGITAL SIGNATURE MECHANISMS

TLS - Use of Composite ML-DSA in TLS 1.3

Status: **Individual Draft**

- Explicit combinations or ML-DSA with RSA, ECDSA, and EdDSA
- EUF-CMA secure against quantum adversaries if the traditional component is broken
- EUF-CMA secure against traditional adversaries if ML-DSA is broken
- Not SUF-CMA secure

Notes

REMINDER: Individual Drafts have no standing at IETF

Hybrid Algorithms:

mldsa44_ecdsa_secp256r1_sha256
mldsa65_ecdsa_secp256r1_sha512
mldsa65_ecdsa_secp384r1_sha512
mldsa87_ecdsa_secp384r1_sha512
mldsa44_ed25519_sha512
mldsa65_ed25519_sha512
mldsa87_ed448_shake256
mldsa44_rsa2048_pkcs15_sha256
mldsa65_rsa3072_pkcs15_sha512
mldsa65_rsa4096_pkcs15_sha512
mldsa44_rsa2048_pss_sha256
mldsa65_rsa3072_pss_sha512
mldsa87_rsa3072_pss_sha512
mldsa65_rsa4096_pss_sha512
mldsa87_rsa4096_pss_sha512

HYBRID DIGITAL SIGNATURE MECHANISMS

IKEv2 - **Post-Quantum Traditional (PQ/T) Hybrid PKI Authentication in the Internet Key Exchange Version 2 (IKEv2)**

Status: **Individual Draft**

- Two proposals
  - Composite ML-DSA from PKIX/X.509 document
  - Two Certificates
    - Individual private keys sign as if they were a private key from the Composite ML-DSA PKIX/X.509 document

Notes

REMINDER: Individual Drafts have no standing at IETF

Hybrid Algorithms:

See Composite ML-DSA for use in X.509 Public Key Infrastructure

HYBRID DIGITAL SIGNATURE MECHANISMS

IKEv2 - Multi-Authentication in IKEv2 with Post-quantum Security

Status: **Individual Draft**

- Communicates authentication policy and performs multiple authentications

Notes

REMINDER: Individual Drafts have no standing at IETF

Hybrid Algorithms:

Doesn’t specify algorithms, leaves it up to policy.

HYBRID DIGITAL SIGNATURE MECHANISMS

CMS - Composite ML-DSA for use in Cryptographic Message Syntax (CMS)

Status: **IESG Review**

- Takes all algorithms from Composite ML-DSA for use in X.509 Public Key Infrastructure and applies them to CMS

Hybrid Algorithms

MLDSA44-RSA2048-PSS-SHA256
MLDSA44-RSA2048-PKCS15-SHA256
MLDSA44-Ed25519-SHA512
MLDSA44-ECDSA-P256-SHA256
MLDSA65-RSA3072-PSS-SHA512
MLDSA65-RSA3072-PKCS15-SHA512
MLDSA65-RSA4096-PSS-SHA512
MLDSA65-RSA4096-PKCS15-SHA512
MLDSA65-ECDSA-P256-SHA512
MLDSA65-ECDSA-P384-SHA512
MLDSA65-ECDSA-brainpoolP256r1-SHA512
MLDSA65-Ed25519-SHA512
MLDSA87-ECDSA-P384-SHA512
MLDSA87-ECDSA-brainpoolP384r1-SHA512
MLDSA87-Ed448-SHAKE256
MLDSA87-RSA3072-PSS-SHA512
MLDSA87-RSA4096-PSS-SHA512
MLDSA87-ECDSA-P521-SHA512

HYBRID KEY AGREEMENT MECHANISMS

S/MIME

S/MIME uses CMS

- CMS has ongoing hybrid digital signatures work
- No ongoing work to make hybrid digital signatures mandatory in S/MIME

HYBRID KEY AGREEMENT MECHANISMS

SSH - Composite ML-DSA Signatures for SSH

Status: **Individual Draft**

- Takes a few algorithms from Composite ML-DSA for use in X.509 Public Key Infrastructure and applies them to SSH.

