TLS 1.3 纯 ML-KEM-768 1-RTT 的安全性分析

本报告的其他语言版本：English。

draft-ietf-tls-mlkem-11 · ProVerif+ Agent 报告

文稿	ML-KEM Post-Quantum Key Agreement for TLS 1.3 （协议定义）; The Transport Layer Security (TLS) Protocol Version 1.3 （辅助）
检查的性质	15 条：12 条成立，3 条不成立（3 个攻击）
发现	无

目录

1 协议和范围
2 发现
3 检查过的性质
4 本次分析的局限
附录 A：逐条性质的解读
附录 C：主张和证明器的答案
附录 D：分析是怎么跑的

1 协议和范围

这是 TLS 1.3，ML-KEM-768 是唯一的密钥建立组，使用服务器证书认证，完整的一轮往返握手；HelloRetryRequest、PSK、0-RTT 和客户端证书不在范围内。客户端生成 ML-KEM 密钥对，把封装密钥作为 key_share 发送；服务器向其封装，返回密文，导出共享秘密，并用证书、对 transcript 的 CertificateVerify 和 Finished 进行认证。客户端检查这三者，发送自己的 Finished，双方随后在由共享秘密派生的密钥下交换应用数据。

交换结束后，客户端知道自己正在与证书所命名的服务器通信，因为签名覆盖的 transcript 包含客户端自己的随机数和自己的封装密钥。双方都知道各自持有相同的会话密钥，且网络上的攻击者无法推导它们。服务器不会得知客户端是谁：在此范围内，它只被保证对端知道本次握手的共享秘密。

各方：

- **客户端**：希望与 srv 建立通道。持有：提供的 ML-KEM-768；服务器名 srv；CA 公钥。
- **服务器**：单一运营者，仅支持 ML-KEM-768。持有：证书 (srv, pk_srv)；签名私钥 sk_srv。

图 1 展示了文稿所规定的交换过程。



图 1. 文稿所规定的协议。

分析的范围见附录 D：分析了什么、省略了什么，以及它依据的文稿章节。

2 发现

没有发现。 分析没有发现文稿未已写明或未已排除的内容。第 3 节的表格给出了每条性质的结果。

在分析范围内，协议提供了它应当提供的东西：服务器认证、密钥份额和会话密钥的一致性、这些密钥对网络的保密性，以及关于服务器签名私钥的前向保密，对任意数量的会话都成立。没有发现：三个不成立的结果重现了文稿自己写明的弱点，即 ML-KEM 解封装密钥和封装随机性必须被擦除，因为持有它们的人

可以推导会话；以及仅认证自身的服务器无法区分合法客户端和其他任何人。这些结果依赖于符号化的 ML-KEM、公开证书目录和建模为哈希链的 transcript；记录层、编码检查和 ML-KEM 的计算安全性不在其中。要解决此抽象所移除的部分，需要 Tamarin 加上 Agent，它们带有真实的代数模型和状态模型。

3 检查过的性质

每一行是一条提交给 ProVerif 的性质。结果是 ProVerif 对分析者所建协议模型的回答；价值和文稿是否已写明该结果是分析者的判断，理由见附录 A。

ID	性质	假定的泄露	结果	价值	文稿是否已写
A02	**服务器对客户端的认证。** 客户端接受握手完成时，存在一个服务器运行，其意图通信对象是客户端身份。	无	成立	中	否
A03	**匹配服务器运行的协议一致性。** 客户端接受时，存在匹配的服务器运行，双方身份、协商的 ML-KEM-768 组、密钥份额与共享秘密一致。	无	成立	中	否
A04	**服务器运行的单射一致性。** 客户端每次接受握手完成，都对应一个不同的服务器运行（单射匹配）。	无	成立	中	否
A06	**服务器Finished密钥确认。** 客户端验证服务器 Finished 时，服务器已导出并持有本次会话相同的握手密钥。	无	成立	中	否
A08	**协商组和共享绑定。** 客户端接受的 ML-KEM-768 组、服务器密钥份额（密文）与协商参数，对应其本会话提议并经 transcript 绑定。	无	成立	中	否
K01	**会话密钥与共享秘密的保密性。** 攻击者不能推导 ML-KEM 共享秘密及派生的握手与应用流量密钥。	无	成立	高	否
C01	**服务器签名密钥的前向保密。** 会话完成后泄露服务器长期签名私钥，攻击者仍不能推导此前完成的会话密钥。	服务器长期签名私钥 (会话完成后泄露)	成立	高	否

ID	性质	假定的泄露	结果	价值	文稿是否已写
C04	**KEM密钥泄露下的会话保密性。** 泄露客户端 ML-KEM 解封装私钥后，攻击者仍不能推导该会话的共享秘密与流量密钥。	客户端 ML-KEM 解封装私钥 sk (会话期间或之后泄露)	攻击成立（文稿已写明）	低	是
C02	**密钥泄露下的服务器认证。** 服务器长期签名私钥泄露后，攻击者仍不能冒充客户端或未泄露对端使服务器接受。	服务器长期签名私钥 (握手前泄露)	攻击成立（文稿已写明）	低	是
K01.10	**会话密钥保密性与一致性。** 攻击者不能推导会话密钥，且双方接受同一会话密钥。	无	成立	高	否
A03.11	**服务器身份与转录绑定。** 客户端接受时，服务器身份与完整握手 transcript 一致。	无	成立	中	否
C01.12	**签名密钥泄露后的前向保密。** 会话完成后泄露服务器长期签名私钥，历史会话密钥仍保密；KEM 私钥泄露另行分析。	服务器长期签名私钥 (会话完成后泄露)	成立	高	否
X01	**应用数据机密性。** 握手后传输的应用数据，控制网络且不持有其他信息的攻击者无法读取。	无	成立	中	否
X02	**封装随机性下的会话保密性。** 即使攻击者获得了为某次握手抽取的 ML-KEM 封装随机性，该次握手的共享秘密和流量密钥也必须保持保密。	the encapsulation randomness that ML-KEM drew for one handshake, i.e. raw output of the RNG as described in main:295, obtained after the handshake	攻击成立（文稿已写明）	低	是
X03	**握手密钥与应用密钥的独立性。** 为握手派生的流量密钥不得泄露同一次握手或其他握手应用流量密钥。	the handshake traffic keys of one completed handshake (client_handshake_traffic_secret and server_handshake_traffic_secret, or the keys derived from them), obtained after the handshake	成立	中	否

“攻击成立（文稿已写明）”：证明器找到了这个攻击，而文稿自己已经写明它存在。这条性质不成立；这是已知结果，不算发现（附录 B）。

文稿写明而起草的清单里没有、由分析者补充的目标（见附录 D）：X01, X02, X03.

