TLS 复合 ML-DSA 认证的安全性分析

本报告的其他语言版本：English。

draft-ietf-tls-composite-mldsa-00 · ProVerif+ Agent 报告

文稿	Use of Composite ML-DSA in TLS 1.3（协议定义）; The Transport Layer Security (TLS) Protocol Version 1.3（辅助）; Post-Quantum Traditional (PQ/T) Hybrid Key Agreement Mechanisms for TLS 1.3（辅助）
检查的性质	12 条：11 条成立，1 条不成立（1 个攻击）
发现	无；另在附录 B 复现已知结论：2

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 B：复现的已知结论
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

草案将复合 ML-DSA 用于 TLS 1.3 服务器认证：服务器证书携带一个 ML-DSA 密钥和一个传统密钥（Ed25519 或 ECDSA），CertificateVerify 携带每个分量对同一 TLS 签名输入所做的签名。密钥协商使用 X25519MLKEM768，其共享秘密由 ML-KEM 秘密与 X25519 秘密拼接而成。要求复合认证的客户端持有服务器的复合公钥，并校验两个分量签名；分析中还考虑了第二个客户端，它仍然接受仅含传统密钥的证书。

交换结束时，客户端确信持有复合私钥的服务器参与了本次连接，签名是用协商的方案、在服务器 context string 下对本次连接的转录做出的，并且自己持有的流量密钥无人能够推导。服务器确信客户端计算出了相同的共享秘密和相同的转录。任一分量密钥泄露都不会破坏这两项保证；会话结束后整个长期密钥泄露，会话的流量密钥仍然保密，因为密钥协商是临时的。

参与方：

- **客户端**：要求复合证书；固定复合公钥。持有：固定的 pk_MLDSA、pk_trad；对证书的信任。
- **服务器**：持有复合长期签名密钥。持有：sk_MLDSA、sk_trad；证书（密钥类型为复合，pk_MLDSA、pk_trad）。

图 1 展示了文稿规定的交换过程。

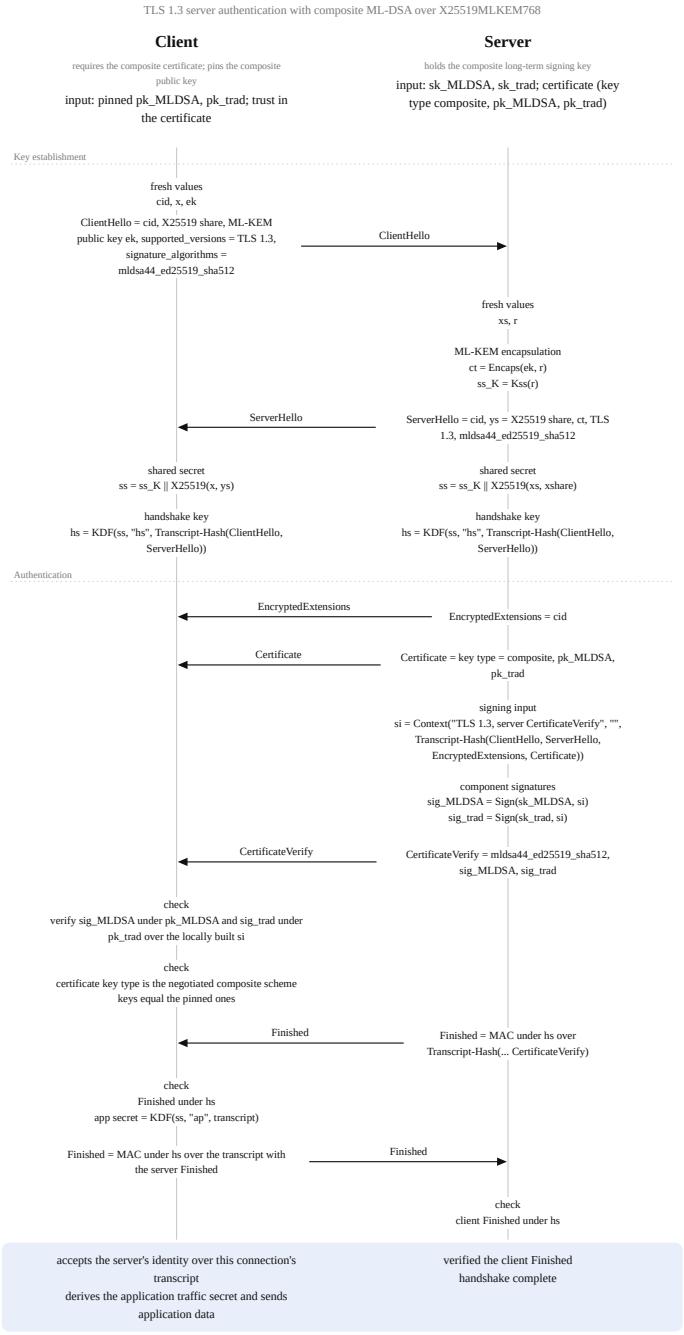


图 1. 文稿规定的协议。

分析范围见附录 D：分析了什么、省略了什么，以及依据文稿的哪些章节。

2 发现

没有发现。 分析没有发现文稿未写明或未排除的问题。性质不成立之处，文稿已知晓：附录 B 复现了 2 条此类结果。

在所分析的范围内，复合握手的行为符合草案意图：仅面对网络时、持有一个分量私钥时、X25519 一半被攻破时、会话密钥被读取时，以及存在来自其他协议上下文的签名时，只有当真实服务器用复合方案对本次连接的转录签名，客户端才接受服务器的 CertificateVerify；任何受限方案、错误密钥类型或 TLS 1.2 连接都不会被接受。流量秘密保持保密，会话结束后泄露的长期签名密钥不会暴露它们。唯一不成立的性质正是草案所警告的：同时接受仅含传统密钥证书的客户端会被传统分量密钥冒充，而复合客户端会拒绝同一

攻击者；此处以确切条件复现，因此这是已知结果而非发现。全部十二个目标都有答案，但这些证明是关于签名和转录的证明：模型中握手飞行未加密，密钥调度是单向的，因此没有证明记录层方面的任何内容，也没有证明攻击者获知共享秘密时的任何内容。

3 检查过的性质

每行是一条提交给 ProVerif 的性质。结果是 ProVerif 对分析者所建协议模型的回答；价值以及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	**每次接受背后都有实时服务器运行。** 客户端接受服务器 CertificateVerify 并完成握手时，存在一个以该客户端为意图对端的服务器运行。	无	成立	中	是
A03	**对等方、方案、秘密、记录的一致性。** 客户端接受服务器认证时，存在匹配的服务器运行，双方身份、协商签名方案、密钥交换共享秘密与转录一致。	无	成立	中	是
A04	**每次服务器运行仅一次接受。** 客户端每次接受服务器认证，都对应一个不同的服务器运行；同一服务器运行不能被两次接受。	无	成立	中	是
A08	**方案、密钥、上下文与记录的绑定。** 客户端接受的 CertificateVerify 所用签名方案、证书公钥类型、context string 与转录，均对应客户端提议与当前连接上下文。	无	成立	中	是
K01	**密钥对网络的保密性。** 攻击者不能获得服务器长期签名私钥，也不能获得诚实会话派生的握手与流量密钥。	无	成立	中	是
C01	**流量密钥的前向保密。** 会话完成后泄露服务器长期签名私钥，不暴露此前完成会话的握手与流量密钥。	服务器长期签名私钥 (会话完成之后)	成立	低	是
C08	**认证在一个组件损坏后仍成立。** 在传统分量签名私钥或 ML-DSA 分量签名私钥之一被攻破时，服务器认证目标（客户端接受真实服务器）仍成立。	组合签名的一个分量私钥 (攻击者获得分量私钥后)	成立	中	是
R05	**方案或版本不可降级。** 客户端接受的签名方案与协议版本满足草案规定的最低策略，攻击者不能通过操纵协商使接受配置降级。	无	成立	中	是
C08.9	**一个组件泄露后认证仍成立。** 在组合签名的一个分量私钥泄露时，客户端仍只接受真实服务器的认证。	组合签名的一个分量私钥 (攻击者获得分量私钥后)	成立	中	是
A08.10	**签名者角色、上下文与记录绑定。** 客户端接受的 CertificateVerify 的签名者角色、context string 与转录，均对应本连接的真实上下文。	无	成立	中	是

