Transport Layer Security
Internet-Draft
Intended status: Informational
Expires: 6 December 2026

M. U. Sardar
TU Dresden, Germany

4 June 2026

Potential Risks of Standalone ML-KEM in TLS 1.3
draft-usama-tls-risks-of-mlkem-02

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 believe this requires a new symbolic proof in ProVerif. To help inform the analysis, we share our understanding of *exactly* where the ProVerif proofs break, namely transition from symmetric DHKE to asymmetric KEM. More specifically, our understanding is that the existing proofs of TLS in ProVerif are based on commutativity property, whereas commutativity does not apply to standalone ML-KEM in TLS.

In general, we see no reason to believe that hybrid key exchanges are not _at least_ as strong as the stronger of the two components. We invite collaborations or independent analysis to extend the ProVerif models to perform such analysis and offer a statement for security considerations of [I-D.ietf-tls-mlkem]. In our understanding, a couple of WG participants have already started formal analysis in 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, formally request the chairs to initiate the FATT review of standalone ML-KEM in TLS.

This draft also offers some preliminary discussion to help the developers and policy makers make informed choices. Finally, the draft also aims to reduce the endless repitition of arguments from both sides presented on several lists by documenting these arguments so they can simply be referred to. We sincerely believe this will help to focus the discussion on technical matters, such as system model, threat model, security properties, and deployments.

We acknowledge several IETF participants who have contributed to this draft with their insights. This draft captures what _we_ understand them to be saying.

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

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 6 December 2026.

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]. Please note that the draft has currently several hyperlinks.

We assert that the security considerations of [I-D.ietf-tls-mlkem] are insufficient. We believe that consistent with [TLS-FATT] process, _symbolic_ and _computational_ analysis (to be interpreted as in SoK (https://eprint.iacr.org/2019/1393.pdf)) of *integration* of standalone ML-KEM in the context of TLS is helpful here.

We believe that the focus of symbolic analysis ought to be on the *integration* details (transcript binding, key schedule, agreement) for standalone ML-KEM in the context of TLS, rather than the *primitive* itself.

We believe that if any WG participant has done any formal analysis, it would be very helpful to share the results with the WG for discussion.

Literature review is currently ongoing. Some existing computational analysis for standalone ML-KEM in TLS include this (https://eprint.iacr.org/2021/844) and this (https://eprint.iacr.org/2024/1360). Both are based on pen-and-paper (computational) proofs. At the symbolic level, some analysis -- such as this (https://eprint.iacr.org/2022/1111.pdf) for KEMTLS in Tamarin -- exists. In our understanding, both client and server encapsulate, which may bring the symmetry.

1.1. Gap Analysis

We are currently not aware of any peer-reviewed work on *integration* of standalone ML-KEM in TLS based on ProVerif. Getting a confirmation on the symbolic level seems valuable.

Some WG participants seem to disagree with the statement that the hybrid key exchange in TLS is at least as good as standalone ML-KEM in TLS. We are not aware of any literature which claims that standalone ML-KEM in TLS is _better_ than hybrid key exchange in TLS. Getting a confirmation on these subtleties via formal analysis seems very useful for resolving this difference of opinions.

1.2. Motivation

[rfc3552] requires to document the risks in the security considerations. To support those requirements for [I-D.ietf-tls-mlkem], this draft aims to formally study the security of standalone ML-KEM in TLS 1.3. This is because of the following reasons.

In the last WGLC, [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). We see 2 possible options:

* Continue tabletop discussions on *subjective* estimation of urgency, risks, costs, tradeoffs, etc., and keep burning WG energy by endless repitition.

* Do some technical analysis using (_symbolic_ and _computational_) formal methods to get a confirmation on the security of *integration* of standalone ML-KEM in the context of TLS and offer a statement for security considerations.

We believe the former cannot resolve the dispute. We sincerely *hope* the latter will 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 hybrid key exchanges, and potential issues regarding KEM binding in TLS. WG participants have provided significant feedback during the two WGLCs. However, not much of that is actually reflected in the updated editor’s version at the time of writing.

1.2.1. 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. The opinions of WG participants here vary from "ML-KEM is secure" to "ML-KEM is probably already secrectly broken." Formal methods can operate under the assumption that ML-KEM is secure, and focus on the *integration* of ML-KEM in TLS under this assumption.

* As an example, formal methods can help justify design choices, such as the preference for hybrid key exchanges. 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 shared secret ss to the TLS transcript hash.

