PQC EAP-TLS 完整认证的安全性分析

本报告的其他语言版本：English。

draft-ietf-emu-pqc-eap-tls-04 · ProVerif+ Agent 报告

文稿	Post-Quantum Enhancements to TLS-Based EAP Methods（协议定义）; EAP-TLS 1.3: Using the Extensible Authentication Protocol with TLS 1.3（辅助）; 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	18 条：3 条成立，2 条不成立（3 个攻击），13 条没有答案
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

该协议是面向 EAP 部署的完整 EAP-TLS 1.3 认证，这些部署必须能够抵御未来的量子攻击者。参与方有两方：EAP peer 持有自己的证书和 ML-DSA-65 签名密钥；EAP server 持有自己的证书和签名密钥。两者都信任同一个根 CA，peer 配置了它所接受的 server 名称。中间的认证器只转发 EAP 报文；它从 EAP server 收到的密钥不在分析范围内。交换过程为：EAP identity 请求与响应、TLS start，随后是在 X25519MLKEM768 混合组上进行的 TLS 1.3 完整握手，采用双向 ML-DSA-65 证书认证。之后 server 发送加密的 TLS 应用数据 0x00 作为受保护的成指示，peer 以空 EAP 响应作答，server 再发送 EAP-Success。双方都从 TLS exporter 导出 Key_Material，取其前 64 个八位组作为 MSK，接下来的 64 个八位组作为 EMSK。交换结束时，peer 确信对端已用它所信任的证书的密钥对握手转录签名，server 同样确信 peer 证书的对应事实。最后的 EAP-Success 本身不携带任何保护，因此它不能使任何一方确信任何事情。

参与方：

- **EAP peer**：持有自己的证书、ML-DSA-65 密钥、根 CA 密钥以及它期望的 server 名称。持有：certP, skP, skCA, nameS。
- **EAP server**：持有自己的证书、ML-DSA-65 密钥和根 CA 密钥。持有：certS, skS, skCA。

图 1 展示了文稿所规定的交换过程。

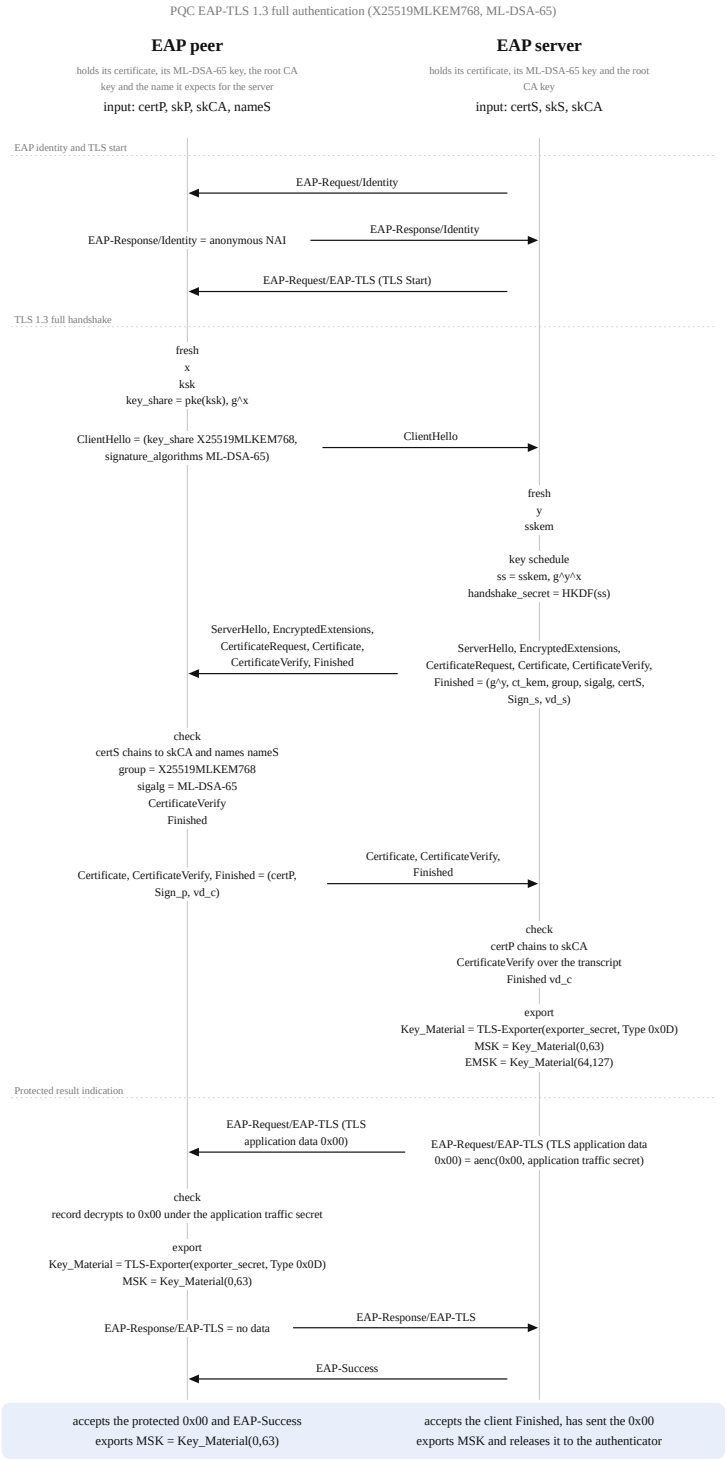


图 1. 文稿所规定的协议。

分析范围见附录 D：分析了什么、遗漏了什么，以及它依据的文稿章节。

2 发现

没有发现。 分析没有发现文稿尚未说明或已排除的内容。第 3 节的表格给出了每条性质的结果。

凡证明器能够判定的地方，文稿规定的握手行为都符合预期，也没有发现新的弱点。文稿已经说明的两处弱点得到了复现：攻击者若恢复了 server 的 ML-DSA-65 签名密钥，就可以向 peer 冒充 EAP server；EAP-Success 不携带任何保护，因此攻击者可以告诉 peer 本次运行成功了，而 peer 真正可以依赖的可能是受保护

的 0x00 记录。成功指示相关的顺序规则对任意数量的会话都成立，证书签名密钥无法从网络中获取，客户端证书中的身份对被动观察者保持隐藏。单次运行上的一致、单射性、新鲜性、可协商参数、nonce 规则以及混合组件相关的主张都没有答案：共享秘密仅按 g^xy = g^yx 这一规律建模，ProVerif 无法识别它的两种写法，因此有六对 target 与攻击者对保持开放，若干已记录的答案被标记为模型的产物。在 18 个 target 中，5 个基于已证明的查询或已验证的保密性结果，2 个是文稿已说明的攻击，5 个未定，6 对开放；混合组的算术问题需要由 Tamarin+ Agent 来判定。

3 检查过的性质

每一行都是一条提交给 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说明该结果，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	**真实对端正常流程上的双向一致。** EAP peer 接受 EAP-Success 时，存在一个以该 peer 为对端的 EAP server 真实运行；EAP server 接受 client Finished 时，存在一个以该 server 为对端的 EAP peer 真实运行。	无	攻击成立（文稿已写明）	中	是
A03	**导出密钥材料的一致性。** peer 与 server 各自接受时，双方对身份、协商组、签名算法、transcript 与 exporter_secret 导出的 Key_Material 一致。	无	无答案	none	不确定
A04	**密钥建立上的单射一致。** peer 的每次 EAP-Success 接受与 server 的每次 client Finished 接受，都对应互不相同的对端运行（单射匹配）。	无	无答案	none	不确定
A06	**客户端 Finished 后的成功指示。** peer 接受受保护的 0x00 成功指示时，server 已导出同一会话密钥并已处理 client Finished。	无	无答案	中	是
A08	**转录与协商绑定。** peer 与 server 接受的协商组、签名算法、transcript 与 exporter 上下文均来自本次握手的真实提议与选择。	无	只确认可达	none	不确定
K01	**MSK 与长期密钥的保密性。** 攻击者不能导出 MSK/Key_Material、exporter_secret，也不能获得 peer 或 server 的长期签名私钥。	无	成立	中	否
C01	**MSK 的前向保密。** 会话完成后泄露双方长期签名私钥，攻击者仍不能导出该会话的 Key_Material/MSK 或解密其载荷。	peer 与 server 的长期签名私钥 (会话完成后泄露)	无答案	none	不确定
C08	**一部分被攻破下的混合安全。** 在 X25519 或 ML-KEM-768 之一被攻破的假设下，会话密钥与载荷仍保密。	X25519 临时私钥或 ML-KEM-768 私钥之一 (握手期间或之后)	无答案	none	不确定
R01	**成功指示的新鲜性。** peer 接受的 0x00 成功指示与 server 接受的 client Finished 均来自本次握手的随机数与 transcript，而非旧会话重放。	无	无答案	none	不确定
R02	**每条消息仅接受一次。** 同一会话内，同一 0x00 成功指示或 client Finished 至多被接受一次。	无	无答案	none	不确定