ID	性质	假定的泄露	结果	价值	文稿是否已写
A08.11	**方案与证书密钥兼容。** 客户端接受的签名方案与证书公钥类型兼容；不匹配时连接被拒绝而非接受。	无	成立	中	是
X01	**仅传统回退冒充。** 同时接受仅含传统密钥服务器认证的客户端，在攻击者持有传统分量私钥时，会接受一个并未发生的服务器运行；而要求复合证书的客户端会持续拒绝该攻击者。	服务器复合证书中传统分量（Ed25519/ECDSA）的私钥，在 CRQC 可用后获得，只能以该分量签名	攻击成立（文稿已写明）	中	是

“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）。

分析者补充的目标，文稿已写明但起草清单中未列出（见附录 D）：X01。

提出的问题及分析者的回答：

问题	回答	依据
Q1: 单个签名分量泄露时服务器认证	对要求复合证书的客户端而言，是：只持有一个分量私钥，且只有该分量对其选定签名输入的一个签名时，攻击者永远不会被接受，因为同一转录上缺少第二个分量；而同时接受仅传统认证的客户端会被同一把单一密钥冒充。	C08, C08.9, A02, X01
Q2: CertificateVerify 的角色/context/transcript 绑定	被接受的 CertificateVerify 绑定到服务器角色及其 context string、协商方案和本次连接的转录；同一密钥在其他协议上下文中做出的复合签名无法通过校验。	A08.10, A08, A02
Q3: 签名方案协商和证书/公钥算法不匹配拒绝	不匹配会被拒绝而非接受：命名 RSA-PKCS1-v1_5 复合方案或单独 ML-DSA 的 CertificateVerify，以及密钥类型不是协商复合方案的证书，均不会被接受。	A08.11, R05

4 本次分析的局限

这里的结果是协议模型上的结果：它说明规定的消息和检查能提供或不能提供什么，密码学被视为理想，且不涉及任何实现。

分析者指出的本次分析未覆盖的内容：

- 握手飞行建模时未包含其记录层加密，因此这里没有任何结论依赖 AEAD 保护 EncryptedExtensions、Certificate、CertificateVerify 或 Finished。
- 密钥调度和 Finished MAC 是单向（私有）函数，因此即使共享秘密已知，攻击者也无法推导流量密钥；保密性回答在该方向上偏保守，对获知共享秘密本身的攻击者不说明任何内容。
- 攻击者的单分量能力是对一条消息的一个分量签名，这正是一次冒充尝试所需，而不是该分量的签名预言机。
- X25519 用共享秘密的单一法则加上恢复标量的归约来建模，ML-KEM 作为符号 KEM；没有证明计算安全性，也没有证明任何后量子安全性。
- PKI 不在范围内：客户端固定复合公钥，证书链、撤销和名称绑定均未建模。
- HelloRetryRequest、PSK 恢复、0-RTT、客户端认证、密钥更新、记录层、告警和消息顺序的强制执行均未建模。
- 考虑了任意数量的客户端和服务器会话，但会话之间不携带状态（无恢复秘密、无票据）。

每条性质仍有哪些未决之处，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、能否直接使用，以及其价值如何。

A02: 每次接受背后都有实时服务器运行

客户端接受服务器 CertificateVerify 并完成握手时，存在一个以该客户端为意图对端的服务器运行。

只有当客户端持有服务器用协商方案、在固定证书密钥下对本次连接转录所做的签名时，客户端才接受 CertificateVerify。这一性质在任一单分量密钥、会话流量密钥和 X25519 标量泄露后仍然成立。

- 结果：成立；是否回答了所问：是；可在有保留条件的情况下使用
- 价值中：这是对草案自身要求的证明，其中明确指出了攻击者；保留条件限定了它的价值。
- 文稿中已陈述：是。文稿中的位置：s0019、s0020、s0406
- 依据：模型中握手消息流未加密，因此攻击者可以自由到达证书检查，证明依赖复合签名而非记录层。；密钥调度是单向的，因此攻击者即使扮演服务器也无法计算 Finished 值。；客户端固定复合公钥；未建模证书链。
- 待解决：在此模型中无法让攻击者通过 Finished 检查，因此客户端的完成未证明受到同一论证的保护。

A03: 对等方、方案、秘密、记录的一致性

客户端接受服务器认证时，存在匹配的服务器运行，双方身份、协商签名方案、密钥交换共享秘密与转录一致。

当客户端完成时，服务器已针对同一新鲜转录值、同一拼接的 X25519MLKEM768 秘密和同一握手流量秘密发送了其 Finished，因此双方在方案和密钥上达成一致。

- 结果：成立；是否回答了所问：是；可在有保留条件的情况下使用
- 价值中：表明转录与密钥协商按草案所依赖的方式绑定在一起。
- 文稿中已陈述：是。文稿中的位置：s0357、s0410、s0561
- 依据：共享秘密是两个分量秘密的拼接，其中 X25519 作为普通密钥协商。；只有客户端的完成与服务器的 Finished 相关联；服务器在客户端 Finished 之后的自身完成不在此主张范围内。

A04: 每次服务器运行仅一次接受

客户端每次接受服务器认证，都对应一个不同的服务器运行；同一服务器运行不能被两次接受。

每次接受都对应服务器针对同一新鲜转录生成的一个不同 CertificateVerify，包括攻击者能对已签名转录生成不同有效签名的情况。为某一连接制作的签名不会在另一连接中被接受，因为转录包含客户端的新鲜值。

- 结果：成立；是否回答了所问：是；可用
- 价值中：这正是草案为说明失去 SUF-CMA 在 TLS 中不会造成损失而给出的论证，现已机械化展示。
- 文稿中已陈述：是。文稿中的位置：s0046
- 依据：客户端的新鲜值是每次客户端运行的新鲜名称，转录哈希基于该新鲜值和两个密钥份额计算。

A08: 方案、密钥、上下文与记录的绑定

客户端接受的 CertificateVerify 所用签名方案、证书公钥类型、context string 与转录，均对应客户端提议与当前连接上下文。

客户端接受的方案是服务器签名且客户端提议的方案；任何 RSA-PKCS1-v1_5 复合方案、独立的 ML-DSA 以及 TLS 1.2 连接都不会被接受。使用同一密钥在其他协议上下文中生成的复合签名不会通过验证。

- 结果：成立；是否回答了所问：是；可在有保留条件的情况下使用
- 价值中：绑定精确成立，但排除独立回退的策略是模型的假设，而非草案所修复的内容。
- 文稿中已陈述：是。文稿中的位置：s0017、s0023、s0025、s0026、s0417、s0423、s0425
- 依据：客户端策略由模型固定：它只提议复合方案，因此独立 ML-DSA 回退是针对已选择该策略的依赖方的回答。；其他协议被建模为使用同一密钥在不同 context string 下对攻击者选择的消息签名。
- 待解决：模型中的客户端只提议 TLS 1.3，因此 TLS 1.2 连接在构造上就被拒绝，而不是通过对接收消息的单独检查来拒绝。

