SSH 的安全分析：ML-KEM/X25519 混合 KEX 与主机认证

本报告的其他语言版本：English。

rfc10042 · ProVerif+ Agent 报告

文稿	Post-Quantum/Traditional Hybrid Key Exchange with the Module-Lattice-Based Key-Encapsulation Mechanism for Use in SSH （协议定义）; The Secure Shell (SSH) Transport Layer Protocol （辅助）
检查的性质	19 条：18 条成立，1 条不成立（1 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

协议是 SSH 的传输层密钥交换，采用混合方法 mlkem768x25519-sha256。客户端和服务器先交换各自的算法名称列表，然后执行一次 X25519 密钥协商和一次 ML-KEM-768 封装，服务器用其 host 密钥对得到的 exchange hash 签名。客户端在这一层不持有长期密钥；服务器持有一对 host 密钥，客户端应当事先知道服务器的公开 host 密钥。

交换结束时，双方都持有共享秘密 K、作为会话标识的 exchange hash H，以及两个方向的加密密钥和完整性密钥。客户端确信对端持有服务器的 host 私钥，并且它哈希过的每个值都来自该服务器的这一次运行。服务器在这一层对客户端的身份没有任何确信，因为用户认证是单独的协议。

参与方：

- **客户端 C**：在这一层不持有长期密钥；每次连接都生成新的 X25519 和 ML-KEM-768 密钥对，并事先知道服务器的公开 host 密钥。持有：V_C；服务器的公开 host 密钥。
- **服务器 S**：持有 host 密钥对，其公开部分为 K_S。持有：V_S；host 密钥对（host 私钥，K_S）。

