Transport Layer Security
Internet-Draft
Intended status: Informational
Expires: 30 November 2026

M. U. Sardar
TU Dresden, Germany

29 May 2026

# Potential Risks of Standalone ML-KEM in TLS 1.3
## draft-usama-tls-risks-of-mlkem-01

## 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. In this draft, we show *exactly* where the ProVerif proofs break, namely transition from symmetric DHKE to asymmetric KEM. More specifically, the existing proofs of TLS in ProVerif are based on commutativity property, whereas commutativity does not apply to standalone ML-KEM in TLS.

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. A few WG participants have already volunteered to do formal analysis in ProVerif.

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.

## 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 consistent with [TLS-FATT] process, _symbolic_ and _computational_ analysis (to be interpreted as in SoK (https://eprint.iacr.org/2019/1393.pdf)) of standalone ML-KEM in the context of TLS is helpful here. We believe that if the author or any WG participant has done any formal analysis, it would be very helpful to present the current state of formal analysis in the next meeting for discussion.

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 proofs.

### 1.1. 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 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 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 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.

#### 1.1.1. 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 here vary from "ML-KEM is probably 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 hybrids. 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 to the TLS transcript hash.

#### 1.1.2. 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/) and this (https://mailarchive.ietf.org/arch/msg/tls/7lj6fYAweMBwNMxFerNl7xhY0pk/).

## 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)

## 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, 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, 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 necessarily mean that the ML-KEM proposal in TLS is insecure. It just means that a new proof is required. We welcome feedback from the community on how to fix the ProVerif proofs 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)

    o 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)

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

### 4.1. Hybrid ML-KEM?

Some participants have raised concern that the same issue _may_ apply to hybrid ML-KEM as well. 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 are no substantive changes from the perspective of formal proof.

Moreover, we believe that the two drafts are incomparable on this specific point as hybrid ML-KEM [I-D.ietf-tls-hybrid-design-09] still has some level of symmetry. From formal (symbolic) analysis perspective, g^x and g^y are still sent in hybrid ML-KEM, 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 may be viewed as:

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

## 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.

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.

### 5.1. 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 _at least_ 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.

### 5.2. 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.

### 5.3. 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.

## 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 repititions 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. Thorough Review

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

### 6.2. '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.

Given the very different type of cryptographic constructions involved, we believe independence might be a reasonable assumption.

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). Some participants disagree with 'significantly harder' argument, but in our understanding, the counter-arguments seem to break the 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.

### 6.3. 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 are not yet justified.

Moreover, in our understanding, these deadlines are for PQ-based protection in general regardless of hybrid KEMs or standalone KEMs in TLS. Since hybrid KEMs already exist, 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.

### 6.4. "Cost"

"Cost" has been presented on the list as the motivation for standalone ML-KEM in TLS but no supporting analysis has yet been presented. 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:

| Field | ML-KEM 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 this analysis and share the results with the WG.

### 6.5. 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.

### 6.6. 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.

### 6.7. 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.

### 6.8. 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).

### 6.9. Too Early

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

## 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, and Nadim Kobeissi 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 ML-KEM Section 4.1

## Author’s Address

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