A08.10: 签名者角色、上下文与记录绑定

客户端接受的 CertificateVerify 的签名者角色、context string 与转录，均对应本连接的真实上下文。

客户端接受的 CertificateVerify 在服务器 context string 下覆盖本连接的转录，该转录由客户端的新鲜值和两个密钥份额构建。密钥持有者在其他协议上下文中生成的复合签名永远不会作为 TLS CertificateVerify 通过验证。

- 结果：成立；是否回答了所问：是；可在有保留条件的情况下使用
- 价值中：机械化展示空的复合 context string 在 TLS 中之所以安全，仅因为 TLS context string 和转录位于被签名输入之内。
- 文稿中已陈述：是。文稿中的位置：s0017、s0410、s0417
- 依据：其他上下文签名是使用不同 context string 对同一消息生成的，因此隔离完全依赖 context string 和转录，这正是草案所述依据。

A08.11: 方案与证书密钥兼容

客户端接受的签名方案与证书公钥类型兼容；不匹配时连接被拒绝而非接受。

客户端接受的证书具有与协商复合方案完全相同的密钥类型；其他密钥类型的证书永远不会被接受，因此方案与密钥不会错误配对。

- 结果：成立；是否回答了所问：是；可在有保留条件的情况下使用
- 价值中：确认草案要求的检查存在，且在此流程中不可绕过。
- 文稿中已陈述：是。文稿中的位置：s0020、s0387、s0425
- 依据：密钥类型是模型证书携带的值；真实绑定是证书编码，模型用常量表示。

K01: 密钥对网络的保密性

攻击者不能获得服务器长期签名私钥，也不能获得诚实会话派生的握手与流量密钥。

网络攻击者无法推导诚实会话的握手流量秘密或应用流量秘密，且复合长期密钥在会话期间不会整体被获得。即使 CRQC 恢复 X25519 标量，流量秘密仍无法获得，因为仍需要 ML-KEM 秘密。

- 结果：成立；是否回答了所问：是；可在有保留条件的情况下使用
- 价值中：X25519 被攻破的一半是有趣的部分：混合方案在一个分量失效时仍保持密钥秘密。
- 文稿中已陈述：是。文稿中的位置：s0406、s0489、s0561
- 依据：模型中密钥调度是单向的，因此此证明是保守的，对共享秘密泄露不作任何说明。；ML-KEM 和 X25519 分量是符号化的；不声称计算安全性或量子安全性。
- 待解决：未询问共享秘密本身的泄露，因此对获知该秘密的攻击者不作任何说明。

C01: 流量密钥的前向保密

会话完成后泄露服务器长期签名私钥，不暴露此前完成会话的握手与流量密钥。

会话后泄露的服务器长期签名密钥不会暴露该会话的应用流量秘密，因为密钥协商是临时的，长期密钥不是其输入之一。一旦复合签名密钥被攻破，过往流量仍保持机密。

- 结果：成立；是否回答了所问：是；可在有保留条件的情况下使用
- 价值低：这直接来自设计，因为签名密钥不是密钥调度的输入；草案已在此基础上规划迁移。
- 文稿中已陈述：是。文稿中的位置：s0045、s0047
- 依据：泄露建模在会话之后；此后开始的会话不在此主张范围内。；模型中密钥调度是单向的。
- 待解决：未询问泄露后的握手流量秘密；只询问了应用流量秘密。

C08: 认证在一个组件损坏后仍成立

在传统分量签名私钥或 ML-DSA 分量签名私钥之一被攻破时，服务器认证目标（客户端接受真实服务器）仍成立。

恰好一个分量私钥被攻破时，无论二者中哪一个，客户端仍只在服务器为本连接转录签名时才接受服务器 CertificateVerify。一个分量被攻破不会破坏 TLS 服务器认证，这正是草案的纵深防御主张。

- 结果：成立；是否回答了所问：是；可在有保留条件的情况下使用
- 价值中：这是草案的核心安全主张，现已在明确攻击者下机械化展示；关于攻击者能力和密钥调度的保留条件使其尚不构成完整证明。
- 文稿中已陈述：是。文稿中的位置：s0045、s0046、s0048
- 依据：损坏分量攻击者获得对其所选签名输入的一个分量签名，这正是单次冒充尝试所需的；未赋予该分量无界签名预言机。；握手消息流未加密，因此攻击者确实能到达证书检查。；密钥调度是单向的，因此攻击者即使拥有有效 CertificateVerify 也无法完成握手。
- 待解决：未询问关于损坏分量签名预言机而非单个签名的陈述。

C08.9: 一个组件泄露后认证仍成立

在组合签名的一个分量私钥泄露时，客户端仍只接受真实服务器的认证。

当泄露的是会话期间的分量密钥而非会话之前时，同一结论仍成立：客户端仍只接受服务器为该转录生成的 CertificateVerify。

- 结果：成立；是否回答了所问：是；可在有保留条件的情况下使用
- 价值中：陈述了草案讨论的会话内泄露形式，条件与 C08 相同。
- 文稿中已陈述：是。文稿中的位置：s0045、s0048
- 依据：与 C08 相同的考虑：对一条消息的一个分量签名、未加密的消息流、单向密钥调度。

R05: 方案或版本不可降级

客户端接受的签名方案与协议版本满足草案规定的最低策略，攻击者不能通过操纵协商使接受配置降级。

客户端对方案、证书密钥类型和协议版本的检查都已到位：如果 CertificateVerify 指名了受限方案、证书属于其他密钥类型，或者连接是 TLS 1.2，客户端会拒绝，而不是接受。只有在模型所假定的策略下，独立的 ML-DSA 回退才会被排除。

- 结果：成立；回答了所问：是；可用但有保留
- 价值中：拒绝检查得到确认，但降级答案的可靠性取决于模型固定的策略。
- 文稿中有说明：是。文稿中的位置：s0023、s0025、s0026、s0052
- 依据：拒绝独立 ML-DSA 的策略是模型的选择，正如草案所说，这属于依赖方策略问题。；版本在 ServerHello 上检查；客户端只提供 TLS 1.3。
- 未决：模型将依赖方策略固定为仅复合；采用其他策略的部署未作分析。

X01: 仅传统回退冒充

一个同时接受仅传统服务器认证的客户端，在面对持有传统组件私钥的攻击者时，会接受一次并未发生的服务器运行；而要求复合证书的客户端则始终拒绝该攻击者。

一个同时接受仅传统服务器证书的客户端，会被只持有传统组件密钥的攻击者冒充：攻击者对本连接的签名输入签名，出示传统证书，并在完全没有服务器运行的情况下被接受。要求复合证书的客户端始终拒绝同一攻击者，因此差异在于客户端愿意接受的证书密钥类型。

- 结果：攻击成立（文稿中有说明）；回答了所问：是；可用但有保留
- 价值中：草案说明了暴露面；模型以确切条件和与复合客户端的确切差异复现了它。
- 文稿中有说明：是。文稿中的位置：main:558、s0048、s0052
- 依据：仅传统客户端以明文证书检查建模，且没有 Finished 交换；该主张针对证书密钥类型和组件签名，而非记录层。；攻击者获得一个对其所选签名输入的传统组件签名，这正是持有该组件密钥所赋予的能力。
- 未决：缓存先前后量子能力的客户端（草案所指的连续性机制）会怎样，未作分析。

附录 B：复现的已知结论

文稿已经说明或已搁置的不成立性质，不算发现。以下属于此类：分析在模型上复现了它，并给出了确切条件。

B.1 同时接受仅传统证书的客户端会被传统组件密钥冒充

类型	协议规范本身的弱点
文稿出处	main:558
价值	中
涉及的性质	X01 （见第 3 节）
原文	s0048, s0052
依据	ProVerif attack query `x01_fallback`