1.2.2. Minimum Viable Modeling

Based on the discussion on list, simply replacing ideal DHKE by ideal ML-KEM in the formal model is not very useful. We ought to focus on the more security-critical questions about *integration* of ML-KEM in TLS. We present a few high-level observations to consider for security considerations of [I-D.ietf-tls-mlkem]:

* The model ought to consider that any agent could have initiated the TLS, rather than assigning the agents with static roles of client and server in the model. When agents are assigned non-static roles, it would be interesting to see whether the asymmetry issue becomes visible in some property. We consider it very critical for security considerations of [I-D.ietf-tls-mlkem] and this is the key point of this draft.

* Different failure modes proposed on list can be modeled.

* A large part of the problem is the careful investigation of what to model, under what threat model, under what system model, under what implementation scenarios etc. We believe some of this is important for security considerations of [I-D.ietf-tls-mlkem].

* It will be interesting to see some analysis about any subtle cases where hybrid key exchange in TLS is _not_ at least as good as standalone ML-KEM in TLS. Our understanding is that some participants would like to see some statement on the comparison since hybrid key exchange is the de facto industry standard.

* We believe brainstorming about some robustness (vs. security) properties would also be useful. Even if the security properties hold, does standalone ML-KEM make side-channel leakage easier? This might be a valuable consideration for the implementers.

* Analysis may be helpful to ensure that the changes -- such as the removal of hash function (cf. Appendix C.1, bullet 3 in [NistFips203]) -- from Kyber to ML-KEM preserve the security proofs of Kyber.

We invite collaborations or independent analysis to extend the ProVerif models to perform this analysis. Any analysis on these or related security and robustness matters is very welcome.

1.2.3. Previous Formal Requests for 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/), this (https://mailarchive.ietf.org/arch/msg/tls/7lj6fYAweMBwNMxFerNl7xhY0pk/), and this (https://mailarchive.ietf.org/arch/msg/tls/2LukH1riSE5PQPpMVlVGygp4lpg/).

1.2.4. FATT Review is Harmless

For those who are worried, please note the legitimate outcome _nothing required_ in [TLS-FATT].

Recommendations output from the FATT for a particular document may range from ’nothing required’ to ’pen-and-paper proof can be updated’ to ’a formal methods model using a specific tool ought to be done’ - the ’formal’ is not limited to formal methods but to formal security modeling generally.

Please also note [TLS-FATT]:

The output may say that additional analysis is not warranted or it may indicate what type of analysis should be done.

Moreover, WG retains its authority [TLS-FATT]:

The working group is not obligated to follow the FATT recommendation.

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)
* Standalone ML-KEM refers to [I-D.ietf-tls-mlkem].
* Hybrid key exchange refers to [I-D.ietf-tls-ecdhe-mlkem] and [I-D.ietf-tls-hybrid-design].

We believe that symbolic and computational models are complementary and not a substitute of each other.

3. Where ProVerif Proofs Break

We attest that:

1. existing proofs of TLS in ProVerif are based on commutativity
2. commutativity does not apply to standalone ML-KEM in TLS
Hence, a new proof is required.
This entails updating ProVerif models, e.g., modeling KEMs.

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 assigned to the agents, and that the order does not matter. The existing proofs in ProVerif, therefore, utilize this symmetry for the commutativity of the key shares g^x and g^y, where g^x and g^y represent the public key shares of the endpoints. In ProVerif syntax: (see original source here (https://github.com/Inria-Prosecco/reftls/blob/634f7da5940f8d1f09cfcd56280b4ef3b533df6b/pv/tls-lib-draft20.pvl#L45-L48) and re-used 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 key shares g^x and g^y in a traditional key exchange (DHKE), a KEM creates a roles-asymmetry where only one of the endpoints (client in above example) sends the public encapsulation key ek and the peer (server) sends a ciphertext ct. This asymmetry breaks the existing proofs of TLS 1.3 in ProVerif and requires a new proof.

Please note that breaking the existing ProVerif proof does not imply that the standalone ML-KEM proposal in TLS [I-D.ietf-tls-mlkem] is insecure. It just means that a new proof is required. We welcome

feedback and collaborations from the community on doing a thorough analysis in ProVerif -- such as in Section 1.2.2 -- while preserving the cryptographic soundness.

4. Justification based on FATT Process

Our formal request for FATT review is fully in conformance with the current [TLS-FATT] process, which explicitly states:

For example a proposal that modifies the TLS key schedule or the authentication process or any other part of the cryptographic protocol that has been formally modeled and analyzed in the past would likely result in asking the FATT, whereas a change such as modifying the SSLKEYLOG format would not.