图 1 展示了文稿所规定的交换过程。

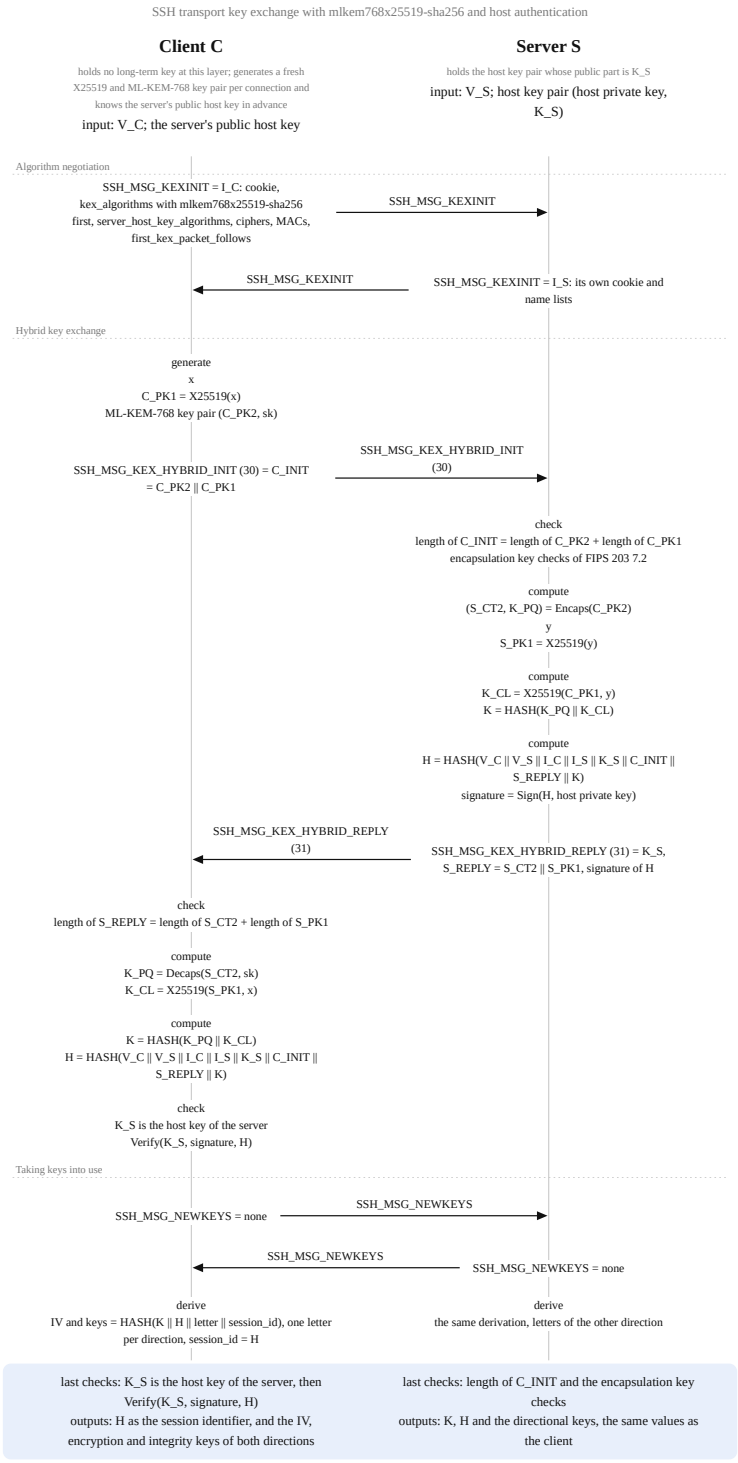


图 1. 文稿规定的协议。

分析的范围见附录 D：分析了什么、省略了什么，以及依据的是文稿的哪些章节。

2 发现

没有发现。 分析没有发现文稿尚未写明或尚未排除的任何问题。第 3 节的表格给出了每条性质的结果。

在所分析的范围内，混合密钥交换给出了它应当给出的东西：共享秘密和会话密钥不会落入攻击者之手，服务器认证成立并绑定到会话，一个组件失效不会泄露密钥。分析没有发现文稿尚未写明的弱点。它复现了文稿写明的那一个弱点：客户端若不将 host 密钥与已存储的密钥比较，就会接受主动攻击者，且复现出

的正是该攻击的确切条件。每个证明都带有对照查询，一旦移除其所依赖的保护，对照查询就会失败，因此这些验证结果并非空转。这些结果依赖于模型将哈希、签名和 ML-KEM 视为理想原语、X25519 仅带有 g^xy = g^yx 这一规律；长度检查、记录层和重新交换均未建模。

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
K01	**混合共享密钥保持保密。** 攻击者不能从协议交互中推导出共享秘密 K、K_CL、K_PQ 以及由 K 派生的会话加密密钥和完整性密钥。	无	成立	中	是
A02	**host 密钥属于真实服务器。** 客户端每次接受服务器主机密钥与签名时，都存在一个真实运行的服务器，其意图身份与本客户端一致。	无	成立	中	是
A03	**exchange hash 输入达成一致。** 客户端接受时，服务器真实运行中使用的 V_C、V_S、I_C、I_S、K_S、C_INIT、S_REPLY 与 K 与客户端本地值一致。	无	成立	中	是
A04	**每次接受均为独立运行。** 客户端每次成功接受都对应不同的服务器运行；同一服务器运行不能被客户端接受两次。	无	成立	中	是
A05	**无第三方获得 K。** 除客户端与服务器外，任何其他主体都不能获得 K 或由 K 派生的会话密钥。	无	成立	低	是
A08	**协商参数绑定到 H。** 客户端接受的会话参数（kex 方法、主机密钥算法、哈希）与双方 KEXINIT 中协商的结果一致，且被 H 绑定。	无	成立	中	是
R05	**接受的方法满足两项策略。** 接受的 kex 方法与哈希满足双方策略的最低要求，且协商结果被 H 与签名认证。	无	成立	低	不确定
C01	**针对 host 密钥的前向保密。** 会话完成并擦除临时状态后，即使服务器主机私钥泄露，攻击者也不能得到该会话的 K 或会话密钥。	服务器主机私钥 (NEWKEYS 完成并擦除临时密钥之后)	成立	中	是
C04	**单次临时泄露后 K 仍保密。** 在 X25519 临时私钥或 ML-KEM 临时私钥之一泄露时，K 与派生会话密钥仍对攻击者保密。	X25519 或 ML-KEM 临时私钥 (密钥交换进行中)	成立	中	不确定
C08	**单组件泄露后 K 仍保密。** 在 K_CL 或 K_PQ 之一被攻击者获知时，K 与派生会话密钥仍保密。	K_CL 或 K_PQ (密钥交换进行中)	成立	中	不确定
D01	**回复内容真实。** 客户端接受的 HYBRID_REPLY 内容（K_S、S_REPLY、签名）确实由被声称的服务器发送且未被篡改。	无	成立	低	是
A07	**接受的 host 密钥属于服务器。** 客户端接受的主机密钥 K_S 的主体与被声称的服务器身份一致，签名由该主机密钥对应私钥产生。	无	成立	低	是