发生了什么。 接受仅传统服务器证书的客户端会被只持有传统组件密钥的攻击者冒充。攻击者基于该客户端自身的新鲜值构造 TLS 签名输入，获得一个传统组件签名，发送密钥类型为仅传统的证书，客户端便接受。在同一模型中，要求复合证书的客户端始终拒绝同一攻击者。

攻击者需要什么。 服务器的传统组件私钥（拥有 CRQC 的攻击者可以获取，s0048），以及一个同时接受仅传统服务器认证的客户端。

攻击者能得到什么。 攻击者对该客户端而言成为该服务器：它被接受，推导出客户端推导出的密钥，客户端相信它连到了传统证书中所指的服务器。

拟议修改。 移除较弱的接受路径：客户端必须只提供复合签名方案，要求证书密钥类型为协商的复合密钥类型，并验证 TLS 签名输入上的两个组件签名。该修改已放入模型副本并检查：冒充不再存在，正常流程仍可找到，其他所有记录的答案不变。

用 ProVerif 复核。 该修改已放入模型副本：原先接受仅传统服务器认证的客户端不再接受：它只提供复合签名方案，要求证书密钥类型为协商的复合密钥类型，并验证 TLS 签名输入上的两个组件签名。其他不变。在修改后的模型上，ProVerif 证明 `x01_fallback`：该攻击在那里不再可能。协议的正常流程仍可找到。其他 17 个记录的答案不变。这是对模型的首轮检查：它表明该修改在那里移除了此攻击，而非表明它不会引入其他问题。修改后的模型：`records/fixes/0072-fix-fallback.pv`。

B.2 一个组件失效时，复合验证仍保持服务器认证

类型	已确认的保证
文稿出处	main:527
价值	中
涉及的性质	C08, C08.9, A02 （见第 3 节）
原文	s0045, s0046

发生了什么。 在恰好一个组件私钥被攻破（传统或 ML-DSA），且攻击者拥有一个对其所选签名输入的一个组件签名的情况下，攻击者始终无法让客户端接受一个服务器未为该连接签名的 CertificateVerify。它缺少的是同一转录上的第二个组件签名，且无法从其他连接移来，因为转录携带客户端的新鲜值。

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，以及每个攻击及分析者的判定。

目标	级别	界	查询	对照	攻击	主张
A02	proved		a02_agree	checked		6 proved
A03	proved		a03_agree	checked		1 proved
A04	proved		a04_inj	checked		2 proved
A08	proved		a02_agree	checked		4 proved
K01	proved		k01_hs	checked		6 proved
C01	proved		c01_fs	checked		1 proved
C08	proved		a02_agree	checked		3 proved
R05	proved		a08_pkcs15	checked		3 proved
C08.9	proved		a02_agree	checked		2 proved
A08.10	proved		a02_agree	checked		2 proved
A08.11	proved		a08_keytype	checked		1 proved
X01	none				x01_fallback (holds_in_source)	1 attack

逐条主张，按分析者对每条主张的措辞；每条答案来自证明器：

目标	主张	攻击者	回答	查询	对照
A02	客户端接受的每个 CertificateVerify 都由服务器使用协商出的组合方案和固定的证书密钥类型，针对本连接的握手记录签名；攻击者仅拥有网络。	`network`	proved	a02_agree	checked
A02	当攻击者还持有组合证书的传统分量私钥并能用该分量签名时，客户端仍然只接受由服务器签名的 CertificateVerify。	`comp_traditional`	proved	a02_agree	checked
A02	当攻击者还持有 ML-DSA 分量私钥并能用该分量签名时，客户端仍然只接受由服务器签名的 CertificateVerify。	`comp_mldsa`	proved	a02_agree	checked
A02	当攻击者还能针对服务器已经签过名的握手记录生成另一个有效的组合签名时，客户端仍然只接受由服务器签名的 CertificateVerify。	`suf_cma`	proved	a02_agree	checked
A02	当攻击者还能用 CRQC 恢复 X25519 临时标量时，客户端仍然只接受由服务器签名的 CertificateVerify。	`crqc_x25519`	proved	a02_agree	checked
A02	当攻击者还能读取该会话的握手流量密钥和应用流量密钥时，客户端仍然只接受由服务器签名的 CertificateVerify。	`session_state`	proved	a02_agree	checked
A03	当客户端完成握手时，服务器已针对同一个新鲜握手记录值、同一个 X25519MLKEM768 共享密钥和同一个握手流量密钥发送了它的 Finished；攻击者仅拥有网络。	`network`	proved	a03_agree	checked
A04	客户端每次接受 CertificateVerify，都对应服务器针对同一个新鲜握手记录给出的一个不同 CertificateVerify，因此为一条连接生成的签名不会在另一条连接中被接受；攻击者仅拥有网络。	`network`	proved	a04_inj	checked

目标	主张	攻击者	回答	查询	对照
A04	即使攻击者能针对已经签过名的握手记录生成另一个有效签名，仍然每次服务器运行只对应一次接受，因为握手记录中带有每条连接各自的新鲜客户端值。	`suf_cma`	proved	a04_inj	checked
A08	客户端永远不会接受任何指名 RSA-PKCS1-v1_5 组合方案的 CertificateVerify。	`network`	proved	a08_pkcs15	checked
A08	要求使用组合方案的客户端永远不会接受任何指名独立 ML-DSA 的 CertificateVerify。	`network`	proved	a08_sa	checked
A08	客户端接受的 CertificateVerify 中指名的签名方案，就是服务器签名所用的方案，也是客户端所提供的方案；攻击者仅拥有网络。	`network`	proved	a02_agree	checked
A08	用同一密钥在另一协议上下文中生成的组合签名永远不会被接受：被接受的 CertificateVerify 携带服务器的 CertificateVerify 上下文串和本连接的握手记录。	`cross_protocol`	proved	a02_agree	checked
K01	攻击者无法推导出诚实会话的握手流量密钥；攻击者仅拥有网络。	`network`	proved	k01_hs	checked
K01	攻击者无法推导出诚实会话的应用流量密钥，尽管该密钥用于保护线上的数据记录；攻击者仅拥有网络。	`network`	proved	k01_app	checked
K01	攻击者在会话期间不会获得完整的组合长期签名密钥（两个分量私钥）；攻击者仅拥有网络。	`network`	proved	k01_ltk	checked
K01	当攻击者恢复 X25519 临时标量时，握手流量密钥仍然无法获得，因为 X25519MLKEM768 密钥还包含 ML-KEM 密钥。	`crqc_x25519`	proved	k01_hs	checked
K01	当攻击者恢复 X25519 临时标量时，应用流量密钥仍然无法获得，因为仍然需要共享密钥中的 ML-KEM 那一半。	`crqc_x25519`	proved	k01_app	checked
K01	当攻击者恢复 X25519 临时标量时，仍然无法获得完整的组合长期签名密钥。	`crqc_x25519`	proved	k01_ltk	checked
C01	会话完成且服务器的两个分量私钥都已泄露之后，攻击者仍然无法推导出该会话的应用流量密钥。	`ltk_after_session`	proved	c01_fs	checked
C08	在仅持有一个分量私钥（传统分量）并具备用其签名能力的情况下，客户端仍然只接受服务器针对该握手记录生成的 CertificateVerify。	`comp_traditional`	proved	a02_agree	checked
C08	在仅持有一个分量私钥（ML-DSA 分量）并具备用其签名能力的情况下，客户端仍然只接受服务器针对该握手记录生成的 CertificateVerify。	`comp_mldsa`	proved	a02_agree	checked
C08	读取会话的握手流量密钥和应用流量密钥并不能让攻击者生成可接受的 CertificateVerify。	`session_state`	proved	a02_agree	checked
R05	不会接受任何指名 RSA-PKCS1-v1_5 组合方案的 CertificateVerify，因此草案将这些方案限制在证书扩展中的规定成立。	`network`	proved	a08_pkcs15	checked
R05	当攻击者操纵协商时，策略要求组合认证的客户端永远不会回退到独立 ML-DSA。	`network`	proved	a08_sa	checked
R05	本文件所述组合方案的任何 CertificateVerify 都不会在 TLS 1.2 连接中被接受：只有当所选版本为 TLS 1.3 时客户端才会继续。	`network`	proved	a08_ver12	checked
C08.9	会话期间传统分量私钥泄露后，客户端仍然只接受服务器针对该握手记录生成的 CertificateVerify。	`comp_traditional`	proved	a02_agree	checked