As presented in Section 3, we attest that [I-D.ietf-tls-mlkem] modifies the:

* TLS key schedule
* cryptographic protocol such that commutativity property is no longer valid.

This breaks the following proofs in ProVerif:

* Bhargavan et al.’s model of draft 20 of TLS 1.3: [reftls] and [reftls-Repo] and all 5 _public_ forks as well as one nested fork:
  - arthuraa/reftls (https://github.com/arthuraa/reftls/blob/d6bc5dd8eb4373683cb1ce64845691954d0d7601/pv/tls-lib-draft20.pvl#L44-L47)
  - blipp/reftls (https://github.com/blipp/reftls/blob/5bc66d14d4accbff6edb0ae7a263df5ea880857d/pv/tls-lib-draft20.pvl#L44-L47)
  - chris-wood/reftls (https://github.com/chris-wood/reftls/blob/d6bc5dd8eb4373683cb1ce64845691954d0d7601/pv/tls-lib-draft20.pvl#L44-L47)
  - ekr/reftls (https://github.com/ekr/reftls/blob/5bc66d14d4accbff6edb0ae7a263df5ea880857d/pv/tls-lib-draft20.pvl#L44-L47)
    - ajayeeralla/reftls (https://github.com/ajayeeralla/reftls/blob/b97196fa0c3885da0fe0f412c9902e85a7f5323a/pv/tls-lib-draft20.pvl#L44-L47)

  - jhoyla/reftls (https://github.com/jhoyla/reftls/blob/d6bc5dd8eb4373683cb1ce64845691954d0d7601/pv/tls-lib-draft20.pvl#L44-L47)

* Our previous work extending the model of Bhargavan et al. to the current state of [I-D.ietf-tls-rfc8446bis] and integrating remote attestation: [ID-Crisis] and [ID-Crisis-Repo] (under Apache-2.0 License) and all 3 public forks:

  - jupenur/formal-spec-id-crisis (https://github.com/jupenur/formal-spec-id-crisis/blob/de2bdec9967bf535f648f0cc8e8d2d90a49104a4/TLS-a/fix/tls-lib-simple.pvl#L38-L41)

  - nathanaelritz/formal-spec-id-crisis (https://github.com/nathanaelritz/formal-spec-id-crisis/blob/a028cec823b7d9bf13dd5a1dd71ab14c75b1a83d/TLS-a/fix/tls-lib-simple.pvl#L38-L41)

  - telephonicrobotics/formal-id-crisis-spec (https://github.com/telephonicrobotics/formal-id-crisis-spec/blob/c1953127ce004e51b888250591ec9971ad50e98c/TLS-a/fix/tls-lib-simple.pvl#L38-L41) (owner of this repo is the same as the one before this and has indicated that there was no active _private_ development in this repo)

* A couple of our ongoing works which are not yet public

Note: Forks may or may not have substantive changes. It is hard for us to know and judge how much research and development effort someone did _privately_ and did not make it public and hence, we do not want to add our personal estimation of whether someone has substantially worked after forking. Readers are welcome to make their own opinions by exploring the repos or contact the respective repo owners for further details. The hyperlinks provide one instance of usage of equation in the main branch based on what is publicly available.

In our understanding, a couple of WG participants are already working on formal analysis of [I-D.ietf-tls-mlkem] analyzing the items mentioned in Section 1.2.2. A new section will be added when they will share their results of the items in Section 1.2.2.

4.1. Comparison with RFC8773bis

Please note that RFC8773bis is a much smaller change: it’s pretty much standard TLS and still went for FATT review. Based on that, we see no reason to believe that [I-D.ietf-tls-mlkem] -- with key schedule level changes -- should not be sent to FATT.

4.2. FATT Review for Hybrid Key Exchange?

Some participants have raised concern that the same issue _may_ apply to hybrid key exchange as well and one of the proposals is to block that draft. We *very strongly* oppose this proposal because of the following reasons:

4.2.1. Stage of Publication

[I-D.ietf-tls-hybrid-design] has IETF consensus and is in the publication queue. Given the consensus, we see absolutely no reason to block that. As we understand, FATT process was specifically designed to *resolve* concerns rather than *gatekeeping*.

In contrast, as mentioned in Section 1.2, standalone MLKEM [I-D.ietf-tls-mlkem] is still within the WG and has a very different profile with ca. 25 oppositions in our understanding in the last WGLC. FWIW, this is exactly what makes formal analysis potentially helpful to resolve the issue, build high confidence and offer a statement for security considerations after a careful formal analysis.