ID	性质	假定的泄露	结果	价值	文稿是否已写
R05	**仅接受 PQ 参数**。 peer 与 server 接受的版本、密钥交换组与签名算法满足 TLS 1.3 与 PQ/混合的最低策略。	无	只确认可达	none	不确定
S01	**每个流量密钥唯一 nonce**。 同一会话密钥下，每次 AEAD 加密调用的 nonce 唯一。	无	无答案	none	不确定
S04	**MSK 绑定到 TLS 身份**。 导出的 MSK 与本次 TLS 会话的认证身份、transcript 正确关联；EAP 外层标识不被用于授权。	无	无答案	中	是
A03.14	**对端与服务器就密钥材料达成一致**。 peer 与 server 各自接受时，双方对身份、transcript 与 exporter_secret 导出的 Key_Material 一致。	无	无答案	none	不确定
U01	**MSK 保密性与有序 EAP-Success**。 MSK 仅对合法双方可知；EAP-Success 仅在处理合法 client Finished 并发送受保护 0x00 成功指示之后发送。	无	成立	中	是
U02	**外部身份不能剥离 MSK**。 外层 EAP 标识的替换或重放不改变 MSK 与本次 TLS 认证身份及 transcript 的绑定，且攻击者不能导出 MSK。	无	无答案	中	是
X01	**签名密钥恢复后的服务器冒充**。 持有服务器证书签名密钥的攻击者，可以在没有任何 EAP server 运行的情况下，使 EAP peer 完成握手并导出 MSK；没有该密钥，就无法构造出在没有 server 运行的情况下被 peer 接受的运行	peer 证书与 server 证书的 ML-DSA-65 私钥，在会话之前或期间被恢复	攻击成立（文稿已写明）	中	是
X02	**握手过程中对端身份保密性**。 记录握手的被动攻击者不会得知 peer 证书中携带的身份，除非它攻破了混合密钥协商中的某个组件	无	成立	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）。

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。

所提出的问题，以及分析者的回答：

问题	回答	依据
Q1: EAP peer/server 认证与 TLS 会话绑定	server 侧成立而 peer 侧不成立：持有 server 的签名密钥，攻击者可以让 peer 在没有 server 的情况下完成运行；而仅凭网络，peer 会接受 server 从未发送过的 EAP-Success。双方就导出密钥材料达成一致这一性质无法判定。	A02, A03, X01
Q2: MSK 保密及 EAP Success 不能早于合法 TLS 成功	网络上的攻击者无法获得 MSK，且顺序已得到证明：EAP-Success 在 client Finished 与受保护的 0x00 之后发送，对任意数量的会话都成立。MSK 保密性结果在模型中成立，仅仅是因为要攻破它需要两个组件同时泄露。	K01, U01
Q3: 外层标识混淆/重放对 exporter 会话的影响	替换或重放外层身份既不会移动 MSK 也不会泄露 MSK，因为 MSK 跟随的是 TLS 内部认证的证书身份，而该身份对被动观察者保持隐藏，除非混合组中的某个组件被攻破。	U02, S04, X02

4 本次分析的局限

这里的结果是关于协议模型的结果：它说明所规定的消息与检查能给出或不能给出什么，其中密码学被视为理想原语，对实现不作任何断言。

正如分析者所述，本次分析不涵盖以下内容：

- 混合组被建模为一个 KEM 秘密与一个 Diffie-Hellman 项的配对，只带一条定律 g^xy = g^yx；两方以不同顺序写出该项，ProVerif 不认为它们相同，这正是无结论结果以及所有标记为 model artifact 的答案的原因。
- 两个组件泄露被建模为任何攻击者随时可以发出的请求，因此对于不获取这两者的攻击者，任何目标都无法作答；这就是前向保密与混合组目标没有可用答案的原因。
- 证书依据一个根 CA 密钥和一个预期服务器名进行检查；吊销、OCSP stapling、中间链和证书压缩均未纳入。
- 没有会变化的值：没有 NewSessionTicket，没有恢复，没有 KeyUpdate，也没有记录序列号，因此计数器、重密钥和记录重复都无法考察，这些需要界模型。
- EAP server 到 authenticator 的密钥传输、AAA 协议、下层四次握手和接入网信道性质是任务设定的边界，因此对它们不作任何说明。
- 符号模型不证明 ML-KEM 或 ML-DSA 的计算强度，也不证明侧信道、随机数生成或实现故障方面的任何东西。
- 时序、可用性和拒绝服务不是证明器会作答的性质，因此伪造 Success 包用作拒绝服务这一点未被考察。
- ProVerif 在单射查询和记录匹配查询上达到时间上限，因此单射性和新鲜性问题在这里没有答案。
- 六个目标与攻击者组合仍无结论：对端侧关于运行确实发生的检查、MSK 前向保密与混合组的两个组件情形，以及 MSK 与证书身份的绑定。它们的迹需要假定的泄露，因此记录为无答案，而不是模型自行产生的攻击。

宿主记录的内容：

- 无答案：A03、A04、A06、A08、C01、C08、R01、R02、R05、S01、S04、A03.14、U02。
- 在本解读写成之后答案发生变化的：A02、A03、K01。
- A02、A03、A08、K01、C01、C08、R05、S01、S04、U01、U02、X01、X02 的答案依赖于模型简化的某个操作（附录 D）。
- 证明器没有给出结论的查询（附录 C）：A03: a03_km, A04: a04_inj_peer, A06: a06_zero, R01: r01_zero, R02: r02_once, S01: s01_once, A03.14: a03_km。这条性质问过了，证明器既没有证明它也没有找到攻击，所以结果写“无答案”。
- 分析者为性质设定了攻击者条件，但既没有证明也没有攻击的（附录 D）：A08 在 `network` 下; C01 在 `ltk_leak` 下; C08 在 `x25519_broken` 下; C08 在 `mlkem_broken` 下; R05 在 `network` 下; S04 在 `network` 下; U02 在 `network` 下。
- 没带对照查询的证明（附录 C）：K01: k01_ltk_p, K01: k01_ltk_s, U01: u01_order, U01: u01_zero, X02: id_ctl。对照查询是这条性质的一个必然不成立的变体；没有它，就没有东西表明这条性质照现在的写法在模型里有可能不成立。

每条性质还留下什么未决，以及什么能解决它，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答案的解读，不是证明器的答案：这些答案对协议意味着什么，解读依据什么，能否按原样使用，以及它值多少。

A02: 真实对端正常流程上的双向一致

EAP peer 接受 EAP-Success 时，存在一个以该 peer 为对端的 EAP server 真实运行；EAP server 接受 client Finished 时，存在一个以该 server 为对端的 EAP peer 真实运行。

本目标在本解读写成之后答案发生了变化。

server 侧成立：没有任何迹能让 server 接受一个没有 peer 产生的 client Finished。在 peer 侧，两条主张分道扬镳。在拥有证书密钥的情况下，攻击者能让 peer 接受并导出一个 MSK，而没有任何 server 运行；甚至仅靠网络，peer 也会接受 server 未发送的 EAP-Success，因为 EAP-Success 不带保护；peer 可以依赖的是受保护的 0x00 记录。

- 结果：攻击成立（文稿中已说明）；回答所问：部分；带保留可用
- 价值中：peer 侧的失败正是 RFC 9190 第 5.5 节针对 EAP-Success 陈述的弱点，冒充则是正文第 4 节针对已恢复证书密钥陈述的那一个；分析补充的是确切的消息和确切的迹。
- 文稿中已说明：是。文稿在哪里说的：rfc9190:1095, main:306, s0155, s0018
- 依据：X25519 共享秘密用单一律 g^xy = g^yx 建模，ML-KEM 秘密建模为单向函数，ProVerif 不认为该秘密的两种写法相同；证书依据一个根 CA 密钥和一个预期服务器名进行检查；吊销和 OCSP 未纳入；authenticator 是透传的，其密钥传输属于排除在外的边界
- 未决：server 侧方向作为已证明的对应关系；就一次运行而非 transcript 值达成一致