目标	主张	攻击者	回答	查询	对照
C08.9	会话期间 ML-DSA 分量私钥泄露后，客户端仍然只接受服务器针对该握手记录生成的 CertificateVerify。	`comp_mldsa`	proved	a02_agree	checked
A08.10	客户端接受的 CertificateVerify 覆盖本连接的握手记录哈希，该哈希由客户端的新鲜值和双方的密钥份额构成；攻击者仅拥有网络。	`network`	proved	a02_agree	checked
A08.10	服务器密钥持有者在另一协议上下文中生成的组合签名无法作为 TLS CertificateVerify 通过验证，因为签名输入携带服务器的 CertificateVerify 上下文串和本连接的握手记录。	`cross_protocol`	proved	a02_agree	checked
A08.11	客户端接受的证书的密钥类型等于协商出的组合方案；其他密钥类型的证书永远不会被接受。	`network`	proved	a08_keytype	checked
X01	当攻击者持有传统分量私钥时，同时接受仅传统服务器认证的客户端会接受一次从未发生过的服务器运行。	`comp_traditional`	attack (holds_in_source)	x01_fallback (the control of a02_agree)	

附录 D：分析是怎么跑的

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 节的每个结果都是证明器给出的答案。

ProVerif+ Agent 会话 session-20261114T230251，2026-11-14。状态：已完成。模型：deepseek-v4-1-flash-261017。步数：74，证明器 78 秒，实际耗时 22 分钟，费用 0.201 USD。

上下文：宿主从对话中移除了过期内容 1 次，共 498169 个字符（较早的推理，以及模型在最近一次修改之前所看到的内容）。轨迹文件保留了一切。

ProVerif：运行了 153 次证明，147 次有结论，6 次无答案，34 秒，记录 57 个答案。

分析了什么

协议	TLS composite ML-DSA authentication
版本	draft-ietf-tls-composite-mldsa-00
分析范围	固定草案中 ML-DSA+Ed25519 组合签名和 X25519MLKEM768，服务器认证完整 1-RTT；每分量和正确 TLS context 都验证，实际签名方案以草案为准；排除 PKI 构建、HRR/PSK/0-RTT。
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	564 numbered passages (149815 bytes) from the sections listed below. A reference such as s0001 in this report is one of these passages

分析所依据的文稿章节：

文稿	章节	标题
main	2	ML-DSA SignatureSchemes
main	3	Signature Algorithm Restrictions
main	4	Selection Criteria for Composite Signature Algorithms
main	4.1	Mapping TLS SignatureSchemes to Composite ML-DSA
main	5	Security Considerations
rfc9846	4	Handshake Protocol
rfc9846	4.1	The Transcript Hash
rfc9846	4.2	Key Exchange Messages
rfc9846	4.2.1	Cryptographic Negotiation
rfc9846	4.2.2	Client Hello
rfc9846	4.2.3	Server Hello
rfc9846	4.2.4	Hello Retry Request
rfc9846	4.3	Extensions
rfc9846	4.3.1	Supported Versions
rfc9846	4.3.2	Cookie
rfc9846	4.3.3	Signature Algorithms
rfc9846	4.3.4	Certificate Authorities
rfc9846	4.3.5	OID Filters
rfc9846	4.3.6	Post-Handshake Certificate-Based Client Authentication
rfc9846	4.3.7	Supported Groups

文稿	章节	标题
rfc9846	4.3.8	Key Share
rfc9846	4.3.8.1	Diffie-Hellman Parameters
rfc9846	4.3.8.2	ECDHE Parameters
rfc9846	4.3.9	Pre-Shared Key Exchange Modes
rfc9846	4.3.10	Early Data Indication
rfc9846	4.3.11	Pre-Shared Key Extension
rfc9846	4.3.11.1	Ticket Age
rfc9846	4.3.11.2	PSK Binder
rfc9846	4.3.11.3	Processing Order
rfc9846	4.4	Server Parameters
rfc9846	4.4.1	Encrypted Extensions
rfc9846	4.4.2	Certificate Request
rfc9846	4.5	Authentication Messages
rfc9846	4.5.1	Certificate
rfc9846	4.5.1.1	OCSP Status and SCT Extensions
rfc9846	4.5.1.2	Certificate Selection
rfc9846	4.5.1.3	Receiving a Certificate Message
rfc9846	4.5.2	Certificate Verify
rfc9846	4.5.3	Finished
rfc9846	4.6	End of Early Data
rfc9846	4.7	Post-Handshake Messages
rfc9846	4.7.1	New Session Ticket Message
rfc9846	4.7.2	Post-Handshake Authentication
rfc9846	4.7.3	Key and Initialization Vector Update
rfc9846	7	Cryptographic Computations
rfc9846	7.1	Key Schedule
rfc9846	7.2	Updating Traffic Secrets
rfc9846	7.3	Traffic Key Calculation
rfc9846	7.4	Asymmetric Shared Secret Calculation
rfc9846	7.4.1	Finite Field Diffie-Hellman
rfc9846	7.4.2	Elliptic Curve Diffie-Hellman
rfc9846	7.5	Exporters
rfc10024	4	Negotiated Groups
rfc10024	4.1	Client Share
rfc10024	4.2	Server Share
rfc10024	4.3	Shared Secret

协议的目的，即原文步骤在文稿中定位到的内容：固定草案中 ML-DSA+Ed25519 组合签名和 X25519MLKEM768，服务器认证完整 1-RTT；每分量和正确 TLS context 都验证，实际签名方案以草案为准；排除 PKI 构建、HRR/PSK/0-RTT。