提出的问题及分析者的回答:

问题	回答	依据
Q1: 会话密钥保密及双方密钥一致性	在任意数量的会话中，会话密钥对网络上的攻击者保持保密，且双方最终持有相同的密钥；唯一能触及会话密钥的泄露是那次握手自身的临时 ML-KEM 秘密，而文稿要求将其擦除。	K01, K01.10, C04, X02
Q2: 服务器认证与 Finished 的完整 transcript 绑定	服务器通过认证，因为 CertificateVerify 签名的 transcript 包含客户端的随机数和封装密钥，且客户端只接受在该 transcript 和 ML-KEM 共享秘密派生密钥下的 Finished。	A02, A03, A03.11, A08, A06
Q3: 长期签名密钥事后泄露与 KEM 密钥泄露分开分析	事后窃取服务器签名密钥不会打开过去或当前的会话，因为会话密钥仅来自 ML-KEM 秘密；ML-KEM 解封装密钥和封装随机性是另一种暴露，确实会打开会话，正如文稿在要求擦除这些值时警告的那样。	C01, C01.12, C04, X02

4 本次分析的局限

这里的结果是对协议模型的结果：它说明规定的消息和检查提供了什么或未能提供什么，其中密码学被理想化处理，不涉及任何实现。

本次分析未覆盖的内容，如分析者所述：

- ML-KEM 是一个抽象的封装/解封装对：其计算安全性、隐式拒绝，以及 main s0026 至 s0028 的畸形密钥和密文长度检查均未建模。
- 仅建模了 ML-KEM-768；密码套件、版本和多组协商、HelloRetryRequest、PSK 和 0-RTT 不在范围内。
- 记录层未建模：没有序列号、警报、填充、密钥更新或长度隐藏，因此关于流量分析或记录级重放没有任何结论。
- transcript 是消息内容的哈希链，而非带握手头和记录分帧的字节精确 Transcript-Hash。
- 证书是来自目录的公开值，因此 Certificate 消息的保密性以及 F.1 对端点身份的保护没有被考察。
- 后妥协安全性，即在长期密钥泄露之后安排的握手，没有被考察：各阶段将所有握手置于泄露之前。
- 没有目标依赖于计数器、序列号或可替换密钥，因此没有建模跨步骤变化的值，也没有进行有界模型或状态审查。
- 可用性、时序、侧信道和请求的开销超出了本模型轨迹所能说明的范围。

每条性质仍有哪些未决问题，以及什么能解决它们，见附录 A。

附录 A：逐条性质的解读

这是分析者自己对证明器回答的解读，而非证明器的回答：这些回答对协议意味着什么、解读依据什么、能否按原样使用，以及它的价值。

A02: 服务器对客户端的认证

客户端接受握手完成时，存在一个服务器运行，其意图通信对象是客户端身份。

每当客户端接受时，存在该次握手的服务器运行，它为相同的 ML-KEM 密文发送了 Finished，且客户端接受的证书命名了它所请求的服务器。主张中关于服务器意图对端的说法在协议中没有对应物：TLS 1.3 只认证服务器，因此客户端的保证是关于服务器身份的。

- 结果：成立；回答了所问：部分；有保留地可用
- 价值 中：对等认证是该协议的核心承诺，并且对任意数量的会话都成立；价值被降低，是因为证书和转录被理想化了。我在文稿中查找“服务器认证不成立”的陈述，没有找到：F.1 将其列为一个目标。
- 文稿中有陈述：否
- 依据：证书来自公共目录，CA 密钥按假设受信任；路径校验和名称匹配不在模型范围内。；只对 ML-KEM-768 建模，因此对组间协商没有任何说明。

A03: 匹配服务器运行的协议一致性

客户端接受时，存在匹配的服务器运行，双方身份、协商的 ML-KEM-768 组、密钥份额与共享秘密一致。

接受的客户端与同一握手的服务器运行共享组标识符、自己的封装密钥、密文和 ML-KEM 共享秘密。任何网络攻击者都无法让双方以不同的份额或秘密结束。

- 结果：成立；回答了所问：是；可用
- 价值 中：密钥调度输入的一致性正是 ML-KEM 替换必须保持的东西，并且它对任意数量的会话都成立。
- 文稿中有陈述：否
- 依据：转录是消息内容的显式哈希链，而不是字节精确的 TLS 转录哈希。

A04: 服务器运行的单射一致性

客户端每次接受握手完成，都对应一个不同的服务器运行（单射匹配）。

记录的服务器飞行消息不能被呈现给第二个客户端接受：密文是对该客户端所发送的封装密钥封装的，因此每次接受都属于其自己的服务器运行，且每次握手的密钥都是新鲜的。

- 结果：成立；回答了所问：是；可用
- 价值 中：单射一致性正是使每次握手的密钥新鲜性可用的东西，并且它在没有任何触及这一点的简化下成立。
- 文稿中有陈述：否
- 依据：未对恢复建模，因此握手飞行消息是唯一的重放面。

A06: 服务器Finished密钥确认

客户端验证服务器 Finished 时，服务器已导出并持有本次会话相同的握手密钥。

当客户端接受服务器 Finished 时，它已经持有握手流量密钥，而该运行的服务器导出了相同的密钥，因此该 Finished 是真正的密钥确认。

- 结果：成立；回答了所问：是；可用
- 价值 中：密钥确认是 Finished 消息的一个陈述目标，并且它对任意数量的会话都成立。
- 文稿中有陈述：否
- 依据：该 Finished 是基于从握手流量秘密派生的 finished_key 的 MAC，而不是真正的 HMAC。