4.2.2. Technical Rationale

Technically, a proof of [I-D.ietf-tls-hybrid-design-09] is done in the computational model using CryptoVerif (cf. ref (https://bblanche.gitlabpages.inria.fr/publications/BlanchetJacommeCSF24.pdf)). As per list discussion, it appears that the proof applies to the latest version of the spec [I-D.ietf-tls-hybrid-design], as there seem to be no substantive changes from the perspective of formal proof.

Moreover, we believe that the two drafts [I-D.ietf-tls-hybrid-design] and [I-D.ietf-tls-mlkem] are incomparable on this specific point as hybrid key exchange still maintains the symmetry in the DHKE part. From formal (symbolic) analysis perspective, g^x and g^y are still sent in hybrid key exchange, g^xy is still computed and we believe the commutativity property is applicable for that part as-is. From formal (symbolic) analysis perspective, ML-KEM is complementary to that.

Specifically, from Section 4 of [I-D.ietf-tls-ecdhe-mlkem], for the symbolic analysis, X25519MLKEM768 in TLS may be viewed as:

client’s key_exchange value = ek || gx
server’s key_exchange value = ct || gy
shared secret = ss || gxy

4.2.3. Marginal Additional Effort for Hybrid Key Exchange

Once the formal analysis for standalone ML-KEM in TLS [I-D.ietf-tls-mlkem] is done, we expect that the additional effort for hybrid key exchange in TLS [I-D.ietf-tls-hybrid-design] will only be marginal, as the building blocks for DHKE already exist in ProVerif.

4.2.4. What if Issue is Found?

Should there be a groundbreaking discovery of an issue which applies also to [I-D.ietf-tls-hybrid-design], we are confident that the WG will find a way out, such as very quickly applying the fix proposed by the formal analysis, doing a very quick WGLC and IETF LC while requesting the AD and RFC Editor to keep the draft at its place in the publication queue.

5. Formal Analysis (Work-in-progress)

We have presented observation from our ongoing symbolic security analysis (cf. limitations in Section 7) using ProVerif on the mailing list.

For brevity, we omit other assumptions in the properties below and focus on the difference. This assumes hybrid constructor to be secure.

We believe that _in general_:

1. Migration from ECDHE to hybrid key exchange is security improvement.
2. Migration from hybrid key exchange to standalone ML-KEM is security regression.

5.1. Hybrid Key Exchange

More formally, the property hybrid key exchange _should_ provide is:

Security properties of TLS hold unless *both* ‘gxy‘ and ‘ss‘ are available to the adversary.

As presented in Section 4.2.2, hybrid key exchange preserves ECDHE component gxy, and concatenates ML-KEM component ss as an additional factor. So as long as _at least_ one of these two secrets is not available to the adversary, all security properties should hold. In particular, even if ML-KEM is completely broken, i.e., ss is available to the adversary, the protocol retains the security level of ECDHE.

5.2. Standalone ML-KEM

On the other hand, the formal property standalone ML-KEM can provide is:

Security properties of TLS hold unless ‘ss‘ is available to the adversary.

5.3. Comparison

Leaking out the ECDHE key from hybrid key exchange should downgrade the security to the level of a standalone ML-KEM. Therefore, hybrid key exchange is _in general_ more secure, unless:

* ECDHE is fully broken, in which case it still falls equivalent to standalone ML-KEM,

* in the _hypothetical_ scenario that there is an implementation bug in the ECDHE part which is triggered only in composition. We have not yet seen any concrete evidence of such a scenario on the list.

6. Issues That Formal Methods Probably Cannot Solve

The answers to the following issues are largely dependent on several factors, and the opinions vary largely.

It is necessary to mention that even several respectable cryptographers in the community are not aligned on the issue -- for example see the long bet (https://github.com/FiloSottile/ecc-vs-lattices-long-bet). Hence, our personal opinion is probably not that important. Probably the best we can do is to capture _our_ understanding of the views of WG participants.

Disclaimer: This is not meant to be an exhaustive list.
This is also not meant to pritoritize any concerns over others.
This is a sincere attempt to slowly capture the opinions
to avoid endless repetitions from both sides.
Many substantive concerns are missing.
We are slowly collecting the concerns, as time allows.
If your substantive concern is missing, it is unintentional.
Please simply submit a *precise* and *concise* PR.