对照文稿定性质

在编写查询之前，分析者把起草的性质清单与文稿自身所述协议的目的进行了对照。名称、性质涉及的内容以及文稿所述的内容均为分析者的解读；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	每次接受背后都有实时服务器运行	客户端确信：它接受其 CertificateVerify 的服务器持有被固定复合证书的私钥，并且运行了本次握手。	对等方 MUST 用协商出的复合方案验证 CertificateVerify 签名（s0019）；CertificateVerify 明确证明端点持有其证书的私钥，并为迄今的握手提供完整性（s0406）；终端实体证书 MUST 携带与协商方案兼容的公钥（s0020）。	s0019, s0020, s0406
A03	对等方、方案、秘密、记录的一致性	当客户端接受服务器时，客户端与服务器对两个身份、协商出的复合 SignatureScheme、X25519MLKEM768 共享秘密以及握手记录达成一致。	CertificateVerify 是对 Transcript-Hash(Handshake Context, Certificate) 的签名（s0357, s0409, s0410），使用协商出的复合方案生成（s0018）；对 X25519MLKEM768 而言，共享秘密是 ML-KEM 秘密与 X25519 秘密的拼接（s0561）。	s0018, s0357, s0410, s0561
A04	每次服务器运行仅一次接受	每次对服务器认证的接受都对应一次不同的服务器运行：一个只签名一次的 CertificateVerify 无法被重放到另一条连接中，即使攻击者能对已签名的记录生成另一个有效签名也是如此。	每条 TLS 连接中签名的握手记录都是唯一的，因此冒充需要对一条新连接的记录伪造签名；失去 SUF-CMA 只会让攻击者能对已经签过名的记录生成另一个有效签名，而这无法在另一条连接中复用（s0046）。	s0046, s0357
A08	方案、密钥、上下文与记录的绑定	CertificateVerify 中声明的签名方案、证书的密钥类型、TLS 上下文字符串以及被检查的记录，都与客户端所提议的内容以及本条连接相对应。	签名与验证都 MUST 使用空的复合上下文字符串，TLS 通过协议特定的上下文字符串加记录提供域分离（s0017）；服务器签名上下文字符串将不同上下文的签名分开，以抵御跨协议攻击（s0417）；算法 MUST 是客户端提议过的（s0423），并与证书密钥兼容（s0425）。	s0017, s0417, s0423, s0425
K01	密钥对网络的保密性	服务器的复合长期签名私钥，以及一次诚实会话的握手流量秘密和应用流量秘密，都对拥有网络的攻击者保密，也对额外用 CRQC 恢复出密钥协商中 X25519 那一半的攻击者保密。	CertificateVerify 证明持有私钥而不泄露私钥（s0406）；握手把输入秘密与记录结合成工作密钥材料（s0489, s0499）；对 X25519MLKEM768 而言，共享秘密拼接了 ML-KEM 秘密与 X25519 秘密，因此一半被攻破不会暴露它（s0561, rfc10024:299）。	s0406, s0489, s0561
C01	流量密钥的前向保密	会话完成之后服务器长期复合签名私钥的泄露，不会暴露该会话的握手流量密钥和应用流量密钥。	草案论证：退回单一安全组件可在过渡期保持认证（s0045），并且一旦 CRQC 出现，复合方案无需停用（s0047）；TLS 前向保密限制了单个方向密钥泄露的影响（rfc9846:6545, rfc9846:6562）。	s0045, s0047, s0489
C08	认证在一个组件损坏后仍成立	当服务器复合证书中恰好一个组件私钥（传统或 ML-DSA）被攻破时，要求复合认证的客户端仍然只接受真实服务器。	传统组件贡献纵深防御，只要至少一个组件算法仍然安全，就能维持 TLS 认证安全（s0045）；CRQC 让攻击者能推导出传统私钥，但仅此并不能击败复合认证（s0048）。	s0045, s0046, s0048

ID	简称	对象	文稿怎么说	出处
R05	方案或版本不可降级	客户端接受 CertificateVerify 时所依据的签名方案和协议版本符合草案的策略：CertificateVerify 中不使用 RSA-PKCS1-v1_5 复合方案，本文件的方案不在 TLS 1.2 中使用，不回退到独立的 ML-DSA。	pkcs15 复合方案 MUST NOT 与 "signature_algorithms" 一起使用，且未为已签名握手消息定义；收到此类 CertificateVerify 的对等方 MUST 中止（s0025, s0026）；这些方案 MUST NOT 在 TLS 1.2 中使用（s0023）；防止降级到独立 ML-DSA 属于依赖方策略问题（s0052）。	s0023, s0025, s0026, s0052
C08.9	一个组件泄露后认证仍成立	当复合证书的一个组件私钥在会话期间泄露时，客户端仍然只接受真实服务器。	只要至少一个组件算法仍然安全，认证安全就能维持（s0045）；仅传统组件被攻破并不能击败复合认证（s0048）。	s0045, s0048
A08.10	签名者角色、上下文与记录绑定	客户端接受的 CertificateVerify 是由本条连接的服务器、使用服务器上下文字符串、针对本条连接的记录生成的，而不是在其他角色或协议上下文中生成的。	服务器签名的上下文字符串是 "TLS 1.3, server CertificateVerify"，用于将不同上下文中生成的签名分开，以抵御跨协议攻击（s0417）；复合上下文字符串 MUST 为空，因为 TLS 提供域分离（s0017）；签名覆盖 Transcript-Hash(Handshake Context, Certificate)（s0410）。	s0017, s0410, s0417
A08.11	方案与证书密钥兼容	客户端接受的签名方案与终端实体证书中的公钥兼容，不匹配会被拒绝而不是被接受。	终端实体证书 MUST 包含与协商出的 SignatureScheme 兼容的公钥（s0020）；证书密钥 MUST 能用于对等方 signature_algorithms 中的某个方案（s0387），并且算法 MUST 与证书密钥兼容（s0425）。	s0020, s0387, s0425
X01	仅传统回退冒充	客户端使用携带仅传统组件密钥的证书（而非复合证书）接受服务器认证。	使用传统证书接受服务器认证的客户端即使也支持复合认证，仍然脆弱：一旦 CRQC 可用，攻击者就能推导出对等方证书中传统公钥对应的私钥，并冒充该对等方（s0048）；反方向的同一问题属于依赖方策略问题（s0052）；剥离一个组件之所以可行，是因为传统组件能单独通过验证（rfc10073:2226）。	main:558, main:585, main:596, s0048, s0052

分析者依据文稿补充的性质：X01。

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 72 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 25 段，没选 47 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:509-510	没有陈述目标	指向 [I-D.ietf-lamps-pq-composite-sigs] 第 11 节；其实质内容是 RFC 10073 第 9 节的复合签名考虑事项，已在 P19..P46 逐段对应。
P02	main:520-520	没有陈述目标	草案页脚，无内容。
P03	main:524-524	没有陈述目标	草案页眉，无内容。
P04	main:527-533	C08, C08.9, `comp_traditional`, `comp_mldsa`	
P05	main:535-551	A04, `suf_cma`	
P06	main:553-556	C01, `ltk_after_session`	

段落	出处	定为	不选的原因
P07	main:558-571	X01, `comp_traditional`	
P08	main:578-578	没有陈述目标	草案页脚，无内容。
P09	main:582-582	没有陈述目标	草案页眉，无内容。
P10	main:585-594	X01, `comp_traditional`	
P11	main:596-604	R05	
P12	rfc10024:268-271	A03, A08.10, `cross_protocol`	
P13	rfc10024:273-276	证明器问不了	实现侧的侧信道抗性；证明器把原语建模为理想原语，没有实现行为导致的时序或泄漏的概念。
P14	rfc10024:278-280	没有陈述目标	说明这些群使用定长公钥、密文和共享秘密；这是编码的格式性质。
P15	rfc10024:282-287	证明器问不了	涉及 ML-KEM 封装所用随机数及其携带的关于生成器的信息；符号模型把新鲜随机数视为理想。
P16	rfc10024:289-297	证明器问不了	RNG 状态攻击需要弱随机数生成器；无法向证明器询问 RNG 的强度。
P17	rfc10024:299-306	K01, `crqc_x25519`	
P18	rfc10024:308-310	证明器问不了	两个算法共用同一个不安全 RNG 的状态；这是 RNG 质量问题，在符号模型中无法询问。
P19	rfc10073:2083-2084	没有陈述目标	指向 [RFC9881]：ML-DSA 组件自身的安全考虑事项，属于计算性问题，无法询问；复合层面的后果已在 P23..P39 对应。
P20	rfc10073:2088-2090	C08, C08.9, `comp_traditional`, `comp_mldsa`	
P21	rfc10073:2092-2109	A02, C08, C08.9, `comp_traditional`, `comp_mldsa`	
P22	rfc10073:2111-2126	C08, `comp_traditional`, `comp_mldsa`	
P23	rfc10073:2130-2142	证明器问不了	禁止在复合密钥与非复合密钥之间复用组件密钥材料，以及 X.509 信任假设，属于密钥管理和 PKI 配置事项，不在一次握手运行之内；一次运行能展示的部分，即签名与其上下文绑定，是目标 A08.10。
P24	rfc10073:2144-2148	C08, C08.9, `comp_traditional`, `comp_mldsa`	
P25	rfc10073:2152-2156	没有陈述目标	EUF-CMA 的定义，不是本协议的性质。
P26	rfc10073:2158-2164	A02, C08, `comp_traditional`, `comp_mldsa`	
P27	rfc10073:2166-2167	C08, C08.9, `comp_traditional`, `comp_mldsa`	
P28	rfc10073:2169-2170	C08, C08.9, `comp_traditional`	
P29	rfc10073:2172-2173	C08, C08.9, `comp_mldsa`	
P30	rfc10073:2175-2180	不在范围内	关于伪造 Composite ML-DSA 签名的 X.509 证书，即证书签发和证书路径，任务已排除；握手签名对应的同一双组件性质是目标 C08。