A08: 协商组和共享绑定

客户端接受的 ML-KEM-768 组、服务器密钥份额（密文）与协商参数，对应其本会话提议并经 transcript 绑定。

客户端接受的组标识符、客户端的封装密钥和服务器的密文都属于同一握手的服务器运行，因此密钥调度承诺了实际交换的值，份额无法被替换。

- 结果：成立；回答了所问：是；有保留地可用
- 价值 中：这正是使重封装攻击不适用的一点，并且是机械地展示的；保留之处在于两项被省略的编码检查。
- 文稿中有陈述：否
- 依据：FIPS 203 的封装密钥检查和密文长度检查未建模；只对组标识符和值的绑定建模。

A03.11: 服务器身份与转录绑定

客户端接受时，服务器身份与完整握手 transcript 一致。

客户端用本次握手的服务器运行的签名密钥验证 CertificateVerify，而该签名覆盖一个包含客户端随机数、客户端的封装密钥、服务器随机数和密文的转录，因此在客户端做决定之处，服务器身份与握手转录被绑定在一起。

- 结果：成立；回答了所问：是；有保留地可用
- 价值 中：CertificateVerify 的转录绑定承载了整个服务器认证，但转录被理想化了。
- 文稿中有陈述：否
- 依据：转录是内容的哈希链，而不是包含握手头部的字节精确转录哈希。

K01: 会话密钥与共享秘密的保密性

攻击者不能推导 ML-KEM 共享秘密及派生的握手与应用流量密钥。

控制网络且不持有其他任何东西的攻击者，永远无法获知客户端已认证的某次握手的 ML-KEM 共享秘密、握手流量密钥或应用流量密钥，对任意数量的会话都如此。被证伪的对照查询表明，一旦交出一个握手秘密，同一查询就会失败。

- 结果：成立；回答了所问：是；有保留地可用
- 价值 高：会话密钥的保密性是密钥交换的首要目标，并且对任意数量的会话都成立，可抵御网络攻击者。
- 文稿中有陈述：否

- 依据：ML-KEM 是抽象的封装/解封装对，没有隐式拒绝或代数结构；对其计算安全性没有任何证明。；只对 ML-KEM-768 建模。

K01.10: 会话密钥保密性与一致性

攻击者不能推导会话密钥，且双方接受同一会话密钥。

客户端已认证的某次握手，双方以相同的应用流量秘密结束，且没有网络攻击者能读取应用数据；保密性与一致性一并得出。

- 结果：成立；回答了所问：是；有保留地可用
- 价值 高：秘密且一致的会话密钥正是该信道所提供的东西，并且两半都对任意数量的会话成立。
- 文稿中有陈述：否
- 依据：记录层未建模：每个方向一条 AEAD 消息，没有序列号，没有密钥更新。

C01: 服务器签名密钥的前向保密

会话完成后泄露服务器长期签名私钥，攻击者仍不能推导此前完成的会话密钥。

当服务器的长期签名密钥在握手之后被泄露时，已完成握手的应用流量密钥仍保持秘密。模型还在同一步骤把该握手的握手流量密钥交给攻击者，因此结果比目标所要求的更强。

- 结果：成立；回答了所问：是；可用
- 价值 高：相对于签名密钥的前向保密正是让基于证书的部署能在之后的密钥失窃中存活的东西，并且它对任意数量的会话都成立。
- 文稿中有陈述：否
- 依据：泄露发生在该运行的每次握手之后，这正是目标所指的时机。；额外泄露握手流量密钥是我的选择，不是目标的一部分。

C04: KEM密钥泄露下的会话保密性

泄露客户端 ML-KEM 解封装私钥后，攻击者仍不能推导该会话的共享秘密与流量密钥。

一旦某次握手中客户端的 ML-KEM 解封装密钥被泄露，攻击者就从密文恢复封装随机数，并推导出该握手的整个密钥调度。协议中没有任何东西能阻止这一点：解封装密钥是密文与共享秘密之间唯一的屏障，这也是文稿要求将其擦除的原因。

- 结果：攻击成立（文稿中有陈述）；回答了所问：是；可用
- 价值 低：它直接来自目标所假设的暴露；文稿在要求擦除临时密钥时陈述了同样的后果。
- 文稿中有陈述：是。文稿何处如此说：rfc9846:6511
- 依据：解封装密钥在握手之后被泄露，这正是目标所指的时机。；ML-KEM 是抽象的：模型展示的是机制，而不是真实原语的计算困难性。

C02: 密钥泄露下的服务器认证

服务器长期签名私钥泄露后，攻击者仍不能冒充客户端或未泄露对端使服务器接受。

迹显示攻击者发送一个 ClientHello，其封装密钥的解封装密钥由攻击者持有；服务器对其封装，攻击者在没有任何长期密钥泄露的情况下完成两条 Finished 消息。这是攻击者作为客户端行事，本范围内的 TLS 1.3 允许这一点，因为没有客户端证书参与。在签名密钥泄露后幸存下来的是密钥持有：服务器只接受知道该握手共享秘密的对端。

- 结果：攻击成立（文稿中有陈述）；回答了所问：部分；可用
- 价值 低：该失败源于排除客户端证书的范围决定，F.1 陈述了在单向认证连接中攻击者可以与服务器建立自己的会话密钥。
- 文稿中有陈述：是。文稿何处如此说：rfc9846:6485, rfc9846:6482
- 依据：本范围内不存在客户端认证，因此对于只认证服务器的协议，主张的前一半不可能成立。；记录层未建模，因此迹只展示到服务器接受 Finished 为止。

C01.12: 签名密钥泄露后的前向保密

会话完成后泄露服务器长期签名私钥，历史会话密钥仍保密；KEM 私钥泄露另行分析。

已完成握手的历史会话密钥在服务器长期签名密钥事后泄露时仍保持秘密；只有某次握手的临时 ML-KEM 秘密被泄露，才会触及该握手的密钥。

- 结果：成立；回答了所问：是；可用
- 价值 高：它回答了草案的问题：签名密钥失窃不会打开过去的流量，而临时秘密才是必须擦除的。
- 文稿中有陈述：否
- 依据：两种暴露在一个模型中由攻击者持有它们的阶段加以区分，而不是由分离的进程区分。

X01: 应用数据机密性

握手之后传输的应用数据，对于控制网络且不掌握其他信息的攻击者来说不可读。

在应用流量密钥下传输的应用数据，对于网络上的攻击者来说不可读；对照查询显示，一旦该次握手的临时 ML-KEM 秘密被泄露，这些数据就变得可读。

- 结果：成立；回答了所问的问题：是；可在有保留的情况下使用
- 价值 中：它是信道的机密性那一半，在记录层理想化的前提下对任意数量的会话都成立。
- 文稿中有说明：否
- 依据：记录层、其填充和长度隐藏都未建模，因此对于流量分析能了解到什么没有任何说明。；载荷在每个方向上都是一个新值。