A03: 导出密钥材料的一致性

peer 与 server 各自接受时，双方对身份、协商组、签名算法、transcript 与 exporter_secret 导出的 Key_Material 一致。

本目标在本解读写成之后答案发生了变化。

双方在同一 transcript 上运行相同的密钥调度，而证明器既不能证明也不能反驳它们导出的 Key_Material 一致。其推导把两个不同的会话配对，并对仅在 Diffie-Hellman 指数顺序上不同的项做哈希，因此该答案对协议无论哪边都说明不了什么。

- 结果：无答案；回答所问：部分；不可用
- 价值无：没有返回答案。
- 文稿中已说明：不明确
- 依据：单一律 Diffie-Hellman 等式，在推导中应用，但在匹配两个事件时未应用；exporter 作为未解释的单向函数
- 未决：双方导出的 Key_Material 是否一致

A04: 密钥建立上的单射一致

peer 的每次 EAP-Success 接受与 server 的每次 client Finished 接受，都对应互不相同的对端运行（单射匹配）。

无答案：单射查询没有跑完，因此一次接受能否代表两次运行尚未确立。

- 结果：无答案；回答所问：不明确；不可用
- 价值无：证明器没有给出任何可依据的东西。
- 文稿中已说明：不明确
- 依据：完全没有结果；每次运行的标识符位于被哈希的 transcript 之内，这正是单射查询代价高昂的原因
- 未决：任一方向的单射性

A06: 客户端 Finished 后的成功指示

peer 接受受保护的 0x00 成功指示时，server 已导出同一会话密钥并已处理 client Finished。

目标真正要求的顺序已经确立：server 只有在接受 client Finished 之后才发送受保护的 0x00，而 EAP-Success 只在 0x00 之后。peer 对该记录的接受是否对应一个产生了它的 server 运行，无法判定；该记录用只出自同一会话的密钥封装。

- 结果：无答案；回答所问：部分；带保留可用
- 价值中：RFC 9190 第 2.5 节陈述了顺序规则，分析确认模型保持了这些规则；对应关系那一半没有答案。
- 文稿中已说明：是。文稿在哪里说的：rfc9190:1094, s0138
- 依据：记录匹配查询触及时间上限；0x00 用一个 AEAD 记录，没有记录编号或填充
- 未决：peer 对 0x00 的接受与某个 server 运行相匹配

A08: 转录与协商绑定

peer 与 server 接受的协商组、签名算法、transcript 与 exporter 上下文均来自本次握手的真实提议与选择。

模型只能表明某次运行在后量子组和签名方案下完成，因为它没有第二个组或方案可回退。因此无法展示策略失败，答案对同时配置了传统参数的 peer 会怎么做也说明不了什么。

- 结果：仅可达；回答所问：否；不可用
- 价值无：唯一记录的答案是情形可达，这不是所问的性质。
- 文稿中已说明：不明确
- 依据：模型中只有一个组和一个签名方案；策略检查是 peer 中与常量的比较
- 未决：peer 能否被诱导接受传统组或传统签名方案

K01: MSK 与长期密钥的保密性

攻击者不能导出 MSK/Key_Material、exporter_secret，也不能获得 peer 或 server 的长期签名私钥。

本目标在本解读写成之后答案发生了变化。

两个证书签名密钥都无法被网络上的攻击者获得。MSK 本身在模型中可获得，但只有在攻击者获取了两个建模的组件泄露之后，而这正是其他目标所谈论的失陷，不是网络攻击。

- 结果：成立；回答所问：部分；带保留可用
- 价值中：结果源自模型的泄露机制；文稿说明 MSK 从 TLS 会话导出，但未讨论这一点。
- 文稿中已说明：否
- 依据：组件泄露被建模为任何攻击者都可以发出的请求，因此模型无法禁止它们；MSK 是 exporter 输出的未解释函数

- 未决：受限在网络的攻击者下的 MSK 保密性

C01: MSK 的前向保密

会话完成后泄露双方长期签名私钥，攻击者仍不能导出该会话的 Key_Material/MSK 或解密其载荷。

前向保密没有任何展示：用于证伪的迹除了证书密钥之外还需要两个组件披露，因此它并不能说明目标所要求的内容。本可让该答案真正意味着前向保密的排他性同样无法被证明。

- 结果：无答案；回答了所问：否；不可用
- 值为 none：该答案是披露机制的产物。
- 文稿中已说明：不明确
- 依赖于：组件披露始终可用，因此无法把假设的失陷隔离出来；单律 Diffie-Hellman 方程与共享密钥的两种写法
- 未决：MSK 能否在同时丢失两个证书密钥后继续存在

C08: 一部分被攻破下的混合安全

在 X25519 或 ML-KEM-768 之一被攻破的假设下，会话密钥与载荷仍保密。

两个组件主张都没有得到回答：两条证伪迹都同时使用了另一个披露，因此它们无法说明混合组在某一组件被攻破时仍能保密。因此，“混合构造能应对先存储后解密威胁”这一结论依赖的是文稿，而不是本分析。

- 结果：无答案；回答了所问：否；不可用
- 值为 none：两个答案都是披露机制的产物。
- 文稿中已说明：不明确
- 依赖于：组件披露始终可用；证明器未识别出 X25519 共享密钥的两种写法
- 未决：对任一组件按组件分别进行的混合论证

R01: 成功指示的新鲜性

peer 接受的 0x00 成功指示与 server 接受的 client Finished 均来自本次握手的随机数与 transcript，而非旧会话重放。

无答案：查询未完成。transcript 哈希与每次运行的流量密钥确实把 peer 的接受绑定到它自己的 key share，旧记录也无法用新会话的密钥解密，但模型未能确立这一点。

- 结果：无答案；回答了所问：不明确；不可用
- 值为 none：证明器没有给出任何可依据的结论。
- 文稿中已说明：不明确
- 依赖于：无结果；没有记录号或序列号，因此唯一的新鲜性来源是 key share 与流量密钥
- 未决：对 0x00 接受的 freshness

R02: 每条消息仅接受一次

同一会话内，同一 0x00 成功指示或 client Finished 至多被接受一次。

无答案：单射查询触发了时间上限。Finished 检查与 0x00 检查都按会话进行，流量密钥按每次运行生成，但一条记录能否被接受两次并未判定。

- 结果：无答案；回答了所问：不明确；不可用

- 值为 none：证明器没有给出任何可依据的结论。
- 文稿中已说明：不明确
- 依赖于：无结果；没有序列号，因此重复的记录就是重复的项
- 未决：每条消息至多被接受一次

R05: 仅接受 PQ 参数

peer 与 server 接受的版本、密钥交换组与签名算法满足 TLS 1.3 与 PQ/混合的最低策略。

模型只有一个组和一种签名方案，因此它只能展示在这两者之下的一次运行；它无法展示当攻击者提供传统组时 peer 会如何反应。因此，“部署必须协商后量子组与签名”这一要求被当作已满足来建模，而不是被检验。

- 结果：仅可达；回答了所问：否；不可用
- 值为 none：记录的答案是可达性，并非所问的性质。
- 文稿中已说明：不明确
- 依赖于：一个组和一种签名方案；策略检查是与常量比较
- 未决：针对同时支持传统组的 peer 的降级尝试

S01: 每个流量密钥唯一 nonce

同一会话密钥下，每次 AEAD 加密调用的 nonce 唯一。

证明器既不能证明也不能反驳“同一流量密钥下不会用相同 nonce 封装两条记录”；其推导把两次运行配对，而这两次运行仅在过度近似中共享密钥。在普通运行中每条记录都有自己的 nonce，但这不是证明。

- 结果：无答案；回答了所问：不明确；不可用
- 值为 none：没有返回答案。
- 文稿中已说明：不明确
- 依赖于：每个 flight 一次 AEAD 调用且没有序列号；推导是一种过度近似，ProVerif 无法将其转化为迹
- 未决：同一流量密钥能否用相同 nonce 封装两条记录

S04: MSK 绑定到 TLS 身份

导出的 MSK 与本次 TLS 会话的认证身份、transcript 正确关联；EAP 外层标识不被用于授权。

server 认证的是证书身份，而不是外层 NAI，并从同一个 TLS 会话导出 MSK；迹显示了这一点，因为 client flight 以及导出 MSK 所依据的 transcript 都携带证书身份。对应性查询之所以被报告为假，只是因为 peer 自身的密钥事件在运行中出现得更晚。