6.1. Recommendation of Designers

The authors of Kyber/ML-KEM (see this (https://pq-crystals.org/kyber/index.shtml)) say:

For users who are interested in using Kyber, we recommend the following:

* Use Kyber in a so-called hybrid mode in combination with established "pre-quantum" security; for example in combination with elliptic-curve Diffie-Hellman.
[...]

A WG participant shares (https://mailarchive.ietf.org/arch/msg/tls/NnGrdavTY6KGTVQo46xaPbSHQzw/) that:

I recently asked one of the members of the CRYSTALS team whether this is still his view, and the response was: "Yes, of course."

6.2. Thorough Review

Please see a very thorough review here (https://mailarchive.ietf.org/arch/msg/tls/jlsYHENwqMv-4XPRvunqKsAL36k/), which is self-sufficient.

6.3. ’Significantly Harder’ Argument

Some participants believe in the ’significantly harder’ argument, which assumes independence of breakage of ML-KEM and traditionals:

If the probablity of one being broken over the next n years is p, and the probability of the other being broken over the next n years is q, then the probability of both being broken is pq.

Please see this (https://github.com/FiloSottile/ecc-vs-lattices-long-bet#2a-what-counts-as-a-break) for what "broken" may mean here modulo some exclusions (https://github.com/FiloSottile/ecc-vs-lattices-long-bet#5-exclusions).

Given the very different type of cryptographic constructions involved, independence might be a reasonable assumption. However, some participants disagree with 'significantly harder' argument with a reasonable counter-argument that in reality, cryptography is much more complicated than that (cf. this (https://mailarchive.ietf.org/arch/msg/tls/AK7QUiiGX3ynsOhXeUuwn_IY7ik/)):

Depending on the algorithms and the composition method, the probability can clearly be q, or smaller than pq.

In our understanding, most other counter-arguments seem to break the exclusions (https://github.com/FiloSottile/ecc-vs-lattices-long-bet#5-exclusions).

Please note that this argument is based on the security of _primitives_, rather than the _composition_ of primitives in protocols. Hence, formal methods probably have nothing to help here.

6.4. Urgency

It is unclear _whether_ and if applicable _when_ Cryptographically-Relevant Quantum Computer (CRQC) will eventually become practical. The opinions vary from never because of complicated physics (see this (https://eprint.iacr.org/2025/1237)) to be _prepared_ for it as early as 2029 (see Google 2029 (https://blog.google/innovation-and-ai/technology/safety-security/cryptography-migration-timeline/) and Cloudflare 2029 (https://blog.cloudflare.com/post-quantum-roadmap/)). Technically, please note that Google has not even released the *quantum circuit* underlying their recent claims -- apparently the reason for this urgency. So Google's claims may not yet be justified.

Moreover, in our understanding, these deadlines are for PQ-based protection in general regardless of hybrid key exchange or standalone KEMs in TLS. Since hybrid key exchange is wildly in use, these deadlines are mainly for quantum-safe authentication.

In any case, some participants see no reason to create panic for publication of [I-D.ietf-tls-mlkem] based on this because many implementations -- such as OpenSSL -- have already implemented standalone ML-KEM, and it is just a matter of enabling it. And frankly, nobody needs permission from the IETF to enable it.

6.5. "Cost"

"Cost" has been presented on the list as the motivation for standalone ML-KEM in TLS but we have not seen any supporting analysis. Our observation from Section 4 of [I-D.ietf-tls-ecdhe-mlkem] is that -- for example -- for X25519MLKEM768, the traditional part seems negligible compared to ML-KEM part in key_exchange:

Bytes in field	PQ part (ML-KEM)	Traditional part (X25519)
Client share	1184	32
Server share	1088	32

Table 1

We believe other "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 cost analysis and share the results with the WG.

6.6. Is Publication Necessary?

Code Points for ML-KEM have already been assigned. [I-D.barnes-tls-this-could-have-been-an-email] provides detailed rationale as to why publication of such documents and the debates around that may be unnecessary. In our understanding, [I-D.pwouters-crypto-current-practices] makes similar arguments.

6.7. Shiny New Crypto

ML-KEM is quite new in the IETF and even in the IRTF. Some WG participants have shown concern over premature publication of [I-D.ietf-tls-mlkem] until a detailed analysis has been done by CFRG.

CFRG is starting some efforts for analysis. The extended deadline for submission is 22.06. Please see the latest CFRG chairs email (https://mailarchive.ietf.org/arch/msg/cfrg/6K43Ycr062Ym1G0q4WHxZQ2HW8M/) for further details.