ID	性质	假定的泄露	结果	价值	文稿是否已写
K04	**每方向每会话密钥互异**。客户端到服务器与服务器到客户端的加密/完整性密钥不同，不同会话的密钥也不同。	无	成立	低	是
K01.14	**攻击者无法推导 K 或密钥**。攻击者不能推导出 K 及派生的会话密钥。	无	成立	低	是
A03.15	**exchange hash 绑定所有输入**。客户端接受时，H 中绑定的版本、协商内容、主机密钥与双方临时公钥与服务器真实运行一致。	无	成立	低	是
A02.16	**服务器认证绑定到会话**。客户端接受服务器主机密钥与签名时，对端确实是以本客户端为对象运行的服务器，且认证绑定到本次会话的 H。	无	成立	低	是
X01	**每会话新鲜 ML-KEM 封装**。两次服务器封装不会产生相同的 S_CT2，除非它们封装的是同一个秘密 K_PQ；特别地，ML-KEM 密文及其随机数绝不会跨会话重用（查询 g1）。	无	成立	低	是
X02	**host 密钥验证阻止冒充**。如果客户端不将 K_S 与事先已知的 host 密钥比较，那么拥有自己 host 密钥对的主动攻击者就能与客户端完成密钥交换并被接受为服务器（查询 g2 必须失败）。	无	攻击成立（文稿已写明）	中	是
X03	**exchange hash 保持保密**。作为会话标识的 exchange hash H，不能被网络攻击者从交换中推导出来（查询 h_secret 必须成立）。	无	成立	低	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, X03。

提出的问题，以及分析者的回答：

问题	回答	依据
Q1: 共享秘密与会话密钥保密	是：在每个会话中，K、两个组件秘密以及两个方向的会话密钥都只留在客户端和服务器手中，泄露一个组件秘密或一个临时私钥不会泄露 K；只有泄露一个会话的全部临时状态才会。	K01, C04, C08
Q2: exchange hash 绑定版本、协商内容、host key 与双方 share	是：客户端验证的 exchange hash 覆盖了两个标识字符串、两个名称列表（因而覆盖协商出的方法和哈希）、host 密钥、两个交换值和共享秘密，模型表明其中每个字段都等于服务器自己的值。	A03, A03.15, A08, X03
Q3: 服务器认证与会话绑定	是，前提是客户端将所提供的 host 密钥与它已知的某个密钥比较：此时被接受的 host 密钥和签名意味着存在一次使用了本会话交换值的服务器运行，因此认证绑定到会话，而不可重放。	A02, A02.16, A07, A04, X02

4 本次分析的局限

这里的结果是针对协议模型的结果：它说明的是规定的消息和检查给出了什么或未能给出什么，其中密码学被视为理想原语，而与实现无关。

按分析者的陈述，本次分析不覆盖以下内容：

- 对 C_INIT 和 S_REPLY 的长度检查（main s0009, s0010）用类型化解析建模，而不是用长度建模，因此那些移动 C_PK2 与 C_PK1 之间边界或扩展 S_CT2 的攻击不在模型范围内。
- ML-KEM 被建模为封装一个新秘密的 KEM；FIPS 203 的隐式拒绝（解封装返回伪随机密钥而不是失败）、封装密钥与模数检查以及精确编码均未建模。
- X25519 只带有规律 g^xy = g^yx：小阶公钥、全零共享秘密以及 RFC 5656 第 4 节的有效性检查均未建模，也没有其他曲线代数可用。
- 哈希、密钥派生和签名是不透明函数；RFC 4253 7.2 中超出第一个块的密钥扩展以及 main s0047 的 IND-CCA2 假设属于密码学假设，而不是所证明的内容。
- 算法协商固定为 mlkem768x25519-sha256 与一种 host 密钥算法：rfc4253 s0065 的名称列表选择、first_kex_packet_follows 以及降级到其他方法均未建模，模型也没有最低策略集。
- 只存在一个服务器 host 密钥，且客户端事先知道它；证书或 known-hosts 的分发以及首次接触时对该存储的更新均未建模。
- 重新交换、用户认证和记录层（数据包重放、MAC、序列号、重新密钥）不在范围内。
- 关于进展、终止、资源开销、时序、流量分析或侧信道的问题既未提出也未回答。
- 会话密钥派生为 kdf(K, H, letter)；已证明两个方向和不同会话绝不会重合，但该派生的强度是假定的。

