

# ML-KEM Post-Quantum Key Agreement for TLS 1.3

draft-ietf-tls-mlkem-07+

<https://datatracker.ietf.org/doc/draft-ietf-tls-mlkem/>

<https://github.com/tlswg/draft-ietf-tls-mlkem>

# A pure-PQ ciphersuite for TLS 1.3

- No purely post-quantum ciphersuites
- Fills in the other side of [draft-ietf-tls-hybrid-design](#)
- Needed because there are no documents that describe KEM-only key agreement in TLS
- If PQ-only works for your applications, clean key agreement, no hybrid duplicate shares or mixing and matching logic
- ML-KEM-1024 supports users who need to comply with the CNSA 2.0 draft
- I want to be able to do it 🐎

# Changes out of the Second WGLC:

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).

base: draft-ietf-tls-mikem-05 ← ... compare: main

7 commits 1 file changed 1 contributor

Commits on Feb 12, 2026

- Synthesize feedback from the last call**  
dconnolly committed last month ✓ 202f034 <>

Commits on Feb 13, 2026

- Tweak motivation and description of hybrid**  
dconnolly committed last month ✓ b38a6bf <>
- Add cites for regs/policies for ml-kem only; tidy language around hybrid**  
dconnolly committed last month ✓ dfd7d67 <>
- general 'requirements'**  
dconnolly committed last month ✓ 17e815c <>

Commits on Feb 24, 2026

- Tidy per ekkr's comments, add a bunch of citations to existing analyses**  
dconnolly committed 3 weeks ago ✓ 237c19c <>

Commits on Feb 28, 2026

- more hybrid vs non-hybrid language**  
dconnolly committed 2 weeks ago ✓ fab6477 <>

Commits on Mar 3, 2026

- Clean up mentions of reuse, add links**  
dconnolly committed 2 weeks ago ✓ ab5f2bd <>

Showing 1 changed file with 124 additions and 102 deletions. Split Unified

## Updates since draft-ietf-tls-mlkem-05

- Added multiple references
- Reworked # Motivation several times
- Removed structures that were duplicate with 8446(bis)
- Put ‘Implementations MUST NOT reuse randomness..ciphertxts’ in the body, not #Security Considerations
- Added language about why hybrid is preferred by some
- Added multiple references about existing security analysis of KEMs for key agreement in TLS 1.3

# Should we just remove # Motivation?

✓ # Introduction

✓ ## Motivation

FIPS 203 (ML-KEM) `{{FIPS203}}` is a FIPS standard for post-quantum `{{RFC9794}}` key establishment via a lattice-based key encapsulation mechanism (KEM). This document defines key establishment options for TLS 1.3 that use solely post-quantum algorithms, without a hybrid construction that also includes a traditional cryptographic algorithm. Use cases include requirements `{{ITSP.40.111}}``{{CNSSP15}}``{{CNSAFAQ}}` that require standalone post-quantum key establishment, targeting smaller key sizes or less computation, and simplicity.

✓ # Conventions and Definitions

# Does this override 8446bis's 'SHOULD'?

Implementations **MUST NOT** reuse randomness in the generation of ML-KEM ciphertexts– it follows that ML-KEM ciphertexts also **MUST NOT** be reused.

# Current Security Considerations (1/3)

```
# Security Considerations {#security-considerations}
```

This document defines standalone ML-KEM key establishment for TLS 1.3. Hybrid key establishment mechanisms, which support combining a post-quantum algorithm with a traditional algorithm such as ECDH, are supported generically via `{{HYBRID}}` with some concrete definitions in `{{ECDHE-MLKEM}}`. Hybrid mechanisms provide security as long as at least one of the component algorithms remains unbroken, such as combining quantum-resistant and traditional cryptographic assumptions. Standalone ML-KEM relies on lattice-based and hash function cryptographic assumptions for its security. Proponents of hybrid PQ/T key establishment generally consider it a conservative approach to deployment of newer post-quantum schemes alongside older traditional schemes, retaining at least the security currently offered by traditional algorithms.

## Current Security Considerations (2/3)

The main security property for KEMs is indistinguishability under adaptive chosen ciphertext attack (IND-CCA), which means that shared secret values should be indistinguishable from random strings even given the ability to have other arbitrary ciphertexts decapsulated. IND-CCA corresponds to security against an active attacker, and the public encapsulation key / secret decapsulation key pair can be treated as a long-term key or reused in generic usage. ML-KEM satisfies IND-CCA security in the random oracle model  $\{\{KYBERV\}\}$  via a variant of the Fujisaki-Okamoto (FO) transform  $\{\{FO\}\}\{\{HHK\}\}$ . Use of KEMs for key agreement in TLS 1.3 has been analyzed and discussed in multiple settings and security models  $\{\{DOWLING\}\}\{\{KEMTLS\}\}\{\{HV22\}\}\{\{CHSW22\}\}\{\{CZCJWH25\}\}\{\{ZJZ24\}\}$ : ML-KEM's IND-CCA security exceeds the requirements for ephemeral key establishment and secure in case of reuse  $\{\{GHS25\}\}\{\{RFC8446bis\}\}$ .

## Current Security Considerations (3/3)

{{NIST-SP-800-227}} includes guidelines and requirements for implementations on using KEMs securely. Implementers are encouraged to use implementations resistant to side-channel attacks, especially those that can be applied by remote attackers.

TLS 1.3's key schedule commits to the ML-KEM encapsulation key and the ciphertext as the `key\_exchange` field of the `key\_share` extension is populated with those values, which are included as part of the handshake messages. This provides resilience against re-encapsulation attacks against KEMs used for key establishment {{CDM23}}.

# Open Issues

- Fix ref to latest CNSSP 15 (cert issue)

# Timeline

- Resolve last issue
- ???

# ML-KEM Post-Quantum Key Agreement for TLS 1.3

draft-ietf-tls-mlkem-07+

<https://datatracker.ietf.org/doc/draft-ietf-tls-mlkem/>

<https://github.com/tlswg/draft-ietf-tls-mlkem>