Notes

REMINDER: Individual Drafts have no standing at IETF

Hybrid Algorithms:

ssh-mldsa44-es256 (sha256 prehash)

ssh-mldsa65-es256 (sha512 prehash)

ssh-mldsa87-es384 (sha512 prehash)

ssh-mldsa44-ed25519 (sha512 prehash)

ssh-mldsa65-ed25519 (sha512 prehash)

ssh-mldsa87-ed448 (shake256 prehash)

HYBRID KEY AGREEMENT MECHANISMS

SSH - Hybrid Ed25519 with ML-DSA-65 for Secure Shell (SSH)

Status: **Individual Draft**

- Combines Ed25519 and ML-DSA-65 in a way that is intended to be SUF-secure

Notes

REMINDER: Individual Drafts have no standing at IETF

Hybrid Algorithms:

ssh-ed25519-ml-dsa-65 (sha3-256 prehash)

HYBRID DIGITAL SIGNATURE MECHANISMS

OpenPGP - Post-Quantum Cryptography in OpenPGP

Status: **RFC Editor**

- Each component signs the OpenPGP dataDigest, and the signatures are concatenated
- Unlike Composite ML-DSA from LAMPS, this form of composite does not achieve any non-separability
- Must use a hash algorithm of at least 256 bits)

Hybrid Algorithms

ML-DSA-65+Ed25519

ML-DSA-87+Ed448

HYBRID DIGITAL SIGNATURE MECHANISMS

COSE & JOSE - PQ/T Hybrid Composite Signatures for JOSE and COSE

Status: **WG Draft**

- Takes a few algorithms from Composite ML-DSA for use in X.509 Public Key Infrastructure and applies them to COSE and JOSE.

Hybrid Algorithms

ML-DSA-44-ES256 (sha256 prehash)

ML-DSA-65-ES256 (sha512 prehash)

ML-DSA-87-ES384 (sha512 prehash)

ML-DSA-44-Ed25519 (sha512 prehash)

ML-DSA-65-Ed25519 (sha512 prehash)

ML-DSA-87-Ed448 (shake256 prehash)

HYBRID DIGITAL SIGNATURE MECHANISMS

MLS - ML-KEM and Hybrid Cipher Suites for Messaging Layer Security

Status: **WG Draft**

MLS cipher suites offering non-hybrid PQ signatures could be hybridized with cipher suites offering non-hybrid signatures using Amortized PQ MLS Combiner.

Hybrid Algorithms

MLS_192_MLKEM768_AES256GCM_SHA384_MLDSA65

MLS_256_MLKEM1024_AES256GCM_SHA384_MLDSA87

SUMMARY

Protocol	Title	Document	Status
PKIX / X.509	Composite ML-DSA for use in X.509 Public Key Infrastructure	draft-ietf-lamps-pq-composite-sigs	RFC Editor
TLS	Use of Composite ML-DSA in TLS 1.3	draft-reddy-tls-composite-mldsa	Individual Draft
IKEv2	Post-Quantum Traditional (PQ/T) Hybrid PKI Authentication in the Internet Key Exchange Version 2 (IKEv2)	draft-hu-ipsecme-pqt-hybrid-auth	Individual Draft
IKEv2	Multi-Authentication in IKEv2 with Post-quantum Security	draft-wang-ipsecme-multi-auth-ikev2-pq	Individual Draft
CMS	Composite ML-DSA for use in Cryptographic Message Syntax (CMS)	draft-ietf-lamps-cms-composite-sigs	IESG Review

SUMMARY

Protocol	Title	Document	Status
SSH	Composite ML-DSA Signatures for SSH	draft-sun-ssh-composite-sigs	Individual Draft
SSH	Hybrid Ed25519 with ML-DSA-65 for Secure Shell (SSH)	draft-josefsson-ssh-ed25519mldsa65	Individual Draft
OpenPGP	Post-Quantum Cryptography in OpenPGP	draft-ietf-openpgp-pqc	RFC Editor
COSE & JOSE	PQ/T Hybrid Composite Signatures for JOSE and COSE	draft-ietf-jose-pq-composite-sigs	WG Draft
MLS	ML-KEM and Hybrid Cipher Suites for Messaging Layer Security	draft-ietf-mls-pq-ciphersuites	WG Draft

Section 7

Hybrid StandardiSation Status Summary

HYBRID KEY AGREEMENT STANDARDISATION STATUS

Protocol	Framework	ML-KEM	FrodoKEM	ECDH	BP-ECDH	RSA-OAEP	X-ECDH
[CFRG]	WG	WG		WG			WG
HPKE		WG		WG			WG
PKIX/X.509		IESG		WG	WG	WG	WG
TLS	RFC Ed	RFC Ed		RFC Ed			RFC Ed
IKEv2	RFC	IESG	WG	RFC	RFC		RFC
CMS		IESG		IESG	IESG	IESG	IESG
S/MIME							
SSH		RFC Ed		RFC Ed			RFC Ed
OpenPGP							WG
COSE	IESG	Individual		Individual			Individual
JOSE	IESG	Individual		Individual			Individual
MLS		WG		WG			WG
EAP-AKA’		WG		WG			WG