每条性质还有哪些问题未决、怎样才能解决，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，而不是证明器的回答：这些回答对协议意味着什么、解读依据什么、能否照原样使用，以及它的价值如何。

K01: 混合共享密钥保持保密

攻击者不能从协议交互中推导出共享秘密 K、K_CL、K_PQ 以及由 K 派生的会话加密密钥和完整性密钥。

网络攻击者以及客户端和服务器以外的任何一方都无法获得 K、两个分量秘密或派生的会话密钥。每个分量秘密由交换中各自的那一半保护，因此只有当某人在同一会话中同时攻破 X25519 和 ML-KEM-768 交换时，才能触及会话密钥。

- 结果：成立；所问是否有答案：是；有保留地可用
- 价值中：这是该方法的核心保密保证，它依赖理想化的原语，而不依赖对协议本身的任何简化。
- 文稿中已陈述：是。文稿中的位置：main:425、s0034、s0047
- 依据：ML-KEM 被建模为在客户端公钥下封装一个新秘密，不含 FIPS 203 的隐式拒绝及其输入检查。；X25519 只承载 g^xy = g^yx 这一法则。；哈希和密钥派生是不透明的单向函数，因此无法使用长度扩展或随机预言机推理。；会话密钥是 kdf(K, H, letter)，而不是完整的 RFC 4253 7.2 密钥扩展。；协商固定为这一种方法，因此模型中没有其他密码套件。；该主张是对一个绝不应出现的情形提出的保密查询，其对照查询会交出会话的全部临时状态，并且确实能到达 K。
- 下一步：一个带有真实 ML-KEM 与 X25519 模型的基于等价性的证明器，可以在不引入理想化的情况下判定该主张。

A02: host 密钥属于真实服务器

客户端每次接受服务器主机密钥与签名时，都存在一个真实运行的服务器，其意图身份与本客户端一致。

客户端持有服务器 host 公钥并验证签名时，只有当某个服务器运行针对同一次交换产生了正是该回复，客户端才会接受。网络上的攻击者无法让客户端接受没有任何服务器发送过的 host 密钥或签名；对照查询表明，一旦略去对 host 密钥的比较，同一主张就会不成立。

- 结果：成立；所问是否有答案：是；有保留地可用
- 价值中：服务器 host 认证是传输层存在的意义所在；该证明覆盖协议逻辑，但不覆盖它所依赖的 host 密钥分发。
- 文稿中已陈述：是。文稿中的位置：rfc4251:652、s0112
- 依据：签名的建模使得客户端可以验证签名，但攻击者无法从签名中恢复被签名的哈希。；客户端对 host 公钥的知晓是给定的；没有对证书或 known-hosts 基础设施建模。；只存在一个 host 密钥，因此客户端所确信的身份是单一服务器。
- 未决：客户端接受其从未见过的 host 密钥这一情形不属于本主张；关于 host 密钥验证的附加目标覆盖了它。
- 下一步：对首次联系时写入的 host 密钥存储建模，以陈述首次连接能保证什么。

A03: exchange hash 输入达成一致

客户端接受时，服务器真实运行中使用的 V_C、V_S、I_C、I_S、K_S、C_INIT、S_REPLY 与 K 与客户端本地值一致。

客户端哈希进 exchange hash 的每个字段都是与其通信的服务器运行中的值，因此被签名的哈希把两个标识字符串、两个名称列表、host 密钥、两个交换值和共享秘密绑定到同一个会话。攻击者对其中任何一个的改动都会使签名验证失败。

- 结果：成立；所问是否有答案：是；有保留地可用
- 价值中：对 exchange hash 输入的一致性是签名具有意义的前提，模型逐字段展示了这一点。
- 文稿中已陈述：是。文稿中的位置：s0040、s0041
- 依据：哈希是单射的不透明函数，因此对 H 的一致性被视为对其全部输入的一致性。；两个标识字符串和名称列表在模型中是不透明值。
- 下一步：在范围内无需下一步。