段落	出处	定为	不选的原因
P31	rfc10073:2184-2189	没有陈述目标	SUF-CMA 的定义，不是本协议的性质。
P32 至 P35	rfc10073:2191-2212	A04, `suf_cma`	
P36	rfc10073:2216-2219	没有陈述目标	弱不可分离性的定义，不是本协议的性质。
P37	rfc10073:2221-2224	没有陈述目标	强不可分离性的定义，不是本协议的性质。
P38	rfc10073:2226-2239	A08.10, X01, `comp_traditional`	
P39	rfc10073:2241-2248	A08.11	
P40	rfc10073:2252-2259	证明器问不了	生成密钥的最佳实践；属于密钥管理事项，无法向证明器询问。
P41	rfc10073:2261-2273	A08.10, X01, `cross_protocol`, `comp_traditional`	
P42	rfc10073:2275-2286	不在范围内	由证书颁发机构进行吊销检查，属于任务排除的 PKI 部分。
P43	rfc10073:2288-2294	A08.10	
P44	rfc10073:2298-2305	A08.10, A08	
P45	rfc10073:2309-2313	不在范围内	弃用算法和 OID 的组织政策；属于管理流程，不是握手的性质。
P46	rfc10073:2315-2326	不在范围内	已弃用组件算法的物料清单和模块政策；属于握手之外的组织事项。
P47 至 P71	rfc5280:5226-5449	不在范围内	RFC 5280 第 8 节整体讲证书和 CRL 概要、CA 实践、吊销、名称比较和路径验证；任务排除 PKI 构造，其中没有任何内容陈述本范围内 TLS 握手的性质。
P72	rfc9846:4676-4677	没有陈述目标	指向 RFC 9846 附录 C、E 和 F；其中在范围内的考虑事项（转录与认证、前向保密）已在 A03、A08.10 和 C01 的依据中引用。

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 34 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 9 条，没检查 25 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	K01	
K03	强保密／秘密值不可区分性	证明器问不了	强保密要求攻击者区分两个允许的值；这里的应用流量密钥来自新鲜临时随机数，而不是从两个候选值中选出，因此没有可区分的值对，noninterf 无法实例化。K01 是最接近、可询问的性质。
K04	密钥用途、方向与会话分离	不在范围内	密钥用途、方向与 epoch 分离属于 TLS 1.3 的密钥编排（s0488 至 s0513），任务将其排除在范围之外：模型从 X25519MLKEM768 共享秘密推导出一个握手密钥和一个应用密钥，而不是方向与 epoch 的层级结构。
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	

条目	目录里的性质	作为哪条性质检查	不检查的原因
A06	显式密钥确认	A03	
A07	身份—凭据—密钥绑定	不在范围内	将身份绑定到凭据和密钥属于证书链与 PKI 的工作，任务将其排除；模型固定了服务器的组合公钥，既不构建也不验证证书链。
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	该协议不承载计费、控制或一次性令牌操作。
R04	消息接受顺序／窗口安全	不适用	所分析的流程不携带接受窗口、序列号或计数器：握手消息按固定顺序交换一次。
C02	抗密钥泄露冒充（KCI）	不适用	此流程中只有服务器经过认证；客户端在范围内不持有长期凭据，因此不存在可能因泄露而导致冒充对端的自身长期密钥。
C03	已知密钥安全／泄露隔离	C01	
C04	临时秘密／部分状态泄露韧性	C08, K01	
C05	泄露后安全恢复（PCS）	不在范围内	所分析的运行是单次完整的 1-RTT 握手，没有恢复、重新密钥或棘轮；任务排除了 PSK 恢复和 0-RTT，因此不存在可以重新获得保密性的后续会话。
C06	注册凭据泄露后的增强认证	不适用	协议中没有基于口令认证或增强注册记录。
C07	抗离线字典攻击	不适用	协议不使用口令或其他低熵秘密，因此不可能进行离线字典攻击。
Z01	授权执行正确性	不适用	TLS 1.3 认证建立身份和密钥；它不携带授权决策或受保护操作。
Z02	委托权限不扩张	不适用	未使用委托或代理机制。
Z03	授权与身份／会话／操作绑定	不适用	不传递也不绑定任何授权上下文。
Z04	撤销与过期生效	不适用	协议不携带带过期或撤销的许可。
Z05	跨租户／跨服务安全隔离	不适用	租户或多服务隔离不属于握手的一部分。
S01	加密 nonce 唯一性	不在范围内	AEAD nonce 的构造属于 TLS 1.3 记录层（s0519 至 s0529），不属于任务所分析的组合 ML-DSA 认证流程。
S02	重传／密钥重安装安全	不适用	TLS 1.3 没有重传，也没有密钥重安装消息；握手密钥每次握手只安装一次。
S03	抗回滚与状态单调性	不适用	在此范围内，握手不保留任何缓存的版本、计数器或可恢复状态。
S04	跨阶段安全上下文一致性	不在范围内	跨阶段上下文一致性涉及恢复、HelloRetryRequest 和 0-RTT，这些均被任务排除。
S05	终止与旧状态失效	不在范围内	拆除与关闭不属于任务所分析的认证流程。
P01	身份匿名性	不适用	服务器认证以明文传输服务器证书，而客户端根本不发送任何身份；协议不试图隐藏哪一方参与其中。
P02	会话不可关联性	不适用	服务器始终出示同一张证书，因此会话在设计上就可关联到该服务器。
P03	用户／凭据存在性隐藏	不适用	此流程中不查询任何用户名或凭据，因此没有需要隐藏的存在性。
V01	条件性进展／错误恢复	证明器问不了	一次运行能否总是完成或从错误中恢复属于进展性质；证明器无法被询问一次运行是否总会结束。
V02	无非预期死锁	证明器问不了	无死锁是分支时间进展性质，证明器无法被询问。
V03	受限资源滥用／抗放大	证明器问不了	除计数事件之外的资源消耗无法向证明器询问。