X02: 封装随机性下的会话保密性

即使攻击者拿到了某次握手所抽取的 ML-KEM 封装随机性，该次握手的共享秘密和流量密钥也必须保持机密。

一旦某次握手的封装随机性被泄露，攻击者就能推导出该次握手的共享秘密和流量密钥。在 ML-KEM 中，共享秘密正是由这份随机性构成的，所以协议本身无法阻止这一点；能阻止它的是 RNG 卫生和及时擦除该值。

- 结果：攻击成立（文稿中有说明）；回答了所问的问题：是；可用
- 价值 低：它直接来自目标所假设的暴露，文稿本身也已对此提出警告；复现它是为了表明模型能看到这一点。
- 文稿中有说明：是。文稿中的位置：main:295
- 依据：共享秘密被建模为封装随机性的单向函数，这是对“解封装方能够精确恢复该随机性”这一陈述的忠实解读。；泄露被建模为发生在握手之后。

X03: 握手密钥与应用密钥的独立性

为握手推导出的流量密钥不得泄露同一次握手的应用流量密钥，也不得泄露另一次握手的应用流量密钥。

已完成握手的握手流量密钥无法得出同一次握手的应用流量密钥，不同的握手秘密之间也保持独立。密钥调度标签正是将它们区分开的东西。

- 结果：成立；回答了所问的问题：是；可在有保留的情况下使用
- 价值 中：这是握手密钥泄露仍可承受的原因，但这种分离是 KDF 的性质，而模型将其视为给定。
- 文稿中有说明：否
- 依据：这种分离依赖于将 HKDF 标签符号化地处理为不同的常量；KDF 中的碰撞不在本模型所能判定的范围内。

附录 C：主张和证明器的答案

证明器对每个目标的记录：达到的级别、达到该级别的查询、是否检查了对照查询，以及每个攻击及分析者对其的判定。

目标	级别	界	查询	对照	攻击	主张
A02	proved		a02	checked		1 proved
A03	proved		a03	checked		1 proved
A04	proved		a04	checked		1 proved
A06	proved		a06	checked		1 proved
A08	proved		a08	checked		1 proved
K01	proved		k01	checked		1 proved
C01	proved		c01	checked		1 proved
C04	none				c04 (holds_in_source)	1 attack
C02	none				c02 (holds_in_source)	1 attack
K01.10	proved		k10	checked		2 proved
A03.11	proved		a11	checked		1 proved
C01.12	proved		c01	checked		1 proved
X01	proved		x01	checked		1 proved
X02	none				c04 (holds_in_source)	1 attack
X03	proved		c01	checked		1 proved

逐条主张，按分析者对每条主张的措辞列出；每条的答案即证明器的回答：

目标	主张	攻击者	回答	查询	对照
A02	只有当本次握手确实存在一个服务器运行实例时，客户端才会接受：每一次接受都对应着一个为同一 ML-KEM 密文发送了 Finished 的服务器。TLS 1.3 没有用于表示服务器预期对端的客户端身份，因此客户端所能得到的保证是：它检查的对端身份就是它所指定的那台服务器，并且存在一个与之匹配的服务器运行实例。	`network`	proved	a02	checked
A03	当客户端接受时，存在一个同一握手的服务器运行实例，其 ML-KEM-768 组、封装密钥、密文和共享秘密都相同。	`network`	proved	a03	checked
A04	每一次客户端接受都对应着一个不同的服务器运行实例：一段被录下的服务器消息序列不能被重放给第二次客户端接受，因为密文绑定在客户端自己新生成的封装密钥上。	`network`	proved	a04	checked
A06	当客户端接受服务器的 Finished 时，它已持有握手流量密钥，而该运行实例的服务器也已推导出相同的握手流量密钥（密钥确认）。	`network`	proved	a06	checked
A08	客户端所接受的组标识符、客户端封装密钥和服务器密文，都属于本次握手的某个服务器运行实例，因此密钥调度所承诺的密钥份额无法被替换为另一个。	`network`	proved	a08	checked

目标	主张	攻击者	回答	查询	对照
K01	对于任意数量的会话，仅掌握网络的攻击者都无法推导出客户端已认证的握手的 ML-KEM 共享秘密，以及握手流量密钥和应用流量密钥。	`network`	proved	k01	checked
C01	在服务器的长期签名密钥被泄露之后（以及随之泄露的已完成握手的握手流量密钥），该握手的应用流量密钥对攻击者仍然未知。	`server_sig_key`	proved	c01	checked
C04	在某次握手的客户端 ML-KEM 解封装密钥被泄露之后，该握手的共享秘密和流量密钥仍保持机密。	`client_kem_key`	attack (holds_in_source)	c04 (the control of c01)	
C02	持有服务器长期签名密钥的攻击者无法冒充客户端：服务器所接受的每次握手都对应着一个发送了 Finished 的客户端运行实例。协议用来替代客户端认证的机制是：服务器只得到密钥持有性的保证。	`server_sig_key`	attack (holds_in_source)	c02 (the control of k10)	
K01.10	客户端已认证的握手中，双方最终得到相同的会话密钥：对于同一密文，客户端的应用流量秘密等于服务器的应用流量秘密。	`network`	proved	k10	checked
K01.10	已认证握手的会话密钥，对于仅在网络上的攻击者保持机密，并且双方对这些密钥达成一致。	`network`	proved	k01	checked
A03.11	Certificate 中的服务器身份以及 CertificateVerify 所覆盖的 transcript：客户端验证的签名密钥属于本次握手的一个服务器运行实例，且签名覆盖了客户端自己的随机数、客户端的封装密钥、服务器的 Random、密文和证书。	`network`	proved	a11	checked
C01.12	已完成握手的历史会话密钥，在服务器的长期签名密钥事后被泄露时仍保持机密；ML-KEM 私钥和封装随机性作为单独的暴露来分析，此处不予泄露。	`server_sig_key`	proved	c01	checked
X01	握手之后发送的应用数据，对于仅掌握网络的攻击者来说不可读。	`network`	proved	x01	checked
X02	当为某次握手抽取的 ML-KEM 封装随机性被泄露时，该次握手的共享秘密和流量密钥仍保持机密。	`encap_randomness`	attack (holds_in_source)	c04 (the control of c01)	
X03	已完成握手的握手流量密钥不会泄露同一次握手的应用流量密钥，也不会泄露另一次握手的应用流量密钥。	`handshake_key`	proved	c01	checked

