Transport Layer Security
Internet-Draft
Intended status: Informational
Expires: 16 November 2026

M. U. Sardar
TU Dresden, Germany

15 May 2026

Risks of Standalone ML-KEM in TLS 1.3
draft-usama-tls-risks-of-mlkem-00

Abstract

We attest that standalone ML-KEM in TLS 1.3 breaks the existing formal proofs of TLS in state-of-the-art symbolic security analysis tool, ProVerif. We also attest that from a formal analysis perspective, this is a much bigger change than RFC8773bis, which indeed went for FATT review (cf. [TLS-FATT]). We, therefore, kindly ask the chairs to initiate the FATT review of standalone ML-KEM in TLS.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://muhammad-usama-sardar.github.io/risks-of-mlkem/draft-usama-tls-risks-of-mlkem.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-usama-tls-risks-of-mlkem/.

Discussion of this document takes place on the Transport Layer Security Working Group mailing list (mailto:tls@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/tls/. Subscribe at https://www.ietf.org/mailman/listinfo/tls/.

Source for this draft and an issue tracker can be found at https://github.com/muhammad-usama-sardar/risks-of-mlkem.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Readers are assumed to be familiar with [NistFips203], [I-D.ietf-tls-rfc8446bis], and [I-D.ietf-tls-mlkem].

We assert that the security considerations of [I-D.ietf-tls-mlkem] are insufficient. We believe that symbolic and computational analysis of ML-KEM in the context of TLS is helpful here. We request that if the author has done any formal analysis, it would be very helpful to present the current state of formal analysis in the next meeting for discussion.

1.1. Motivation

The draft aims to formally study the security of standalone ML-KEM in TLS 1.3 [I-D.ietf-tls-mlkem].

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.

* Symbolic analysis: see SoK (https://eprint.iacr.org/2019/1393.pdf)

* Computational analysis: see SoK (https://eprint.iacr.org/2019/1393.pdf)

2.1. Where ProVerif Proofs Break

While ML-KEM [I-D.ietf-tls-mlkem] looks like just a "trivial" addition, it makes changes as deep as the key schedule of TLS. It essentially replaces the _key exchange_ by _key encapsulation_. While the former is symmetric, the latter is asymmetric. This symmetry is in terms of exchange of roles, and that the order does not matter. The proof in ProVerif is, therefore, utilizes this symmetry for the commutativity of the components g^x and g^y, where g^x and g^y represent the public keys of the endpoints. In ProVerif syntax: (see details here (https://github.com/CCC-Attestation/formal-spec-id-crisis/blob/6c3d17a428198aa058f805d16fe6baef7894028f/TLS-a/fix/tls-lib-simple.pvl#L38-L41))

```
fun dh_ideal(element,bitstring):element.
equation forall x:bitstring, y:bitstring;
         dh_ideal(dh_ideal(G,x),y) =
         dh_ideal(dh_ideal(G,y),x).
```

Key encapsulation does not enjoy this commutativity property, or even an analogous symmetry argument. There is essentially only one endpoint (say client) which generates the key pair (dk,ek) where dk represents the secret decapsulation key and ek represents the public

encapsulation key. As opposed to both endpoints sending their public keys g^x and g^y in the key exchange, only one of the endpoints (client in above example) sends the public encapsulation key and peer sends a ciphertext. This asymmetry breaks the existing proofs of TLS 1.3 in ProVerif and requires a new proof.

2.2. Current Status and Next Steps

[I-D.ietf-tls-mlkem] had an opposition of several (ca. 25 in our understanding) WG participants -- even more than the supporters (ca. 21 in our understanding) -- in the last WGLC. We see 2 possible options:

* Continue tabletop discussions on subjective calculation of risks, costs, tradeoffs, etc., and keep burning WG energy.

* Do some technical analysis using formal methods (symbolic and computational) to get a confirmation on the security of ML-KEM in the context of TLS and offer a statement for security considerations, and move on to more critical works like hybrid authentication.

We believe the former cannot resolve the dispute. We believe the latter _may_ help.

We believe the security considerations of {{I-D.ietf-tls-mlkem}} are insufficient. We also believe FATT review could have significantly improved it, including but not limited to the preference of hybrids, and potential issues regarding KEM binding in TLS.
We have provided significant feedback during the two WGLCs. However, almost none of that is actually reflected in the updated editor’s version.

2.2.1. "Cost"

"Cost" has been presented on the list as the motivation for ML-KEM but no reference has yet been presented. We believe costs will depend on several factors -- including but not limited to implementation details and deployment scenario -- and it is quite subjective.

There seems to be a need for a thorough study to understand the "cost." We invite the WG participants to perform this analysis and share the results with the WG.