A04: 每次接受均为独立运行

客户端每次成功接受都对应不同的服务器运行；同一服务器运行不能被客户端接受两次。

一个服务器回复至多被接受一次：由于每个会话都使用新的临时密钥，且客户端自身的交换值进入哈希，录制的回复在另一个会话中无法通过验证。因此每次接受都对应其自己的服务器运行。

- 结果：成立；所问是否有答案：是；有保留地可用
- 价值中：这是密钥交换的重放主张，以单射方式而非简单对应关系展示。
- 文稿中已陈述：是。文稿中的位置：rfc4251:857、s0090
- 依据：新鲜性来自新的临时值；不涉及计数器或时间戳。；哈希被视为无碰撞。
- 下一步：在范围内无需下一步。

A05: 无第三方获得 K

除客户端与服务器外，任何其他主体都不能获得 K 或由 K 派生的会话密钥。

只有会话的两个端点能计算 K；第三方，无论是另一个客户端、另一个服务器还是网络上的某人，都被排除在外。模型在密钥交换中没有第三个角色，因此攻击者代表任何其他一方。

- 结果：成立；所问是否有答案：是；有保留地可用
- 价值低：这与 K 的保密性由同一查询回答；它没有增加单独的内容。
- 文稿中已陈述：是。文稿中的位置：rfc4251:652
- 依据：模型在公共信道上有一个客户端角色和一个服务器角色；没有第三个参与实体。；与 K 的保密性相同的理想化适用。
- 下一步：如果要检查带中间方的协议变体，需增加第三方角色。

A08: 协商参数绑定到 H

客户端接受的会话参数（kex 方法、主机密钥算法、哈希）与双方 KEXINIT 中协商的结果一致，且被 H 绑定。

客户端接受的两个名称列表就是服务器自己的，且它们被签名的 exchange hash 覆盖，因此任何对端都不可能以不同参数运行该会话。在模型中方法本身固定为 mlkem768x25519-sha256、哈希固定为 SHA-256，因此该主张关乎绑定，而非关乎选择。

- 结果：成立；所问是否有答案：是；有保留地可用
- 价值中：把协商参数绑定进被签名的哈希正是阻止降级的手段，而这一点对模型所携带的参数得到了展示。

- 文稿中已陈述：是。文稿中的位置：s0016、s0040、s0041
- 依据：模型只有单一密钥交换方法和单一哈希，名称列表是不透明载荷。；未对 rfc4253 s0065 的选择过程建模。
- 未决：要表明降级到更弱方法会被检测到，而不仅仅是参数被绑定，需要一个包含多种方法的模型。
- 下一步：加入 main s0017-s0019 中的三个 PQ/T 方法名以及经典方法，然后询问被接受的会话能否使用某一方未列出的方法。

R05: 接受的方法满足两项策略

接受的 kex 方法与哈希满足双方策略的最低要求，且协商结果被 H 与签名认证。

能够展示的是，会话运行所用参数经过认证：客户端接受的名称列表是服务器的，并被签名覆盖。接受的方法是否满足最低强度策略则没有得到回答，因为模型固定了唯一的方法，而文稿也没有为其设定最低要求。

- 结果：成立；所问是否有答案：部分；有保留地可用
- 价值低：得到回答的那一半由名称列表的绑定推出；强度策略那一半超出本模型所能陈述的范围。
- 文稿中已陈述：不明确。所依据的原文：rfc4251:1046、s0016
- 依据：模型中只有一种固定方法和一种哈希，没有策略阈值。；强度排序是策略问题，迹证明器无法权衡。
- 未决：文稿中没有为该方

如果只泄露某个会话的一把临时私钥，共享秘密仍然无法获得：仅凭 X25519 指数无法得到 K_PQ，仅凭 ML-KEM 私钥也无法得到 K_CL。只有同一会话的两半都被泄露才会给出秘密，这一点由对照查询确认。

- 结果：Holds；是否回答了所问：是；可用但有保留
- 价值 中：这正是该方法所针对的混合性质，失败情形由对照查询展示出来。
- 文稿中是否说明：不明确。所依据的原文：main:425, s0034, s0035
- 依据：ML-KEM 秘密受 KEM 抽象保护，因此该主张的强度仅取决于把 ML-KEM-768 视为安全 KEM。；两个组件秘密由一个不透明单向哈希组合。
- 待解决：文稿引用的是密钥组合的外部证明，而没有以这些措辞说明这一单组件情形。
- 下一步：对组合做计算性证明，或使用具备完整 ML-KEM 模型的证明器，才能确定该主张的强度。