附录 D：分析是怎么跑的

一个 LLM 分析者根据文稿构建协议模型，并把每个性质交给 ProVerif；第 3 节中的每个结果都是证明器的回答。

ProVerif+ Agent 会话 session-20261114T210655, 2026-11-14。状态：已完成。模型：deepseek-v4-1-flash-261017。步数：37，证明器 164 秒，实际耗时 14 分钟，成本 0.117 美元。

ProVerif：运行了 43 个证明，43 个已判定，0 个无答案，157 秒，记录了 16 个答案。

分析了什么

协议	TLS 1.3 pure ML-KEM-768 1-RTT
版本	draft-ietf-tls-mlkem-11
分析范围	固定 ML-KEM-768、服务器签名认证和完整 1-RTT；TLS 基线固定 RFC9846，注明相对草案引用版本的更新；包含 CertificateVerify/Finished 对实际握手 transcript 的绑定和加密握手消息。排除 HRR、PSK、0-RTT、客户端证书。
不在范围内	符号模型不证明算法的计算安全或抗量子安全；攻击轨迹只在明确假设下成立，不直接等同规范漏洞。每个性质必须有可达性/正例控制。
原文	来自下列各节的 592 个带编号的原文单元（151676 字节）。本报告中诸如 s0001 的引用即其中之一

分析所依据的文稿各节：

文稿	章节	标题
main	3	密钥封装机制
main	4	构造
main	4.1	协商
main	4.2	传输封装密钥与密文
main	4.3	共享密钥计算
rfc9846	2	协议概述
rfc9846	2.1	错误的 DHE 共享
rfc9846	2.2	恢复与预共享密钥（PSK）
rfc9846	2.3	0-RTT 数据
rfc9846	4	握手协议
rfc9846	4.1	转录哈希
rfc9846	4.2	密钥交换消息
rfc9846	4.2.1	密码协商
rfc9846	4.2.2	客户端问候
rfc9846	4.2.3	服务器问候
rfc9846	4.2.4	Hello 重试请求
rfc9846	4.3	扩展

文稿	章节	标题
rfc9846	4.3.1	支持的版本
rfc9846	4.3.2	Cookie
rfc9846	4.3.3	签名算法
rfc9846	4.3.4	证书颁发机构
rfc9846	4.3.5	OID 过滤器
rfc9846	4.3.6	握手后的基于证书的客户端认证
rfc9846	4.3.7	支持的群组
rfc9846	4.3.8	密钥共享
rfc9846	4.3.8.1	Diffie-Hellman 参数
rfc9846	4.3.8.2	ECDHE 参数
rfc9846	4.3.9	预共享密钥交换模式
rfc9846	4.3.10	早期数据指示
rfc9846	4.3.11	预共享密钥扩展
rfc9846	4.3.11.1	票据年龄
rfc9846	4.3.11.2	PSK 绑定器
rfc9846	4.3.11.3	处理顺序
rfc9846	4.4	服务器参数
rfc9846	4.4.1	加密扩展
rfc9846	4.4.2	证书请求
rfc9846	4.5	认证消息
rfc9846	4.5.1	证书
rfc9846	4.5.1.1	OCSP 状态与 SCT 扩展
rfc9846	4.5.1.2	证书选择
rfc9846	4.5.1.3	接收证书消息
rfc9846	4.5.2	证书验证
rfc9846	4.5.3	完成
rfc9846	4.6	早期数据结束
rfc9846	4.7	握手后消息
rfc9846	4.7.1	新会话票据消息
rfc9846	4.7.2	握手后认证
rfc9846	4.7.3	密钥与初始化向量更新
rfc9846	7	密码计算
rfc9846	7.1	密钥调度
rfc9846	7.2	更新流量密钥
rfc9846	7.3	流量密钥计算
rfc9846	7.4	非对称共享密钥计算

文稿	章节	标题
rfc9846	7.4.1	有限域 Diffie-Hellman
rfc9846	7.4.2	椭圆曲线 Diffie-Hellman
rfc9846	7.5	导出器

源文步在文稿中定位到的协议目的：固定 ML-KEM-768、服务器签名认证和完整 1-RTT；TLS 基线固定 RFC9846，注明相对草案引用版本的更新；包含 CertificateVerify/Finished 对实际握手 transcript 的绑定和加密握手消息。排除 HRR、PSK、0-RTT、客户端证书。