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 15 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 3 项，没设的 12 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	攻击者得不到更多	客户端只持有服务器的组合公钥证书和自己的临时秘密；这里的性质关注服务器向客户端保证什么，因此不诚实的客户端得不到网络攻击者没有的东西。
S02	服务器	参与方	按定义全部失守	不诚实的服务器正是客户端要认证的那一方：对它的每一条认证性质都平凡成立，而每一条保密性质按定义都已失去，且文稿没有要求在恶意服务器下成立任何性质。
S03	服务器身份	值	攻击者得不到更多	服务器身份是公开的：它就是证书中的名称，也在证书本身之中。
S04	协商签名方案	值	攻击者得不到更多	协商出的签名方案是连接的公开常量。
S05	握手转录	值	攻击者得不到更多	握手转录以明文在线上传输；攻击者已经拥有它。
S06	密钥交换共享秘密	值	按定义全部失守	一旦共享秘密已知，流量密钥性质按定义即已失去，且没有任何文稿要求在完全攻破它的情况下成立密钥协商性质；唯一记录在案的部分攻破是 CRQC 恢复 X25519 标量（rfc10024:299），即攻击者 crqc_x25519，为 K01 和 A02 设置。
S07	签名方案	值	攻击者得不到更多	签名方案标识符是公开常量。
S08	证书公钥	值	攻击者得不到更多	证书公钥是公开的；只有其私钥部分才重要，那是值 S10。
S09	context string	值	攻击者得不到更多	context string 是协议的公开常量。
S10	服务器长期签名私钥	值	`ltk_after_session`	
S11	握手流量密钥	值	`session_state`	
S12	应用流量密钥	值	`session_state`	
S13	组合签名	值	攻击者得不到更多	组合签名以明文在线上随 CertificateVerify 消息传输。
S14	协议版本	值	攻击者得不到更多	协议版本是 hello 消息的公开字段。
S15	签名者角色	值	攻击者得不到更多	签名者角色由 context string 固定，它是公开常量。

每条性质在各攻击者条件下的答案

攻击者条件由分析者设定，在编写查询之前就已确定；每个答案都在其中某个条件下记录。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”是指为该性质设置了该攻击者条件，但在其下证明器让某条主张悬而未决，或没有记录任何内容；空单元格是指分析者未为该性质设置该攻击者条件。

攻击者	持有什么	依据	出处
`network`	仅网络：可以读取、丢弃、更改并发送消息，不持有任何一方的秘密		
`comp_traditional`	服务器复合证书中传统组件（Ed25519/ECDSA）的私钥，在 CRQC 可用后获得，且只能用该组件签名	一旦 CRQC 可用，攻击者就能推导出对端证书中传统公钥对应的私钥，并冒充该对端（s0048）；传统组件只是纵深防御（s0045）。	s0045, s0048
`comp_mldsa`	服务器复合证书中 ML-DSA 组件的私钥，且只能用该组件签名	使用复合认证正是因为 ML-DSA 本身不能被完全信任，所以攻破 ML-DSA 组件不得攻破复合认证（s0052, s0045）。	s0045, s0052
`ltk_after_session`	服务器完整的复合长期签名私钥（两个组件），在所分析的会话完成之后获得	草案计划在 CRQC 到来后迁移离开复合 ML-DSA（s0045, s0047），而 TLS 前向保密限制了较晚时间密钥泄露的影响（rfc9846:6545, rfc9846:6562）。	s0045, s0047

攻击者	持有什么	依据	出处
`suf_cma`	能对诚实服务器已经签过名的消息产生另一个有效签名，即复合签名不是强不可伪造的	只有当每个组件都满足 SUF-CMA 时，复合签名才满足；ECDSA 不满足，而且一旦 CRQC 可用，除使用 RSA-PKCS1-v1_5 的方案外没有复合方案满足（s0046）；ECDSA 签名可以被轻易改成同一消息的另一个有效签名，而这会传递到复合层（rfc10073:2191）。	s0046
`cross_protocol`	在另一协议或角色上下文中使用同一长期密钥生成的复合 ML-DSA 签名，通过密钥重用或跨协议交互获得	禁止在复合上下文与单算法上下文之间重用密钥，因为这会导致剥离攻击以及进一步的跨协议风险（rfc10073:2261, rfc10073:2272）；TLS 通过协议上下文字符串加转录来分隔签名上下文（s0017, s0417）。	s0017, s0417
`crqc_x25519`	已完成会话的 X25519 临时标量，因而也持有 X25519MLKEM768 共享秘密中的 X25519 那一半，借助 CRQC 获得	任何能够使用密码学相关量子计算机的被动观察者都能恢复 ECDH 临时标量，该标量直接由 RNG 输出推导而来（rfc10024:299）。	s0561
`session_state`	该会话的握手流量秘密和应用流量秘密，在会话期间从其中一方的内存中读取	我考虑这一点，是因为必须问及一方暴露的后果：会话状态被读取的一方持有握手流量秘密和应用流量秘密（s0489, s0519），剩下的就是另一方能得到的保证。	s0489, s0519

ID	`network`	`comp_traditional`	`comp_mldsa`	`ltk_after_session`	`suf_cma`	`cross_protocol`	`crqc_x25519`	`session_s`
A02	成立	成立	成立		成立		成立	成立
A03	成立							
A04	成立				成立			
A08	成立					成立		
K01	成立						成立	
C01				成立				
C08		成立	成立					成立
R05	成立							
C08.9		成立	成立					
A08.10	成立					成立		
A08.11	成立							
X01		不成立						

原文和目标

来源：564 个原文单元，149815 字节，冻结在 source-delivery-01.json 中。 目标：已批准 8 个（A02, A03, A04, A08, K01, C01, C08, R05）；提出的问题：3 个。 目标列表事先按草案批准，使用 --approve-targets。

宿主按其文档规定的默认值（任务 `decisions: auto`）做出的决定，供你核对：

- approve_package：批准源文包 protocol-b387105272c6f7bd98b9d23cc2dc

本步骤的自我报告：1 次请求（1 次失败），0.000 USD，输入 69677 tokens，输出 4040 tokens，耗时 27 秒。

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
52	A02, A03, A04, A08, A08.10, A08.11, K01, C01, C08, C08.9, R05, X01	plain_primitives, many_sessions	模型只使用签名（两个组件算法，组合而成）、一个 KEM、作为普通密钥协商且只有单一规律 g^xy = g^yx 的 Diffie-Hellman、哈希、一个 AEAD 和一个带密钥的派生；没有异或，也没有任何性质所依赖的计数器或替换密钥，因此没有编写 bounded.pv。答案依赖两个理想化：密钥调度和 Finished MAC 是私有（单向）函数，因此即使攻击者知道共享秘密也无法计算出流量密钥；握手飞行消息在发送时没有记录层加密。所有目标都在任意数量的客户端和服务器会话上提出。

分析者的结语

全部十二个目标都在一个带对照查询的 ProVerif 模型（model.pv）上得到回答：A02、A03、A04、A08、A08.10、A08.11、K01、C01、C08、C08.9 和 R05 已证明；X01（草案自身警告的仅传统组件的回退，s0048）是唯一的攻击，标记为记录于 main:558，并绘制为图。值得引用的结果：在恰好一个组件私钥被攻破、X25519 那一半被 CRQC 恢复、会话密钥被读取、并且存在来自另一协议上下文的同密钥签名的情况下，客户端只有在真实服务器使用协商方案和固定的复合密钥类型对本次连接的转录签名时，才会接受服务器的 CertificateVerify；不接受任何 RSA-PKCS1-v1_5 复合方案、独立的 ML-DSA 或 TLS 1.2 连接。关于 X01 的建议已在模型的副本中检查：让第二个客户端要求复合证书和两个组件的签名后，冒充消失了，正常流程仍然成立，所有其他答案保持不变。限制在于：握手飞行消息在建模时没有记录层加密，密钥调度和 Finished MAC 是单向的，因此证明针对的是签名和转录，而不是记录层或共享秘密泄露；被攻破组件的攻击者得到的是对一条消息的一个组件签名，而不是签名预言机。没有任何目标依赖计数器或替换密钥，因此不需要 bounded.pv，也不需要状态审查。下一步：使用带记录层和公开 KDF 的 Tamarin+ Agent 模型，看看当攻击者能够端到端扮演一方时是否仍然成立；在这里把这些函数设为公开的尝试使 ProVerif 超时。