HYBRID DIGITAL SIGNATURE STANDARDISATION STATUS

Protocol	Framework	ML-DSA	RSA-PSS	RSA-PKCS1.5	ECDSA	BP-ECDSA	EdDSA
PKIX/X.509		RFC Ed	RFC Ed	RFC Ed	RFC Ed	RFC Ed	RFC Ed
TLS		Individual	Individual	Individual	Individual		Individual
IKEv2	Individual						
CMS		IESG	IESG	IESG	IESG	IESG	IESG
S/MIME							
SSH		Individual			Individual	Individual	
OpenPGP		RFC Ed					RFC Ed
COSE		Individual			Individual	Individual	
JOSE		Individual			Individual	Individual	
MLS	WG						

ACM ALIGNMENT

Subjective ACM Alignment (Indicative)
Good – Exactly follows CatKDF/CasKDF Combiners; component algorithms align with at least at one option at each of NIST security levels 3 & 5
Maybe Okay – Combiners are like CatKDF/CasKDF, but parameters are rearranged; component algorithms align with option(s) at only one of NIST security levels 3 or 5
Bad – Major deviations from CatKDF/CasKDF Combiners, even though they may be technically secure; no option where all component algorithms are allowed by ACM

NOTE: ACM Alignment tables are indicative, not recommendations

HYBRID KEY AGREEMENT ACM ALIGNMENT

Combiner / Protocol	Combiner Alignment (Indicative)	Algorithm Alignment (Indicative)
C2PRI Combiner	No PQ public info included, with valid security proof; no context formatting function; no KDF alignment	N/A
PKIX/X.509	See C2PRI Combiner	SHA3-256 used as KDF
TLS	Conceptually like CatKDF, rearranged	Good
TLS ECH	See C2PRI Combiner	SHA3-256 used as KDF
IKEv2	No PQ public info included, no valid security proof; variable length nonces; KDF parameter differences	Good
CMS	See C2PRI Combiner	SHA3-256 used as KDF
SSH	Conceptually like CatKDF, rearranged	SHA256 and SHA384 used as KDF
OpenPGP	See C2PRI Combiner	None
COSE/JOSE	See C2PRI Combiner	SHA3-256 used as KDF
MLS	See C2PRI Combiner	SHA3-256 used as KDF
EAP-AKA’	See C2PRI Combiner	Unaligned KDFs

HYBRID DIGITAL SIGNATURES ACM ALIGNMENT

Combiner / Protocol	Algorithm Alignment (Indicative)
PKIX/X.509	Good
TLS	Good
IKEv2	Good
CMS	Good
SSH	Good
OpenPGP	None
COSE/JOSE	Good
MLS	Hybrid digital signatures is achieved by running parallel traditional and PQ sessions

NOTE: ACM Alignment tables are indicative, not recommendations

HYBRID STANDARDISATION STATUS SUMMARY

- IETF has significant ongoing work for hybridized key agreement and digital signatures.
- Key agreement is further ahead, with some RFCs published, a few nearing publication and many active working group drafts
- Digital signatures is further behind. Some drafts are nearing publication, but many working groups don't have any adopted work, just individual drafts without IETF standing.
- Apart from OpenPGP, all protocols have options which include ACM algorithms.
- However, many protocols exclusively use hybrid key agreements which don’t align with CatKDF/CasKDF and which don’t use an ACM KDF.
- Most work has focused on ML-KEM and ML-DSA, there is only one working group looking at FrodoKEM.
- Alignment with TS 103 744 is weak, although the hybrids being standardized do have their own security proofs.

enisa
EUROPEAN UNION AGENCY FOR CYBERSECURITY

Thank you for your attention

 +30 28 14 40 9711

 certification@enisa.europa.eu

 certification.enisa.europa.eu

ENISA
European Union Agency
for Cybersecunty

Athens Office
Agamemnonos 14
Chalandri 15231, Attiki, Greece

Brussels Office
Rue de la Loi 107
1049 Brussels, Belgium