6.8. Formal Mapping of FIPS to IETF BCP14

As discussed on the TLS list, we are not aware of any formal mapping of the FIPS recommendations to the IETF BCP14 terminology, such as SHOULD vs. MUST. In general, we believe re-using FIPS recommendations is ambiguous for IETF readers.

6.9. Outstanding NIST Comments

Some participants believe that NIST has rushed through the process and not addressed all the comments that were submitted during the open review. Please see comments here (https://csrc.nist.gov/files/pubs/fips/203/ipd/docs/fips-203-initial-public-comments-2023.pdf).

6.10. Too Early

Some participants simply believe that publication of [I-D.ietf-tls-mlkem] and related discussions are just too early and unnecessary.

6.11. Patents

Some WG participants have raised some concerns related to patents. See some relevant patents here (https://datatracker.ietf.org/ipr/search/?submit=draft&id=draft-ietf-tls-mlkem).

7. 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.

8. IANA Considerations

This document has no IANA actions.

9. References

9.1. 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>.

9.2. Informative References

[I-D.barnes-tls-this-could-have-been-an-email]
Barnes, R., "Stop Doing Cryptographic Algorithm Drafts when Email to IANA is All You Need", Work in Progress, Internet-Draft, draft-barnes-tls-this-could-have-been-an-email-00, 23 February 2026, <https://datatracker.ietf.org/doc/html/draft-barnes-tls-this-could-have-been-an-email-00>.

[I-D.ietf-tls-ecdhe-mlkem]
Kwiatkowski, K., Kampanakis, P., Westerbaan, B., and D. Stebila, "Post-quantum hybrid ECDHE-MLKEM Key Agreement for TLSv1.3", Work in Progress, Internet-Draft, draft-ietf-tls-ecdhe-mlkem-05, 26 May 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-ecdhe-mlkem-05>.

[I-D.ietf-tls-hybrid-design]
Stebila, D., Fluhrer, S., and S. Gueron, "Hybrid key exchange in TLS 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-hybrid-design-16, 7 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-hybrid-design-16>.

[I-D.ietf-tls-hybrid-design-09]
Stebila, D., Fluhrer, S., and S. Gueron, "Hybrid key exchange in TLS 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-hybrid-design-09, 7 September 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-hybrid-design-09>.

[I-D.pwouters-crypto-current-practices]
Wouters, P., "Current practices for new cryptography at the IETF", Work in Progress, Internet-Draft, draft-pwouters-crypto-current-practices-00, 3 November 2024, <https://datatracker.ietf.org/doc/html/draft-pwouters-crypto-current-practices-00>.

[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>.

[ID-Crisis]
Sardar, M. U., Moustafa, M., and T. Aura, "Identity Crisis in Confidential Computing: Formal Analysis of Attested TLS", November 2025, <https://www.researchgate.net/publication/398839141_Identity_Crisis_in_Confidential_Computing_Formal_Analysis_of_Attested_TLS>.

[ID-Crisis-Repo]
Muhammad Usama Sardar, "Identity Crisis in Confidential Computing: Formal Analysis of Attested TLS Protocols", <https://github.com/CCC-Attestation/formal-spec-id-crisis>.

[reftls] Bhargavan, K., Blanchet, B., and N. Kobeissi, "Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate", IEEE, 2017 IEEE Symposium on Security and Privacy (SP) pp. 483-502, DOI 10.1109/sp.2017.26, May 2017, <https://doi.org/10.1109/sp.2017.26>.

[reftls-Repo] Bhargavan, K., Blanchet, B., and N. Kobeissi, "Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate", <https://github.com/Inria-Prosecco/reftls>.

[rfc3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <https://www.rfc-editor.org/rfc/rfc3552>.

Acknowledgments

We would like to thank Yaakov Stein, Ilari Liusvaara, John Preu Mattsson, Eric Rescorla, Brian E Carpenter, Nadim Kobeissi, and Tibor Jager for their valuable feedback and contributions.

Section 6 is largely based on the opinions of many IETF participants.

Text in Section 7 is based on the proposal by John Preu Mattsson.

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 3
* Some minor clarifications

-01

* Added justification based on FATT process: Section 4
* Reorganization, specially in motivation
* Added some common arguments: Section 6

* Comparison with hybrid key exchange Section 4.2

-02

* Added gap analysis Section 1.1

* What to model and analyze? Section 1.2.2

* Added FATT review is harmless Section 1.2.4

* Extended comparison with hybrid key exchange Section 4.2

* Opinion of designers Section 6.1

Author’s Address

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