2.3. ML-KEM: FATT Review

We have formally requested the chairs to initiate the FATT process for [I-D.ietf-tls-mlkem]. See this (https://mailarchive.ietf.org/arch/msg/tls/rClgrWm2hnhESXHx56U8InbwQQs/) and this (https://mailarchive.ietf.org/arch/msg/tls/7lj6fYAweMBwNMxFerNl7xhY0pk/).

2.3.1. Expected Learning

We believe formal methods can provide additional value for security considerations of this draft in order to maintain the high cryptographic assurance of TLS. Since we have no guarantee on whether ECDHE will break before ML-KEM, it seems appropriate to do thorough cryptographic analysis. We believe the Harvest Now, Decrypt Later (HNDL) attack applies equally well to standalone ML-KEM. Adversary can record all traffic and decrypt it when ML-KEM is broken (or probably it is already broken; who knows?)

* As an example, it can help justify design choices, such as the preference for hybrids. It can help identify ways in which ML-KEM can break. It can also help identify all the assumptions under which the properties hold.

* As a relevant data point in the context of standardization, LAKE WG has done formal analysis for EDHOC-PSK with KEM (ref (https://mailarchive.ietf.org/arch/msg/lake/2XGOI9OCwylJUfSCasvvwM2FXmw/)).

* _Computational_ analysis (cf. SoK (https://eprint.iacr.org/2019/1393.pdf))-- using tools such as CryptoVerif -- seems like a reasonable approach to ensure security of ML-KEM in TLS, such as binding.

2.3.2. Formal Analysis (Work-in-progress)

We have presented observation from our ongoing symbolic security analysis (cf. limitations in Section 3) using ProVerif on the mailing list.

We argue that in general:

1. Migration from ECDHE to hybrid is security improvement.

2. Migration from hybrid to standalone ML-KEM is security regression.

2.3.2.1. Hybrid PQ/T

More formally, the property hybrid PQ/T should provide is:

Hybrid PQ/T is secure unless both ECDHE and ML-KEM are broken.

Hybrid preserves ECDHE, and adds ML-KEM as an additional factor. So as long as one of them is not broken, the system is secure. In particular, even if ML-KEM is completely broken, the system retains the security level of ECDHE.

2.3.2.2. Standalone PQ

On the other hand, the formal property standalone PQ provides is:

Standalone PQ is secure unless ML-KEM is broken.

If ML-KEM is broken, the whole system is broken.

2.3.2.3. Comparison

Leak out the ECDHE key from hybrid PQ/T and you get a standalone ML-KEM. Clearly, hybrid is in general more secure, unless ECDHE is fully broken, in which case it still falls equivalent to standalone ML-KEM, or in the hypothetical scenario that there is an implementation bug in the ECDHE part which is triggered only in composition.

3. Security Considerations

The whole document is about improving security considerations.

Like all security proofs, formal analysis is only as strong as its assumptions and model. The scope is typically limited, and the model does not necessarily capture real-world deployment complexity, implementation details, operational constraints, or misuse scenarios. Formal methods should be used as complementary and not as subtitute of other analysis methods.

4. IANA Considerations

This document has no IANA actions.

5. References

5.1. Normative References

[I-D.ietf-tls-mlkem]
Connolly, D., "ML-KEM Post-Quantum Key Agreement for TLS 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-mlkem-07, 12 February 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-mlkem-07>.

[I-D.ietf-tls-rfc8446bis]
Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-rfc8446bis-14, 13 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-rfc8446bis-14>.

[NistFips203]
"Module-lattice-based key-encapsulation mechanism standard", National Institute of Standards and Technology (U.S.), DOI 10.6028/nist.fips.203, August 2024, <https://doi.org/10.6028/nist.fips.203>.

[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>.

[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>.

[TLS-FATT] IETF TLS WG, "TLS FATT Process", June 2025, <https://github.com/tlswg/tls-fatt>.

5.2. Informative References

[I-D.usama-tls-fatt-extension]
Sardar, M. U., "Extensions to TLS FATT Process", Work in Progress, Internet-Draft, draft-usama-tls-fatt-extension-07, 2 May 2026, <https://datatracker.ietf.org/doc/html/draft-usama-tls-fatt-extension-07>.

Acknowledgments

The research work is funded by German Research Foundation ("Deutsche Forschungsgemeinschaft.")

History

-00

* On popular demand, moved from [I-D.usama-tls-fatt-extension] to an independent I-D

* Major change: added Section 2.1

* Some minor clarifications

Author's Address

Muhammad Usama Sardar
TU Dresden, Germany
Email: muhammad_usama.sardar@tu-dresden.de