C08: 单组件泄露后 K 仍保密

在 K_CL 或 K_PQ 之一被攻击者获知时，K 与派生会话密钥仍保密。

知道两个组件秘密之一并不能得到共享秘密：已知 K_CL 时 ML-KEM 秘密仍然安全，已知 K_PQ 时 X25519 秘密仍然安全。因此混合方案在其两半之一失效时仍能存活，但两半同时失效则不能。

- 结果：Holds；是否回答了所问：是；可用但有保留
- 价值 中：它陈述了混合方案的兜底保证，这正是执行两次交换而非一次交换的理由。
- 文稿中是否说明：不明确。所依据的原文：main:425, s0047, s0048
- 依据：组合两个秘密的哈希被视为单向的。；经典部分用单一 Diffie-Hellman 定律建模，因此可以把 K_CL 强制为已知值（如全零点）来表示，但本应阻止这种情况的曲线检查未被建模。
- 待解决：文稿给出了目标并引用了证明，但没有以这些措辞说明单组件情形。
- 下一步：加入 X25519 全零和小阶检查，并再次询问被强制的经典秘密是否仍使密钥保密。

D01: 回复内容真实

客户端接受的 HYBRID_REPLY 内容（K_S、S_REPLY、签名）确实由被声称的服务器发送且未被篡改。

客户端接受的 host 密钥、交换值字符串和签名都是服务器为本会话产生的，因此回复中的任何内容都无法在传输中被修改或伪造。

- 结果：Holds；是否回答了所问：是；可用但有保留
- 价值 低：它是 host 认证主张的实用形式，并共用其查询。
- 文稿中是否说明：是。文稿中的位置：s0008, s0112
- 依据：与 host 认证主张相同的签名和哈希理想化。；回复被建模为一条消息；记录层分帧未被建模。
- 下一步：在范围内无需操作。

A07: 接受的 host 密钥属于服务器

客户端接受的主机密钥 K_S 的主体与被声称的服务器身份一致，签名由该主机密钥对应私钥产生。

客户端接受的 host 密钥就是其私钥签署了本会话交换哈希的那把密钥，也是客户端为该服务器持有的密钥。任何其他人提供的密钥都会在比较或签名检查中失败。

- 结果：Holds；是否回答了所问：是；可用但有保留
- 价值 低：它重述了 host 认证主张中身份的一面，而该面由同一查询回答。
- 文稿中是否说明：是。文稿中的位置：rfc4251:204, s0112
- 依据：每台服务器一把 host 密钥，且客户端事先知道它。；签名不会向攻击者泄露被签名的哈希。
- 下一步：在范围内无需操作。

K04: 每方向每会话密钥互异

客户端到服务器与服务器到客户端的加密/完整性密钥不同，不同会话的密钥也不同。

一个会话的两个方向绝不会得到相同的密钥，两个会话也绝不会派生出相同的客户端密钥：方向和新鲜的交换值都参与派生，因此密钥按方向和按会话分离。

- 结果：Holds；是否回答了所问：是；可用但有保留
- 价值 低：它展示了密钥分离，这是记录层的前提条件，但不是文稿着力论证的保证。
- 文稿中是否说明：是。文稿中的位置：s0090, s0092, s0100
- 依据：派生 kdf(K, H, letter) 被建模为在其参数上单射。；RFC 4253 7.2 的字母被建模为两个不同的常量。；对照查询表明相反陈述不成立，因此模型中的两个方向和两个会话确实不同。
- 下一步：在范围内无需操作。

K01.14: 攻击者无法推导 K 或密钥

攻击者不能推导出 K 及派生的会话密钥。

共享秘密及由其派生的会话密钥留在两个端点；网络上的攻击者既得不到秘密，也得不到任何会话的密钥。

- 结果：Holds；是否回答了所问：是；可用但有保留
- 价值 低：它是用户对保密性主张的重述，并由相同的查询回答。
- 文稿中是否说明：是。文稿中的位置：s0034