- 结果：无答案；回答了所问：部分；可用但有保留
- 值为中：RFC 9190 第 2.2 节与第 5.6 节指出外层标识未经过认证，不得用于授权；分析表明 MSK 跟随证书身份。
- 文稿中已说明：是。文稿中的位置：rfc9190:1136, s0115
- 依赖于：外层标识是攻击者可自由改写的值，这是文档记载的行为；MSK 被交给 authenticator，其传输属于被排除的边界
- 未决：作为已证明对应性关系的绑定

A03.14: 对端与服务器就密钥材料达成一致

peer 与 server 各自接受时，双方对身份、transcript 与 exporter_secret 导出的 Key_Material 一致。

与 A03 答案相同：同一 transcript 上采用同一密钥调度，证明器未给出判定，其推导配对了不同的会话。

- 结果：无答案；回答了所问：部分；不可用
- 值为 none：没有返回答案。
- 文稿中已说明：不明确
- 依赖于：同前的单律 Diffie-Hellman 方程；exporter 作为单向函数
- 未决：对导出密钥材料的一致性

U01: MSK 保密性与有序 EAP-Success

MSK 仅对合法双方可知；EAP-Success 仅在处理合法 client Finished 并发送受保护 0x00 成功指示之后发送。

EAP-Success 只在 server 接受 client Finished 之后发送，并且只在它发送受保护的 0x00 记录之后发送。这两点在任意数量的会话中都成立，这正是用户就成功指示顺序所问的内容。

- 结果：成立；回答了所问：是；可用
- 值为中：RFC 9190 第 2.5 节将这两条规则规定为 MUST，分析确认模型保持了它们。
- 文稿中已说明：是。文稿中的位置：rfc9190:1094, s0138
- 依赖于：server 的程序顺序即规范的顺序；只有一个 server 角色且没有分支；没有握手后消息，也没有恢复
- 未决：目标中 MSK 保密性那一半，在 K01 下作答

U02: 外部身份不能剥离 MSK

外层 EAP 标识的替换或重放不改变 MSK 与本次 TLS 认证身份及 transcript 的绑定，且攻击者不能导出 MSK。

迹显示一个 peer 正常完成，而另一个 server 会话记录了不同的外层标识，这是预期之内的：外层标识由攻击者自由选择，不绑定任何东西。关键在于 MSK 始终与 TLS 会话及其证书身份绑定；迹显示了这一点，但没有任何对应性查询对此加以证明。

- 结果：无答案；回答了所问：部分；可用但有保留
- 值为中：RFC 9190 第 2.2 节指出外层标识未经认证，伪造、修改或重放都很容易；分析重现了这并不会改变 MSK。
- 文稿中已说明：是。文稿中的位置：rfc9190:1136, s0115
- 依赖于：结论要求在外层标识交换的两个不同会话之间进行精确事件匹配；外层标识按设计从不认证
- 未决：作为已证明对应性关系的绑定

X01: 签名密钥恢复后的服务器冒充

持有 server 证书签名密钥的攻击者可以在没有 EAP server 运行的情况下让 EAP peer 完成握手并导出 MSK；没有该密钥，就无法产生无 server 运行的 peer 接受

有了 server 的 ML-DSA-65 签名密钥，攻击者即使没有 EAP server 运行，也能让 peer 完成握手并导出 MSK：它为自己的 CertificateVerify 与 finished flight 签名，而 peer 只检查证书能链到受信任根并指明预期的 server。这就是威胁部分针对证书密钥恢复所描述的冒充。

- 结果：攻击成立（文稿中已说明）；回答了所问：是；可用
- 值为中：正文第 4 节说明了这种冒充；分析给出了逐消息的迹。
- 文稿中已说明：是。文稿中的位置：main:306, s0018
- 依赖于：peer 针对单个根密钥和一个预期名称进行检查；吊销与 OCSP 被略去；借用的密钥在运行前就可用
- 未决：吊销检查是否能缩小窗口，模型略去了这一点

X02: 握手过程中对端身份保密性

记录握手的被动攻击者不会得知 peer 证书中携带的身份，除非它攻破混合密钥协商中的某个组件

记录握手过程的攻击者无法获得客户端证书中的身份：证书封装在由握手流量密钥加密的记录里，而这些密钥源自临时共享秘密。尽管外层 NAI 是明文，这一点仍然成立，因为 NAI 是匿名形式。

- 结果：成立；是否回答了所问问题：是；有保留地可用
- 值为中：RFC 9190 第 2.1.8 节和第 5.8 节指出，在 TLS 1.3 中证书是加密的，用户名不会被泄露，正文第 3 节指出了暴露问题；分析确认它在模型中成立。
- 文稿中是否已说明：是。文稿中的位置：rfc9190:1259、s0105、s0108
- 依据：身份机密性是以证书身份上的保密性查询提出的，而不是对两个对等身份做检验；模型中只有一个对等身份，因此不询问跨运行不可关联性；如果两个组件都泄露，身份就会被泄露，查询将此列为它的例外
- 未决：同一对等方两次运行之间的不可关联性

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，以及攻击及分析者对每个攻击的判定。

目标	级别	界	查询	对照	攻击	主张
A02	none				a02_p2s (holds_in_source)；a02_s2p (model_artifact)；a02_p2s_ltk (holds_in_source)；a02_s2p_ltk (model_artifact)	4 attack
A03	none				a03_km_rev (model_artifact)	1 attack，1 undecided
A04	none					1 undecided
A06	none					1 undecided
A08	reachable		a08_policy			1 reachable
K01	proved		k01_ltk_p		k01_msk (model_artifact)；k01_km (model_artifact)	2 proved，2 attack
C01	none				c01_msk (model_artifact)	1 attack
C08	none				c08_dh (model_artifact)；c08_kem (model_artifact)	2 attack
R01	none					1 undecided
R02	none					1 undecided
R05	reachable		a08_policy			1 reachable
S01	reachable		s01_aead			1 reachable，1 undecided
S04	none				s04_id (model_artifact)	1 attack
A03.14	none					1 undecided
U01	proved		u01_order			2 proved
U02	none				u02_bind (model_artifact)	1 attack
X01	none				a02_p2s_ltk (holds_in_source)	1 attack
X02	proved		id_ctl			1 proved

逐条主张，按分析者对每条主张的措辞；每条的回答来自证明器：

目标	主张	攻击者	回答	查询	对照
A02	对等方接受 EAP-Success 意味着存在一个发送了它的服务器运行，针对仅在网络上的攻击者	`network`	attack (holds_in_source)	a02_p2s	
A02	服务器接受客户端 Finished 意味着存在一个发送了它的对等方运行，针对仅在网络上的攻击者	`network`	attack (model_artifact)	a02_s2p	
A02	对等方接受 EAP-Success 意味着存在一个发送了它的服务器运行，针对持有证书签名密钥的攻击者	`ltk_leak`	attack (holds_in_source)	a02_p2s_ltk	
A02	服务器接受客户端 Finished 意味着存在一个发送了它的对等方运行，针对持有证书签名密钥的攻击者	`ltk_leak`	attack (model_artifact)	a02_s2p_ltk	
A03	对等方导出的 Key_Material 与服务器在同一次运行中导出的一致，针对仅在网络上的攻击者	`network`	undecided	a03_km	