对照文稿定性质

在编写查询之前，分析者将起草的性质清单与文稿本身对协议目的的陈述进行了对照。名称、性质所涉及的内容以及文稿的陈述均属分析者的解读；所提及的位置确实存在于文稿中。

ID	简称	对象	文稿怎么说	出处
A02	服务器对客户端的认证	客户端：它必须以对端身份结束握手，且对端身份正是它所请求的服务器（rfc9846 F.1 Peer Authentication）；TLS 1.3 只认证服务器，在本范围内客户端一侧未经认证	F.1 指出“Peer Authentication: The client's view of the peer identity should reflect the server's identity”，针对的是主动网络攻击者。	rfc9846:6490
A03	匹配服务器运行的协议一致性	客户端和服务器：两者必须以相同的会话密钥、相同的 ML-KEM-768 组、相同的封装密钥和作为 key_share 的密文，以及相同的共享秘密结束	F.1 的“Establishing the same session keys”和“Downgrade Protection”；main s0023 s0024 s0025 把 pk 和 ct 放入 key_share，s0034 把 ML-KEM 共享秘密放入密钥编排。	rfc9846:6478, rfc9846:6499, s0023, s0024, s0025, s0034
A04	服务器运行的单射一致性	每次客户端接受背后的服务器运行：每次接受必须对应一次不同的服务器握手，使得录制的报文序列不能被重放两次	F.1“Uniqueness of the session keys: Any two distinct handshakes should produce distinct, unrelated session keys”；ClientHello.random 和服务器 Random 使每次握手各不相同（s0535）。	rfc9846:6495, rfc9846:6535, s0535
A06	服务器Finished密钥确认	服务器：在客户端接受其 Finished 之前，它必须已经持有本次会话的握手流量密钥	F.1“Establishing the same session keys”；s0063 指出 Finished 是对握手的 MAC，提供密钥确认；s0484 指出 finished_key 由 Base Key 派生，而 Base Key 是 server_handshake_traffic_secret（s0406）。	rfc9846:6483, s0063, s0406, s0484
A08	协商组和共享绑定	命名组、客户端的封装密钥和服务器端的密文，即客户端所接受的这些内容：它们必须是本次握手的，并且被转录承诺	main s0026..s0027 要求对封装密钥和密文做编码检查，main:268 指出密钥编排通过 key_share 承诺 ML-KEM 封装密钥和密文，从而抵御重封装攻击。	main:268, s0011, s0018, s0023, s0024, s0025, s0026, s0027, s0111
K01	会话密钥与共享秘密的保密性	ML-KEM 共享秘密以及由它派生的握手流量密钥和应用流量密钥，对网络攻击者保密	F.1“Secrecy of the session keys”；main s0034 把 ML-KEM 共享秘密插入 TLS 1.3 密钥编排以取代 (EC)DHE 秘密，s0534 s0554 由它派生流量密钥。	rfc9846:6483, s0034, s0535, s0554

ID	简称	对象	文稿怎么说	出处
C01	服务器签名密钥的前向保密	已完成握手的会话密钥，在握手完成后攻击者获得服务器长期签名私钥的情况下仍然保密	F.1“Forward secret with respect to long-term keys: If the long-term keying material (in this case the signature keys in certificate-based authentication modes) is compromised after the handshake is complete, this does not compromise the security of the session key”。	rfc9846:6504
C04	KEM密钥泄露下的会话保密性	某次会话的 ML-KEM 共享秘密和流量密钥，在攻击者持有该会话的 ML-KEM 解封装密钥的情况下仍然保密	F.1 指出前向保密仅在“as long as the session key itself (and all material that could be used to recreate the session key) has been erased”时成立，并把“private keys corresponding to key shares”列为必须擦除的材料之一；main s0032 指出持有解封装密钥 sk 的一方可以精确恢复封装随机性。	rfc9846:6511, s0032
C02	密钥泄露下的服务器认证	服务器对其对端的保证：即使攻击者持有服务器的长期签名密钥，它也只接受运行了本次握手客户端一侧的对端	F.1“Key Compromise Impersonation (KCI) resistance: compromising the long-term secret of one actor should not break that actor's authentication of their peer in the given connection”；客户端证书认证不在本范围内，因此服务器所拥有的对端认证即密钥持有。	rfc9846:6522
K01.10	会话密钥保密性与一致性	一次握手的会话密钥对网络攻击者保密，且双方产生相同的会话密钥	F.1“Establishing the same session keys”和“Secrecy of the session keys”。	rfc9846:6478, rfc9846:6483
A03.11	服务器身份与转录绑定	Certificate 中的服务器身份和 CertificateVerify 所签名的握手转录，即客户端接受时所看到的这些内容	s0403 指出 CertificateVerify 签名的是 Transcript-Hash(Handshake Context, Certificate)，s0452 指出该消息“also provides integrity for the handshake up to this point”；F.1“Peer Authentication”。	rfc9846:6490, s0403, s0452, s0456
C01.12	签名密钥泄露后的前向保密	已完成握手的历史会话密钥，在攻击者事后获得服务器长期签名密钥的情况下仍然保密；KEM 私钥作为单独的泄露情形分析	F.1“Forward secret with respect to long-term keys”，以及另一句话：本应临时的密钥必须被擦除，否则实际就变成了长期密钥。	rfc9846:6504, rfc9846:6516
X01	应用数据机密性	客户端和服务器在应用流量密钥下交换的应用数据	rfc9846:214“Confidentiality: Data sent over the channel after establishment is only visible to the endpoints”，针对的是完全控制网络的攻击者（rfc9846:223）。	rfc9846:214
X02	封装随机性下的会话保密性	一次握手的 ML-KEM 共享秘密和流量密钥	main:295 警告不安全 RNG 输出的泄露可能危及 RNG 状态，并且 ML-KEM 的封装随机性是原始 RNG 输出可能被泄露的又一个位置；main s0032 指出持有 sk 的一方可以精确恢复该随机性，因此这种泄露是一个具体需要询问的暴露面。	main:295, s0032
X03	握手密钥与应用密钥的独立性	应用流量密钥，在攻击者持有已完成握手的握手流量密钥的情况下仍然保密	F.1“Individual session keys produced by a handshake should also be distinct and independent”（rfc9846:6497），以及 rfc9846:6656 关于按 key 和 label 区分 HKDF-Expand 应用的内容。	rfc9846:6497

分析者依据文稿补充的性质：X01, X02, X03.

文稿里的安全表述，逐段去向

宿主把文稿里说明协议保护什么、防什么的章节逐段列出，共 8 段；分析者对每一段说明了去向：定为性质、定为攻击者条件，或者为什么都不选。选了 4 段，没选 4 段。列哪些段落由标准的章节结构决定；每一段的去向是分析者的判断。

段落	出处	定为	不选的原因
P01	main:253-266	K01, A03, `network`	
P02	main:268-273	A08, A03, `network`	
P03	main:275-278	证明器问不了	它让实现者参考 NIST SP 800-227，并建议采用抗侧信道的实现；侧信道和实现质量不在符号模型的范围内。
P04	main:280-284	没有陈述目标	这是部署建议：实现者必须权衡自身的安全、性能和运维约束，并参考注册表的 Recommended 列。它没有陈述任何必须成立或可能被攻击的内容。
P05 至 P06	main:288-292	没有陈述目标	这两个原文单元是草案的页脚和页眉（有效期和页码、Internet-Draft 名称和日期）；它们没有陈述任何目标。
P07	main:295-302	X02, `encap_randomness`	
P08	rfc9846:4676-4677	A02, A03, A04, K01, A08, C01, C02, `network`, `server_sig_key`	