- 依据：与保密性主张相同的理想化：ML-KEM 的 KEM 抽象、单一 Diffie-Hellman 定律、不透明哈希和派生。
- 下一步：在范围内无需操作。

A03.15: exchange hash 绑定所有输入

客户端接受时，H 中绑定的版本、协商内容、主机密钥与双方临时公钥与服务器真实运行一致。

进入交换哈希的所有字段，包括两个名称列表和两个交换值，都是服务器运行自身的值，因此客户端验证的哈希绑定到本会话。

- 结果：Holds；是否回答了所问：是；可用但有保留
- 价值 低：它是用户对哈希输入一致性的重述。
- 文稿中是否说明：是。文稿中的位置：s0040
- 依据：哈希是不透明且抗碰撞的，因此对哈希的一致就是对输入的一致。
- 下一步：在范围内无需操作。

A02.16: 服务器认证绑定到会话

客户端接受服务器主机密钥与签名时，对端确实是以本客户端为对象运行的服务器，且认证绑定到本次会话的 H。

被接受的 host 密钥和签名意味着某个服务器运行参与了这次交换，因为客户端自己的交换值就在服务器签署的哈希之内。因此认证绑定到会话，无法被移到另一个会话。

- 结果：Holds；是否回答了所问：是；可用但有保留
- 价值 低：它是用户对 host 认证主张以会话绑定形式的重述。
- 文稿中是否说明：是。文稿中的位置：s0057, s0112
- 依据：客户端的会话值按会话新鲜，并进入被签署的哈希。；与上述相同的签名理想化。
- 下一步：在范围内无需操作。

X01: 每会话新鲜 ML-KEM 封装

两次服务器封装绝不会产生相同的 S_CT2，除非它们封装了相同的秘密 K_PQ；特别地，ML-KEM 密文及其随机性绝不会跨会话重用（查询 g1）。

每个会话的 ML-KEM 密文都不同，除非同一秘密被封装两次，因此文稿对封装所要求的新鲜性成立。若密文在两个会话中重复，就意味着封装秘密被重用，而模型不会到达这种情况；对照查询表明两个会话确实封装了不同的秘密。

- 结果：Holds；是否回答了所问：是；可用但有保留
- 价值 低：它直接由封装秘密的新鲜性推出，除保密性主张之外增加不多。
- 文稿中是否说明：是。文稿中的位置：main:396
- 依据：封装以新鲜秘密和单射密文构造子建模。；随机性本身在模型中不可见；只有对密文的后果可见。
- 下一步：在范围内无需操作。

X02: host 密钥验证阻止冒充

若客户端不将 K_S 与事先已知的 host 密钥比较，拥有自己 host 密钥对的主动攻击者会与客户端完成密钥交换并被接受为服务器（查询 g2 必须不成立）。

跳过将提供的 host 密钥与其已知密钥比较的客户端，会接受持有自己 host 密钥对的攻击者的回复。攻击者应答客户端的交换，自行计算共享秘密和交换哈希并签名，而客户端没有其他手段察觉。

- 结果：攻击成立（文稿中已有说明）；回答了所问：是；可用但有保留
- 价值中：它展示了文稿所述的弱点，针对确切的配置和确切的攻击者，这正是实现者判断该检查是否承重所需要的。
- 文稿中已有说明：是。文稿出处：rfc4251:984, s0112
- 依据：不进行比较的客户端角色被单独建模，完全符合 rfc4253 s0112 的允许。；攻击者被赋予自己的宿主密钥对，任何人都能生成这样的密钥对。；交换值被建模为攻击者可以自选其经典指数。
- 待解决：客户端被这样配置的频率有多高，以及密钥变化时向用户显示什么，不在模型范围内。
- 下一步：对首次连接即写入的 known-hosts 存储建模，看看首次连接承诺了什么。

X03: exchange hash 保持保密

exchange hash H 即会话标识符，网络攻击者无法从交换中推导出它（查询 h_secret 必须成立）。

exchange hash 同时充当会话标识符，攻击者无法从交换中推导出它，因为它依赖于共享秘密，且只出现在攻击者无法打开的签名内部。因此，对它保密对传输密钥交换并不承重。

- 结果：成立；回答了所问：是；可用但有保留
- 价值低：文稿只是建议对哈希保密，传输交换离开它也不会停止。
- 文稿中已有说明：是。文稿出处：s0121
- 依据：签名被建模为攻击者无法从中恢复被签名的哈希。；哈希中的其他内容都是公开的，因此该主张只依赖于 K 的保密性。
- 下一步：在范围内无需下一步。