目标	主张	攻击者	回答	查询	对照
A03	服务器导出的 Key_Material 与对等方在同一次运行中导出的一致，从服务器一侧读取	`network`	attack (model_artifact)	a03_km_rev	
A04	每次对等方接受都对应一个不同的服务器运行（单射一致），针对网络攻击者	`network`	undecided	a04_inj_peer	
A06	只有当服务器运行处理了其客户端 Finished 并发送了该记录时，对等方才接受受保护的 0x00	`network`	undecided	a06_zero	
A08	对等方只接受其部署所需后量子集合中的组和签名算法，针对网络攻击者	`network`	reachable	a08_policy	
K01	对等方证书签名密钥无法被仅在网络上的攻击者获得	`network`	proved	k01_ltk_p	none
K01	服务器证书签名密钥无法被仅在网络上的攻击者获得	`network`	proved	k01_ltk_s	none
K01	会话的 MSK 和 Key_Material 无法被仅在网络上的攻击者获得	`network`	attack (model_artifact)	k01_msk	
K01	Key_Material 本身无法被仅在网络上的攻击者获得	`network`	attack (model_artifact)	k01_km	
C01	已完成会话的 Key_Material/MSK 对持有两个证书签名密钥的攻击者保持机密	`ltk_leak`	attack (model_artifact)	c01_msk	
C08	当混合组中的 X25519 组件被攻破时，会话密钥保持机密	`x25519_broken`	attack (model_artifact)	c08_dh	
C08	当混合组中的 ML-KEM-768 组件被攻破时，会话密钥保持机密	`mlkem_broken`	attack (model_artifact)	c08_kem	
R01	对等方接受的 0x00 指示是在本次握手中基于对等方自己的密钥份额产生的，针对网络攻击者	`network`	undecided	r01_zero	
R02	客户端 Finished 和 0x00 指示在每个会话中至多被接受一次，针对网络攻击者	`network`	undecided	r02_once	
R05	只协商后量子混合组和 ML-DSA，针对网络攻击者	`network`	reachable	a08_policy	
S01	每条记录都用各自的 nonce 在流量密钥下封装，针对网络攻击者	`network`	reachable	s01_aead	
S01	没有两条记录会在同一流量密钥下使用相同 nonce 封装，针对网络攻击者	`network`	undecided	s01_once	
S04	服务器认证的身份就是 TLS 内部认证的客户端证书中的身份，且 MSK 来自同一个 TLS 会话，针对网络攻击者	`network`	attack (model_artifact)	s04_id	
A03.14	对等方和服务器为一次运行导出相同的 Key_Material（用户问题 Q1）	`network`	undecided	a03_km	
U01	只有在客户端 Finished 被接受后才发送 EAP-Success，针对网络攻击者	`network`	proved	u01_order	none
U01	只有在受保护的 0x00 指示之后才发送 EAP-Success，针对网络攻击者	`network`	proved	u01_zero	none
U02	替换或重放外层 EAP 身份不会让攻击者获得 MSK 或将其与另一身份绑定，针对网络攻击者	`network`	attack (model_artifact)	u02_bind	
X01	持有服务器证书签名密钥的攻击者能让 EAP 对等方在没有 EAP 服务器运行的情况下完成认证	`ltk_leak`	attack (holds_in_source)	a02_p2s_ltk	
X02	记录握手的攻击者无法获得客户端证书中的身份，除非混合密钥协商的某个组件被攻破	`network`	proved	id_ctl	none

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 节中的每个结果都是证明器的答案。

ProVerif+ Agent 会话 session-20261114T224408，2026-11-14。状态：已完成。模型：deepseek-v4-1-flash-261017。步数：39，证明器 176 秒，实际耗时 16 分钟，费用 0.204 USD。

上下文：宿主从对话中移除过时内容 1 次，共 272586 个字符（先前的推理，以及显示模型在最近一次更改之前状态的内容）。轨迹文件保留了全部内容。

ProVerif：运行了 62 个证明，49 个已判定，13 个无答案，163 秒，记录了 21 个答案。

分析了什么

协议	PQC EAP-TLS full authentication
版本	draft-ietf-emu-pqc-eap-tls-04
分析范围	一次完整 EAP-TLS 1.3 双向认证；固定 X25519MLKEM768 与 ML-DSA-65，建模 EAP 标识/Success 条件、TLS exporter 导出 MSK、TLS Finished 和握手 transcript。EAP server 与 authenticator 间可信密钥传输为边界；排除 EST、TEAP、resumption。
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	655 numbered passages (182927 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	3	基于 TLS 的 EAP 方法中的数据机密性
main	4	基于 TLS 的 EAP 方法中的后量子认证
main	7	安全考量
rfc9190	2	协议概览
rfc9190	2.1	EAP-TLS 会话概览
rfc9190	2.1.1	认证
rfc9190	2.1.2	票据建立
rfc9190	2.1.3	恢复
rfc9190	2.1.4	终止
rfc9190	2.1.5	无对端认证
rfc9190	2.1.6	Hello Retry Request
rfc9190	2.1.7	标识
rfc9190	2.1.8	隐私
rfc9190	2.1.9	分片
rfc9190	2.2	标识验证

文稿	章节	标题
rfc9190	2.3	密钥层次
rfc9190	2.4	参数协商与合规要求
rfc9190	2.5	EAP 状态机
rfc9190	3	EAP-TLS 协议的详细描述
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	Client Hello
rfc9846	4.2.3	Server Hello
rfc9846	4.2.4	Hello Retry Request
rfc9846	4.3	扩展
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 Binder
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	Certificate Request
rfc9846	4.5	认证消息
rfc9846	4.5.1	Certificate
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 Certificate 消息

文稿	章节	标题
rfc9846	4.5.2	Certificate Verify
rfc9846	4.5.3	Finished
rfc9846	4.6	早期数据结束
rfc9846	4.7	握手后消息
rfc9846	4.7.1	New Session Ticket 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
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
rfc10024	4	协商群组
rfc10024	4.1	客户端共享
rfc10024	4.2	服务器共享
rfc10024	4.3	共享密钥

源文步在文稿中定位的协议目的：一次完整 EAP-TLS 1.3 双向认证；固定 X25519MLKEM768 与 ML-DSA-65，建模 EAP 标识/Success 条件、TLS exporter 导出 MSK、TLS Finished 和握手 transcript。EAP server 与 authenticator 间可信密钥传输为边界；排除 EST、TEAP、resumption。

对照文稿定性质

在写出查询之前，分析者将草拟的性质清单与文稿自身对协议目的的表述进行了对照。名称、性质所涉及的对象以及文稿的表述都是分析者的解读；所引位置在文稿中确实存在。

ID	简称	对象	文稿怎么说	出处
A02	真实对端正常流程上的双向一致	EAP 对端在收到 EAP-Success 时确信有真实的 EAP 服务器与它完成了握手；EAP 服务器在接受客户端 Finished 时确信有真实的 EAP 对端完成了握手	RFC 9190 s0046/s0049：服务器只在对端的 Finished 完成握手之后才发送 EAP-Success；双方的证书和 CertificateVerify 认证了各方。	s0046, s0049, s0497, s0523
A03	导出密钥材料的一致性	EAP 对端与 EAP 服务器在一次运行中导出相同的 Key_Material/MSK	RFC 9190 s0121-s0124：双方从同一个 TLS 会话导出 Key_Material；RFC 9846 s0523/s0524 将其绑定到转录。	s0121, s0122, s0124, s0523
A04	密钥建立上的单射一致	一方每次接受 MSK 都对应另一方一次不同的运行	由 RFC 9846 s0524 的 Finished 语义与新鲜临时共享（RFC 9190 s0046）蕴含；文稿未明确陈述单射性。	s0046, s0524