性质目录，逐条去向

性质是从一份安全和隐私性质目录起草的。起草时有 33 条没选，分析者把它们逐条过了一遍：作为性质在这里检查，或者写明原因不检查。检查了 10 条，没检查 23 条。“文稿没有提到”不能作为不检查的原因。

条目	目录里的性质	作为哪条性质检查	不检查的原因
K02	载荷与端到端保密性	X01	
K03	强保密／秘密值不可区分性	K01, X01	
K04	密钥用途、方向与会话分离	X03	
D01	消息完整性与数据来源真实性	A03, A08	
A01	对端曾参与性（Aliveness）	A02	
A05	隐式密钥认证	A03, K01	
A07	身份—凭据—密钥绑定	不在范围内	证书与身份验证留给应用协议处理。在本范围内，按假设证书把服务器名绑定到签名密钥上，因此凭据之间的错误绑定不可能在模型内部出现。
R01	认证／挑战新鲜性	A04	
R02	消息级抗重放	A04	
R03	业务操作至多一次	不适用	握手本身不执行任何计费或消耗性操作。
R04	消息接受顺序／窗口安全	不适用	本范围内的 1-RTT 握手没有接受窗口、计数器或重排序规则；记录层自己的窗口不在建模部分之内。
R05	抗降级／安全策略保持	A08	
C03	已知密钥安全／泄露隔离	C01, K01	
C05	泄露后安全恢复（PCS）	证明器问不了	泄露后恢复需要一次排在长期密钥泄露之后的握手，而这次握手无法与性质集其余部分所需的那几次更早的握手一起提问；文稿已经给出了答案，即长期秘密泄露之后的握手，TLS 不提供安全性（rfc9846:6711）。

条目	目录里的性质	作为哪条性质检查	不检查的原因
C06	注册凭据泄露后的增强认证	不适用	这里的协议没有注册记录，也没有基于口令的凭据：PSK 和客户端证书不在范围内。
C07	抗离线字典攻击	不适用	没有任何口令或其他低熵的人类秘密参与。
C08	指定原语失效下的目标保持	证明器问不了	在本范围内，密钥建立完全由 ML-KEM 完成，没有第二组分可以顶上；而某个原语是否失效，是符号模型无法裁定的计算问题。
Z01	授权执行正确性	不适用	握手不授予任何资源访问权限。
Z02	委托权限不扩张	不适用	不存在委托。
Z03	授权与身份／会话／操作绑定	不适用	没有需要绑定的授权令牌。
Z04	撤销与过期生效	不适用	握手内部没有可撤销或会过期的权限。
Z05	跨租户／跨服务安全隔离	不适用	协议中没有多租户边界。
S01	加密 nonce 唯一性	不在范围内	AEAD nonce 属于记录层，本范围不对其建模；main s0029 确实禁止复用 ML-KEM 密文，而模型让每次握手都使用新的密文。
S02	重传／密钥重安装安全	不适用	运行在可靠流上的 TLS 1.3 没有重传，握手过程中也没有密钥重安装步骤。
S03	抗回滚与状态单调性	不适用	本范围内的握手不保留任何对回滚敏感的版本或计数器。
S04	跨阶段安全上下文一致性	不适用	不存在会对不上的注册、迁移或恢复阶段。
S05	终止与旧状态失效	不在范围内	告警处理和连接拆除不在所建模的握手之内。
P01	身份匿名性	不在范围内	匿名性在这里不是目标：TLS 1.3 用证书认证服务器，而任务范围是单个具名服务器。
P02	会话不可关联性	不在范围内	会话之间的不可关联性不是本范围的目标，本范围固定一个服务器且没有恢复。
P03	用户／凭据存在性隐藏	不适用	协议没有可被响应枚举的用户或凭据数据库。
V01	条件性进展／错误恢复	证明器问不了	ProVerif 判定的是迹上的性质，而不是一次运行是否总能取得进展。
V02	无非预期死锁	证明器问不了	无死锁性不是这类符号证明器能够判定的性质。
V03	受限资源滥用／抗放大	证明器问不了	资源滥用与放大超出了这个模型的迹所能表达的内容。

参与方和受保护的值，逐项是否设为攻击者条件

宿主把起草的性质里点到的参与方和受保护的值逐项列出，共 18 项；分析者对每一项说明了哪个攻击者条件控制这个参与方或持有这个值，或者为什么不设。设了攻击者条件的 5 项，没设的 13 项。清单来自性质本身；每一项怎么处理是分析者的判断。

ID	参与方或值	类别	攻击者条件	不设的原因
S01	客户端	参与方	`client_kem_key`	
S02	服务器	参与方	`server_sig_key`	
S03	攻击者	参与方	攻击者得不到更多	这就是网络攻击者本身；除了它的攻击者条目所述之外，它不持有任何东西。
S04	服务器身份	值	攻击者得不到更多	服务器身份按设计就是公开的：客户端知道自己调用的是哪个服务器，而证书是公开值。
S05	客户端身份	值	攻击者得不到更多	客户端证书认证不在范围内，因此攻击者从客户端身份得不到任何好处：目标清单里没有任何性质取决于它。

ID	参与方或值	类别	攻击者条件	不设的原因
S06	ML-KEM 共享秘密	值	按定义全部失守	谁持有某次握手的 ML-KEM 共享秘密，谁就能导出它的所有流量密钥；按定义，该握手的保密性已经失去，在这种暴露下没有目标对此提问。
S07	协商参数	值	攻击者得不到更多	协商参数在本范围内是固定的 ML-KEM-768 组标识符，属于线路上的公开值。
S08	握手 transcript	值	攻击者得不到更多	transcript 由各条消息本身组成，网络攻击者本来就能看到；它无法提供的是各推导所取的秘密。
S09	服务器 Random	值	攻击者得不到更多	服务器 Random 在 ServerHello 中明文发送。
S10	握手流量密钥	值	`handshake_key`	
S11	NamedGroup 标识	值	攻击者得不到更多	NamedGroup 标识符是注册表值，明文发送。
S12	ML-KEM 密文	值	攻击者得不到更多	ML-KEM 密文就是服务器的公开 key_share 值，在 ServerHello 中明文传输。
S13	应用流量密钥	值	按定义全部失守	持有应用流量密钥就能解密该应用数据；没有目标会对应用密钥已被交出的握手提问性质。
S14	历史会话密钥	值	按定义全部失守	历史会话密钥按定义就交出了该会话；仍然成立的那些性质改为在 KEM 密钥和签名密钥攻击者条件下提问。
S15	服务器长期签名私钥	值	`server_sig_key`	
S16	ML-KEM 解封装私钥	值	`client_kem_key`	
S17	会话共享秘密	值	按定义全部失守	某次握手的共享秘密给出该握手的全部流量密钥；目标清单里没有任何性质是在持有它的参与方条件下提问的。
S18	会话密钥	值	按定义全部失守	攻击者持有的会话密钥保护不了任何东西；在这种暴露下没有目标会对该次握手提问性质。