ID	简称	对象	文稿怎么说	出处
A06	客户端 Finished 后的成功指示	EAP 对端在接受受保护的 0x00 时确信服务器已处理其 Finished 并持有会话密钥	RFC 9190 s0138：服务器在处理完客户端 Finished 之前不得发送受保护的成功指示。	s0138, s0046
A08	转录与协商绑定	对端所接受的内容绑定到本次握手的密钥共享以及协商的群组和签名算法	RFC 9846 s0533 Finished 覆盖转录；RFC 10024 s0652 混合共享密钥；main s0011/s0019 后量子群组与签名要求。	s0011, s0019, s0533, s0652
K01	MSK 与长期密钥的保密性	会话的 Key_Material/MSK 对攻击者保密；证书签名密钥无法从网络获得	RFC 9190 s0121-s0124 MSK 从 TLS 会话导出；main s0009/s0010 要求长期保密性。	s0121, s0124, s0009
C01	MSK 的前向保密	已完成会话的 Key_Material/MSK 对后来获得证书签名密钥的攻击者保密	main s0002/s0003/s0009：HNDL 威胁由临时密钥交换应对；RFC 9190 s0046 前向保密来自临时 key_share。	s0002, s0003, s0009, s0046
C08	一部分被攻破下的混合安全	会话密钥对攻破 X25519MLKEM768 某一组件的攻击者保密	main s0012：只要任一组件算法未被攻破，后量子/传统混合共享密钥就保持安全。	s0012, s0652, s0010
R01	成功指示的新鲜性	对端所接受的 0x00 成功指示，是在本次握手中基于对端自身的密钥共享产生而非重放的	RFC 9846 s0527 警告 ClientHello 可能在握手完成前被重放；RFC 9190 s0138 使 0x00 成为已完成运行的证明。	s0138, s0527
R02	每条消息仅接受一次	客户端 Finished 与 0x00 指示，每个会话至多被接受一次	RFC 9846 s0524 接收方必须验证 Finished；RFC 9190 s0138 每次运行一个受保护的成功指示。	s0524, s0138
R05	仅接受 PQ 参数	协商的群组与签名算法，对端只能从其部署所要求的后量子集合中选择	main s0009/s0010/s0019：要求长期保密性的部署必须协商后量子/传统混合群组，并且必须支持后量子签名方案。	s0009, s0010, s0019
S01	每个流量密钥唯一 nonce	同一流量密钥下的每次 AEAD 调用，必须使用不同的 nonce	RFC 9846 s0618 由记录序列号导出每条记录的 nonce；文稿陈述的是要求，而非攻击。	s0138, s0618
S04	MSK 绑定到 TLS 身份	交付给 authenticator 的 MSK 属于 TLS 内部认证的身份，而非外层 EAP 标识	RFC 9190 s0115：EAP-Response/Identity 未经认证，不得用于授权；s0121/s0141 MSK 仅来自 TLS 会话。	s0115, s0121, s0141
A03.14	对端与服务器就密钥材料达成一致	一次运行中对端与服务端的 Key_Material（用户问题 Q1）	同 A03：RFC 9190 s0121-s0124 以及 RFC 9846 s0523 Finished 绑定。	s0121, s0124, s0523
U01	MSK 保密性与有序 EAP-Success	MSK 仅两个参与方知晓；EAP-Success 只在客户端 Finished 和受保护的 0x00 之后发送（用户问题 Q2）	RFC 9190 s0138/s0141 与 s0121/s0124。	s0138, s0141, s0124
U02	外部身份不能剥离 MSK	当外层 EAP 身份被替换或重放时，MSK 属于哪个会话（用户问题 Q3）	RFC 9190 s0115 外层身份未经认证且不得用于授权；s0121 MSK 仅从 TLS 会话导出。	s0115, s0121
X01	签名密钥恢复后的服务器冒充	EAP 对端对于自己与真实 EAP 服务器完成握手的确信	main s0018：能够在证书有效期内恢复服务器私钥的路径攻击者可实时冒充接入点。	s0014, s0018

ID	简称	对象	文稿怎么说	出处
X02	握手过程中对端身份保密性	客户端证书中的对端身份，对被动观察者保密	RFC 9190 s0105/s0108：使用 TLS 1.3 时证书消息被加密，EAP-TLS 始终与隐私一起使用；main s0002 警告一旦密钥交换被攻破，录制的握手会暴露客户端身份。	s0105, s0108, s0004

分析者依据文稿补充的性质：X01, X02.

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 191 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 70 段，没选 121 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
P001 至 P002	draft-ietf-tls-mldsa:181-192	证明器问不了	选择消息攻击下的签名不可伪造性以及侧信道失效是该原语及其实现的特性
P003	main:564-568	不在范围内	它涉及 EST 证书发放扩展，任务将其排除在外
P004 至 P006	main:570-582	没有陈述目标	它只是指向其他文稿的安全考虑；随后是版面内容
P007	rfc10024:268-271	A08, C08	
P008	rfc10024:273-276	证明器问不了	KEM 实现的侧信道抵抗能力不是符号模型的特性
P009	rfc10024:278-280	没有陈述目标	它说明这些组使用定长值，并未禁止证明器可以运行的任何攻击
P010 至 P011	rfc10024:282-297	证明器问不了	封装随机数的泄露以及不安全随机数生成器的状态是随机数生成器的特性
P012	rfc10024:299-306	C08, `x25519_broken`	
P013	rfc10024:308-310	证明器问不了	它涉及两种算法共用一个不安全的随机数生成器
P014 至 P015	rfc3748:2182-2190	没有陈述目标	它引入通用的 EAP 威胁模型，并说明为什么要有主张章节
P016 至 P017	rfc3748:2194-2203	`network`	
P018	rfc3748:2205-2206	X02, `network`	
P019	rfc3748:2208-2208	A02, A06, `network`	
P020	rfc3748:2210-2213	R02, `network`	
P021	rfc3748:2215-2216	没有陈述目标	离线字典攻击需要有人选定的口令；使用证书的 EAP-TLS 没有口令
P022	rfc3748:2218-2219	A02, `network`	
P023	rfc3748:2221-2222	R05, A08, `network`	
P024	rfc3748:2224-2225	K01, A03	
P025	rfc3748:2233-2234	不在范围内	它涉及认证之后在低层使用的密码套件
P026	rfc3748:2236-2238	不在范围内	低层密码套件的降级不在所建模的 EAP-TLS 交换之内
P027	rfc3748:2240-2244	不在范围内	通过 AAA 或低层带外给出的信息处于任务排除的边界
P028	rfc3748:2246-2252	`network`	

段落	出处	定为	不选的原因
P029	rfc3748:2256-2258	没有陈述目标	它要求方法规范中有主张章节
P030	rfc3748:2260-2261	没有陈述目标	它要求规范指明其认证技术
P031	rfc3748:2263-2271	A02, A04, R02, K01	
P032	rfc3748:2273-2276	证明器问不了	以比特计的有效密钥强度不是能向证明器提出的问题
P033	rfc3748:2286-2294	证明器问不了	以比特计的密钥强度定义是计算性的
P034	rfc3748:2296-2299	没有陈述目标	关于定义密钥强度之难处的说明
P035	rfc3748:2301-2308	证明器问不了	它涉及密钥强度如何取决于派生过程
P036	rfc3748:2310-2313	A03	
P037	rfc3748:2315-2317	没有陈述目标	它要求方法说明自己不做出哪些主张
P038	rfc3748:2321-2322	没有陈述目标	随后是术语列表
P039	rfc3748:2324-2328	A08, R05	
P040	rfc3748:2339-2344	A02	
P041	rfc3748:2346-2351	A02, R01	
P042	rfc3748:2353-2355	R02	
P043	rfc3748:2357-2361	K01, X02	
P044	rfc3748:2363-2368	A03, K01	
P045	rfc3748:2370-2374	证明器问不了	以比特计的密钥强度
P046	rfc3748:2376-2383	没有陈述目标	使用证书的 EAP-TLS 不使用口令
P047	rfc3748:2392-2396	不在范围内	快速重连属于恢复，任务将其排除在外
P048	rfc3748:2398-2405	没有陈述目标	密码绑定针对隧道方法；EAP-TLS 不是隧道方法
P049	rfc3748:2407-2411	A04, C01	
P050	rfc3748:2413-2417	不在范围内	分片与重组未被建模
P051	rfc3748:2419-2423	不在范围内	信道绑定需要低层信道性质，而这些处于被排除的边界
P052	rfc3748:2425-2427	没有陈述目标	它说明主张列表并不穷尽
P053	rfc3748:2431-2434	没有陈述目标	它说明 Identity 交换是可选的
P054 至 P055	rfc3748:2436-2449	X02	
P056	rfc3748:2451-2455	S04, U02	
P057	rfc3748:2459-2461	不在范围内	隧道化的 EAP 方法被任务排除在外
P058	rfc3748:2463-2471	没有陈述目标	它指出 EAP 不允许非隧道化的方法序列
P059	rfc3748:2473-2481	不在范围内	对 EAP 进行隧道化的恶意认证方不在所建模的完整握手之内
P060 至 P064	rfc3748:2483-2502	不在范围内	这些缓解措施针对 EAP 隧道方法，任务将其排除在外
P065	rfc3748:2506-2508	R01, R02, `network`	
P066	rfc3748:2510-2514	R02, `network`	
P067	rfc3748:2516-2520	A06, `network`	