每条性质在各攻击者条件下的答案

攻击者是分析者设定的，在写查询之前就已确定；每个答案都是在其中之一条件下记录的。单元格是证明器在该攻击者条件下对该性质给出的答案；“无答案”是指该性质的那个攻击者集合下，证明器把某条主张留作未定，或者没有记录任何内容；空单元格是指分析者没有为该性质设定的攻击者。

攻击者	持有什么	依据	出处
`network`	仅网络：它读取、丢弃、修改和发送消息，不持有任何一方的秘密		
`server_sig_key`	服务器的长期签名私钥，即服务器上存放的那份，在一次握手之前或期间获得	F.1 正是针对这把密钥讨论了前向保密和 KCI 抵抗（rfc9846:6504, rfc9846:6522），而 F.1.5 说该密钥泄露之后的握手，TLS 不提供安全性。	rfc9846:6504, rfc9846:6522, rfc9846:6711
`client_kem_key`	客户端为本次握手生成的 ML-KEM-768 解封装密钥 sk，在握手期间或之后从客户端内存中读出	F.1 要求将与 key share 对应的私钥擦除，并警告不这样做实际上就制造了长期密钥（rfc9846:6511）；main s0032 说持有 sk 的一方能精确恢复封装随机性。	rfc9846:6511, s0032
`encap_randomness`	ML-KEM 为一次握手抽取的封装随机性，即在握手之后获得的、main:295 所述 RNG 的原始输出	main:295 说不安全 RNG 的输出一旦泄露，可被用来危及该 RNG 的状态，并指出 ML-KEM 封装随机性是原始 RNG 输出可能被泄露的另一个地方；main s0032 说持有 sk 的对端能精确恢复这一随机性，因此凡是看到它的人就持有共享秘密。	main:295, s0029, s0032

攻击者	持有什么	依据	出处
handshake_key	一次已完成握手的握手流量密钥（client_handshake_traffic_secret 和 server_handshake_traffic_secret，或由它们导出的密钥），在握手之后获得	F.1 要求擦除握手秘密，并说一次握手的各个会话密钥应当彼此不同且相互独立（rfc9846:6495, rfc9846:6511）。	rfc9846:6495, rfc9846:6511

ID	network	server_sig_key	client_kem_key	encap_randomness	handshake_key
A02	成立				
A03	成立				
A04	成立				
A06	成立				
A08	成立				
K01	成立				
C01		成立			
C04			不成立		
C02		不成立			
K01.10	成立				
A03.11	成立				
C01.12		成立			
X01	成立				
X02				不成立	
X03					成立

原文和目标

原文：592 个原文单元，151676 字节，冻结于 source-delivery-01.json。 目标：9 个已批准（A02, A03, A04, A06, A08, K01, C01, C04, C02）；所提问题：3 个。 目标清单按草案事先获得批准，使用 --approve-targets。

宿主按其记录在案的默认设置（任务 `decisions: auto`）做出的决定，请你检查：

- approve_package：批准源文包 protocol-1701ef07bfcd5a63586c0adfc95e

这一步自报的情况：1 次请求（1 次失败），0.000 USD，输入 70042 tokens，输出 4080 tokens，耗时 26 s。

关于证明器和模型的说明

分析者为目标命名的内容：

步骤	目标	特征	原因
29	A02, A03, A04, A06, A08, A03.11, K01, K01.10, C01, C01.12, C02, C04, X01, X02, X03	plain_primitives, many_sessions	模型只使用哈希、HKDF（extract、derive-secret、finished key）、一个 MAC、AEAD 加密、一个签名，以及作为抽象封装/解封装对的 ML-KEM，完全不含代数定律：ML-KEM 是一个 KEM，其解封装返回封装随机数，绝不是群运算。每个目标都按任意数量的会话提出，握手抄本被建模为显式的哈希链，而不是真实的抄本（这是对附录 F.1 抄本哈希的忠实呈现，但属于理想化：模型没有长度字段、没有记录层、也没有填充）。

分析者的结束说明

纯 ML-KEM-768 的 TLS 1.3 模型（服务器证书认证，完整 1-RTT），依据 main s0021-s0034 和 rfc9846 图 1、第 4.5 节和第 7.1 节构建，带有按阶段分隔的攻陷阶梯（阶段 0 仅握手；阶段 1 加上服务器的签名密钥和握手流量密钥；阶段 2 再加上临时 ML-KEM 秘密）。

结果：全部 15 个目标都有答案。对任意数量的会话均证明成立：服务器认证（A02）、对群、共享值和共享秘密的一致（A03）、单射一致（A04）、服务器 Finished 密钥确认（A06）、所接受的群和密钥共享的绑定（A08）、服务器身份与抄本的绑定（A03.11）、共享秘密和流量密钥对网络的保密性（K01, K01.10）、对应用密钥的一致（K01.10）、应用数据机密性（X01）、服务器签名密钥泄露后的前向保密（C01, C01.12），以及握手密钥与应用密钥的独立性（X03）。被证伪、且每一步都是原文所允许的：攻击者充当客户端时被服务器接受，而没有任何客户端认证（C02, rfc9846:6485）；泄露客户端的 ML-KEM 解封装密钥（C04, rfc9846:6511）或泄露封装随机数（X02, main:295）会给出整个密钥调度。这三处失败都在文稿中写明，因此不构成发现。

受阻 / 未覆盖：ML-KEM 是抽象的（没有隐式拒绝，没有 FIPS 203 编码或长度检查），抄本是哈希链而非逐字节精确的哈希，记录层、流量分析和攻陷后安全性不在模型范围内，也没有任何目标依赖于跨步骤变化的值，因此不适用有界模型或状态审查。代数和记录层部分需要 Tamarin+Agent。笔记见 notes.md。