段落	出处	定为	不选的原因
P068 至 P071	rfc3748:2522-2542	没有陈述目标	EAP-TLS 不使用逐包 MIC；TLS Finished 覆盖握手，而记录的答案正是关于这一点
P072 至 P074	rfc3748:2551-2567	A02, R02	
P075	rfc3748:2571-2576	没有陈述目标	针对口令方法的字典攻击
P076	rfc3748:2578-2580	没有陈述目标	它推荐抗字典攻击的方法
P077	rfc3748:2582-2587	不在范围内	它推荐隧道化，任务将其排除在外
P078	rfc3748:2591-2595	A02	
P079	rfc3748:2597-2598	没有陈述目标	它指出 EAP 不要求全双工
P080	rfc3748:2604-2611	A02	
P081	rfc3748:2615-2619	R05	
P082	rfc3748:2621-2629	没有陈述目标	它建议一个对等方名称映射到一个 EAP 方法
P083	rfc3748:2633-2634	没有陈述目标	低层对实现的依赖
P084	rfc3748:2636-2643	不在范围内	它涉及认证失败时 PPP 与 802.1X 的行为
P085	rfc3748:2645-2651	不在范围内	它涉及低层的重试与失败计数器
P086	rfc3748:2659-2665	A02, A03	
P087	rfc3748:2667-2671	没有陈述目标	它说明 MSK 与 EMSK 不直接用于保护数据
P088	rfc3748:2673-2677	不在范围内	AAA 服务器上的 AAA-Key 派生不在所建模的交换之内
P089	rfc3748:2679-2682	R01, A04	
P090	rfc3748:2684-2687	没有陈述目标	它说明 EAP 方法导出 MSK 与 EMSK，但不导出瞬时会话密钥
P091	rfc3748:2689-2693	没有陈述目标	它指出密码套件协商可能发生在其之前或之后
P092	rfc3748:2695-2699	R05, A08	
P093	rfc3748:2701-2704	证明器问不了	瞬时会话密钥的强度是计算性的
P094	rfc3748:2710-2713	证明器问不了	128 比特有效密钥强度是计算性的
P095	rfc3748:2715-2720	K01	
P096	rfc3748:2722-2729	证明器问不了	子串的密码学分离是密钥派生的计算性质
P097	rfc3748:2731-2735	K01	
P098	rfc3748:2737-2740	不在范围内	各方之间的密钥生命周期处理不在所建模的正常流程之内
P099	rfc3748:2742-2744	没有陈述目标	它说明本文稿未给出详细的密钥派生指导
P100	rfc3748:2746-2752	没有陈述目标	它建议复用已有的密钥派生函数
P101	rfc3748:2765-2770	不在范围内	低层的数据保护不在所建模的交换之内
P102	rfc3748:2772-2776	不在范围内	低层保护不在所建模的交换之内
P103	rfc3748:2778-2783	不在范围内	低层密码套件降级不在所建模的交换之内
P104	rfc3748:2785-2789	不在范围内	保护低层协商不在所建模的交换之内
P105 至 P110	rfc3748:2793-2828	不在范围内	PPP、IEEE 802 与 802.11 的链路层指示不在所建模的 EAP-TLS 交换之内
P111	rfc3748:2832-2838	A02, A03	
P112	rfc3748:2840-2842	不在范围内	认证方与后端服务器之间的分离正是任务置于范围之外的边界

段落	出处	定为	不选的原因
P113	rfc3748:2844-2847	不在范围内	对等方仅凭 EAP 无法验证认证方，这涉及认证方身份，而非此处建模的 EAP 服务器认证
P114	rfc3748:2849-2855	不在范围内	AAA 协议的保护处于被排除的边界
P115 至 P117	rfc3748:2857-2880	不在范围内	安全关联协议与 AAA-Key 传输处于被排除的边界
P118	rfc3748:2884-2887	没有陈述目标	它说明未定义明文口令认证
P119	rfc3748:2889-2892	不在范围内	RADIUS 中的封装不在所建模的交换之内
P120	rfc3748:2894-2898	没有陈述目标	EAP-TLS 不使用明文口令
P121 至 P127	rfc3748:2902-2945	不在范围内	低层标识符的信道绑定与带外信息处于任务排除的边界
P128	rfc3748:2949-2956	A06, R01, `network`	
P129	rfc3748:2958-2963	A06, `network`	
P130	rfc3748:2965-2969	A02, A06, `network`	
P131	rfc3748:2975-2983	证明器问不了	伪造的 Success 包是对对等方的拒绝服务，而可用性不是证明器所回答的内容
P132	rfc3748:2985-2989	证明器问不了	空闲超时是活性性质
P133	rfc3748:2991-2997	A06, U01	
P134	rfc3748:2999-3002	不在范围内	成功指示之前的授权策略不在所建模的交换之内
P135	rfc3748:3004-3014	不在范围内	与认证方之间的授权同步处于被排除的边界
P136	rfc3748:3016-3021	A06	
P137	rfc3748:3028-3030	没有陈述目标	它描述对等方何时丢弃 Failure 包
P138	rfc3748:3032-3037	A06, A02	
P139	rfc3748:3039-3046	证明器问不了	抗拒绝服务能力不是证明器所回答的性质
P140	rfc3748:3048-3056	A06, U01	
P141	rfc9190:992-993	A02, K01	
P142 至 P143	rfc9190:997-1005	A02	
P144	rfc9190:1007-1012	K01, X02	
P145	rfc9190:1014-1018	C01	
P146	rfc9190:1020-1024	A08, R05	
P147	rfc9190:1028-1029	A02	
P148	rfc9190:1033-1035	A02, `network`	
P149	rfc9190:1039-1040	A02, C01, `ltk_leak`	
P150 至 P153	rfc9190:1042-1083	不在范围内	证书吊销检查不属于所建模的消息流
P154	rfc9190:1087-1088	A06, R01, `network`	
P155	rfc9190:1090-1096	A06, U02, `network`	
P156	rfc9190:1098-1104	A06	

段落	出处	定为	不选的原因
P157	rfc9190:1108-1110	不在范围内	它涉及没有对等方认证的部署，任务将其排除在外
P158	rfc9190:1112-1121	A02	
P159	rfc9190:1123-1134	不在范围内	来自封装协议的非 EAP 信息处于被排除的边界
P160	rfc9190:1136-1143	S04, U02, `network`	
P161	rfc9190:1145-1147	不在范围内	基于非 EAP 信息进行拒绝不在所建模的交换之内
P162	rfc9190:1149-1155	A02	
P163 至 P172	rfc9190:1159-1240	不在范围内	第 5.7 节整节都涉及恢复，任务将其排除在外
P173 至 P174	rfc9190:1244-1249	X02	
P175	rfc9190:1251-1255	X02, `network`	
P176	rfc9190:1257-1272	证明器问不了	通过包大小与填充造成的泄露是长度侧信道，不是符号性质
P177	rfc9190:1274-1280	X02	
P178	rfc9190:1282-1293	X02, R05, `network`	
P179	rfc9190:1295-1312	X02	
P180	rfc9190:1316-1316	没有陈述目标	它引入关于普遍监控的章节
P181	rfc9190:1318-1328	X02	
P182	rfc9190:1330-1335	C01	
P183	rfc9190:1339-1339	没有陈述目标	它引入早先 TLS 攻击的列表
P184	rfc9190:1341-1351	没有陈述目标	它告诉实现遵循当前指导并指向相关文稿
P185	rfc9190:1355-1355	没有陈述目标	它引入跨协议恢复
P186 至 P187	rfc9190:1357-1372	不在范围内	跨协议恢复属于恢复，任务将其排除在外
P188	rfc9191:487-490	A02	
P189	rfc9191:492-493	不在范围内	证书压缩未被建模
P190	rfc9191:495-496	没有陈述目标	它指向所引用文稿的安全考虑
P191	rfc9846:4676-4677	A02, K01	

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 31 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 6 项，没设的 25 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	EAP peer	参与方	`ltk_leak`	
S02	EAP server	参与方	`ltk_leak`	
S03	EAP authenticator	参与方	攻击者得不到更多	认证者只转发 EAP 报文：它收到的 MSK 按定义在保密目标下就是失守的，而且它并不持有任何能伪造 TLS 握手的东西，所以剩下的正好是认证目标所依据的网络攻击者

ID	参与方或值	类别	攻击者条件	不设的原因
S04	peer 证书与私钥	值	`ltk_leak`	
S05	server 证书与私钥	值	`ltk_leak`	
S06	对端身份	值	按定义全部失守	持有对端身份正好就是对端身份保密目标的失守，而这个目标是在网络攻击者下回答的
S07	transcript hash	值	攻击者得不到更多	transcript hash 是公开的握手中值
S08	exporter_secret	值	按定义全部失守	exporter secret 可以导出 MSK，所以提到它的保密目标按定义就是失守的；认证目标不依赖它，是在网络攻击者下回答的
S09	Key_Material/MSK	值	按定义全部失守	MSK 就是保密目标所讲的值；持有它按定义就会使这些目标失守
S10	Finished verify_data	值	攻击者得不到更多	Finished verify_data 在握手中传输
S11	transcript	值	攻击者得不到更多	transcript 是公开的握手数据
S12	会话密钥	值	按定义全部失守	TLS 会话的通信密钥正是保密目标所保护的对象
S13	Finished	值	攻击者得不到更多	Finished 消息在握手中传输
S14	协商组	值	攻击者得不到更多	协商出的组是公开的
S15	签名算法	值	攻击者得不到更多	协商出的签名算法是公开的
S16	MSK	值	按定义全部失守	MSK 保密本身就是目标
S17	EMSK	值	按定义全部失守	EMSK 与 MSK 来自同一份 Key_Material；持有它就会使关于导出的密钥材料的那条保密目标失守
S18	Key_Material	值	按定义全部失守	Key_Material 就是保密目标所讲的导出密钥材料
S19	peer 私钥	值	`ltk_leak`	
S20	server 私钥	值	`ltk_leak`	
S21	会话载荷	值	按定义全部失守	会话载荷在会话密钥下传输；持有它就是机密性目标的失守
S22	共享秘密	值	按定义全部失守	混合共享秘密可以导出全部会话密钥，所以保密目标按定义就是失守的；它就是分别交给两个分量攻击者的那个值
S23	0x00 成功指示	值	攻击者得不到更多	0x00 指示是对端自己的成功信号，不是秘密
S24	随机数	值	攻击者得不到更多	这些新鲜值在握手中传输
S25	client Finished	值	攻击者得不到更多	client Finished 在握手中传输
S26	协商版本	值	攻击者得不到更多	协商出的版本是公开的
S27	密钥交换组	值	攻击者得不到更多	密钥交换组是公开的
S28	AEAD 密钥	值	按定义全部失守	AEAD 密钥由会话密钥导出；持有它们按定义就会使机密性目标失守
S29	nonce/IV	值	攻击者得不到更多	nonce 随记录一起传输
S30	EAP 标识	值	攻击者得不到更多	外层 EAP 标识在 Identity Response 中明文发送
S31	TLS 认证身份	值	攻击者得不到更多	TLS 认证身份就是证书身份，双方都知道，用于路由和授权

每条性质在各攻击者条件下的答案

攻击者由分析者设定，在编写查询之前就已确定；每个答案都记录在其中一个攻击者之下。单元格是证明器在该攻击者下对相应性质的答案；“无答案”是指为该性质设定的攻击者下，证明器让某条主张处于未决状态，或没有记录任何结果；空单元格表示分析者没有为该性质设定这个攻击者。

攻击者	持有什么	依据	出处
`network`	只有网络：它能读取、丢弃、修改和发送消息，不持有任何一方的秘密		
`ltk_leak`	peer 证书和 server 证书的 ML-DSA-65 私钥，在会话之前或期间恢复	main s0014 和 s0018：在证书有效期内，恢复私钥的攻击者可以实时冒充证书持有者；RFC 9190 第 5.4 节将密钥泄露列为吊销原因。	s0014, s0018
`x25519_broken`	一次已记录握手的混合密钥协商中的 X25519 部分（临时标量或 X25519 共享秘密）	main s0012：混合共享秘密只有在两个分量算法都未被攻破时才保持安全；RFC 10024 s0644 指出，CRQC 观察者可以恢复 ECDH 标量。	s0012, s0652
`mlkem_broken`	一次已记录握手的混合密钥协商中的 ML-KEM-768 部分（共享秘密或解封装密钥）	main s0012 和 s0010/s0652：X25519MLKEM768 的 ML-KEM-768 分量。	s0012, s0652

ID	`network`	`ltk_leak`	`x25519_broken`	`mlkem_broken`
A02	不成立	不成立		
A03	无答案			
A04	无答案			
A06	无答案			
A08	无答案			
K01	成立			
C01		无答案		
C08			无答案	无答案
R01	无答案			
R02	无答案			
R05	无答案			
S01	无答案			
S04	无答案			
A03.14	无答案			
U01	成立			
U02	无答案			
X01		不成立		
X02	成立			

原文和目标

Source: 655 units, 182927 bytes, frozen in source-delivery-01.json. Targets: 15 approved (A02, A03, A04, A06, A08, K01, C01, C08, R01, R02, R05, S01, S04, U01, U02); the questions asked: 3. The list of targets was approved as drafted, beforehand, with --approve-targets.

宿主按其文档化的默认值做出的决定（任务 `decisions: auto`），供你核对：

- approve_package: 批准源文包 protocol-dd1026aa855b81ab805a71240f38

这一步自身报告的情况：1 次请求（1 次失败），0.000 USD，输入 80177 tokens，输出 5710 tokens，32 s。

关于证明器和模型的说明

- A02：在 ProVerif 中回答，分析者为该目标指定了 dh_algebra
- A03：在 ProVerif 中回答，分析者为该目标指定了 dh_algebra
- A08：在 ProVerif 中回答，分析者为该目标指定了 dh_algebra
- K01：在 ProVerif 中回答，分析者为该目标指定了 dh_algebra
- C01：在 ProVerif 中回答，分析者为该目标指定了 dh_algebra
- C08：在 ProVerif 中回答，分析者为该目标指定了 dh_algebra
- R05：在 ProVerif 中回答，分析者为该目标指定了 dh_algebra
- S01：在 ProVerif 中回答，分析者为该目标指定了 dh_algebra
- S04：在 ProVerif 中回答，分析者为该目标指定了 dh_algebra
- U01：在 ProVerif 中回答，分析者为该目标指定了 dh_algebra
- U02：在 ProVerif 中回答，分析者为该目标指定了 dh_algebra
- X01：在 ProVerif 中回答，分析者为该目标指定了 dh_algebra
- X02：在 ProVerif 中回答，分析者为该目标指定了 dh_algebra

简化代数的 ProVerif 模型（xor、dh_algebra、other_algebra）给攻击者的能力比真实运算少：在这种模型里的证明是对更弱攻击者的证明，而这里的攻击仍然是攻击。省略了对变化值的检查（changing_state）的 ProVerif 模型会让参与方接受更多：在这种模型里的证明成立，而这里的攻击可能只是缺少了该检查。Tamarin+ Agent 按原样对这些建模。

分析者为各目标指定的内容：

step	targets	features	reason
31	A02, A03, A04, A06, A08, K01, C01, C08, R01, R02, R05, S01, S04, A03.14, U01, U02, X01, X02	dh_algebra, plain_primitives, many_sessions	混合共享秘密被写成一个 KEM 秘密与一个满足单一规律 g^xy = g^yx 的 Diffie-Hellman 项组成的对，所以凡是提到会话密钥或 transcript 的答案都依赖这一简化；ProVerif 不会在 transcript hash 内部把 peer 的 exp(exp(g,x),y) 与 server 的 exp(exp(g,y),x) 视为相同，未决结果和若干被证伪的结果正源于此。密钥派生、AEAD、KEM 和签名其余部分都是普通原语，按其通常的完美密码学规律建模。这些性质是在任意数量的双方角色会话上询问的，ProVerif 确实覆盖这一点，但各次运行的指纹被一起哈希，所以在这一形态下无法判定单次运行上的一致性。

分析者的结语

纯正常流程的模型（peer、EAP server、X25519MLKEM768 混合共享秘密、TLS 1.3 密钥调度、TLS exporter、基于转录本的 Finished、受保护的 0x00 成功指示、EAP-Success）位于 model.pv 中，其正常流程已被找到。全部 18 个目标都有答案，或者有明确记录表明没有返回答案。对任意数量的会话都证明：EAP-Success 在已接受的客户端 Finished 和受保护的 0x00 之后出现；证书签名密钥无法从网络获取；被动攻击者无法获得客户端证书中的身份。两个攻击复现了文稿所述内容：持有已恢复服务器签名密钥的攻击者向 peer 冒充 EAP server；仅位于网络上的攻击者使 peer 接受服务器从未发送的 EAP-Success，因为 EAP-Success 不携带任何保护。未发现新的弱点。六对目标和攻击者仍处于开放状态，若干答案被标记为模型的产物：混合共享秘密仅以 g^xy = g^yx 这一规律写入，ProVerif 无法在转录本哈希中识别 peer 和服务器对它的不同写法，因此单次运行中的一致性、单射性、新鲜性、协商参数、nonce 规则以及混合组件相关主张都无法判定；模型还允许任意攻击者取得所建模的组件披露，这就是前向保密和混合目标没有可用答案的原因。阻碍其余部分的是该算术问题（Tamarin+ Agent 可以解决它），以及一个按阶段区分披露的模型。不需要有界模型，也不需要状态审查：没有任何目标依赖于会变化的值。