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，以及各次攻击及分析者对每次攻击的判定。

目标	级别	界	查询	对照	攻击	主张
K01	proved		k01_kcl	checked		5 proved, 1 reachable
A02	proved		a02	checked		1 proved, 1 reachable
A03	proved		a02	checked		1 proved, 1 reachable
A04	proved		a04	checked		1 proved, 1 reachable
A05	proved		k01_k	checked		1 proved, 1 reachable
A08	proved		a02	checked		1 proved, 1 reachable
R05	proved		a02	checked		1 proved, 1 reachable
C01	proved		c01	checked		2 proved, 1 reachable
C04	proved		c04_x	checked		2 proved, 2 reachable
C08	proved		c08_kcl	checked		2 proved, 2 reachable
D01	proved		a02	checked		1 proved, 1 reachable
A07	proved		a02	checked		1 proved, 1 reachable
K04	proved		k04b	checked		2 proved, 1 reachable
K01.14	proved		k01_k	checked		2 proved, 1 reachable
A03.15	proved		a02	checked		1 proved, 1 reachable
A02.16	proved		a02	checked		1 proved, 1 reachable
X01	proved		g1	checked		1 proved, 1 reachable
X02	reachable		reach_nocheck		g2 (holds_in_source)	1 reachable, 1 attack
X03	proved		h_secret	checked		1 proved, 1 reachable

逐条主张，按分析者对每条主张的措辞；每条主张的答案即证明器的答案：

目标	主张	攻击者	回答	查询	对照
K01	会话到达客户端持有 K、K_CL、K_PQ 以及两个会话密钥的阶段。		reachable	reach_secrets	
K01	网络攻击者无法推导出 K_CL（X25519 共享密钥），除非某个会话本身将其泄露。	network	proved	k01_kcl	checked
K01	网络攻击者无法推导出 K_PQ（ML-KEM-768 共享密钥），除非某个会话本身将其泄露。	network	proved	k01_kpq	checked
K01	网络攻击者无法推导出 K = HASH(K_PQ ‖ K_CL)。	network	proved	k01_k	checked
K01	网络攻击者无法推导出由 K 派生的客户端到服务器加密/完整性密钥。	network	proved	k01_kcs	checked
K01	网络攻击者无法推导出由 K 派生的服务器到客户端加密/完整性密钥。	network	proved	k01_ksc	checked

目标	主张	攻击者	回答	查询	对照
A02	客户端接受主机密钥和签名（主机认证步骤结束）。		reachable	reach_accept	
A02	每当客户端接受主机密钥和签名时，某个服务器运行针对相同的会话值恰好发送了该 K_S 和该交换哈希 H。	network	proved	a02	checked
A03	客户端在对 V_C、V_S、I_C、I_S、K_S、C_INIT、S_REPLY 和 K 进行哈希后接受。		reachable	reach_accept	
A03	客户端哈希进 H 的每个字段（I_C、I_S、K_S、C_INIT、S_REPLY 和 K）都等于产生该回复的服务器运行的值。	network	proved	a02	checked
A04	客户端接受一个回复（即询问其新鲜性的情形）。		reachable	reach_accept	
A04	客户端每次接受主机密钥和签名都对应一次不同的服务器运行：一次服务器运行的回复绝不会被接受两次。	network	proved	a04	checked
A05	会话到达 K 及其各组成部分存在的阶段。		reachable	reach_secrets	
A05	除会话的客户端和服务器之外，任何一方都无法获得 K：作为任何第三方出现的攻击者都无法推导出它。	network	proved	k01_k	checked
A08	客户端接受一个回复，因此 KEXINIT 载荷已被哈希进 H。		reachable	reach_accept	
A08	被接受的 I_C 和 I_S 携带名称列表，因而携带唯一的方法 mlkem768x25519-sha256，它们正是服务器所用的值，并被绑定进已签名的交换哈希。	network	proved	a02	checked
R05	客户端接受一个回复。		reachable	reach_accept	
R05	一方最终使用的协商参数是经过认证的：客户端接受的 I_C 和 I_S 是服务器自己的值，因此任何对等方都不可能为该会话使用不同的方法。	network	proved	a02	checked
C01	会话完成，随后服务器主机私钥被泄露（即前向保密主张所谈的情形）。		reachable	reach_c01	
C01	会话结束后，被泄露的主机私钥不能给出该会话的 K，除非该会话的全部临时状态（X25519 指数和 ML-KEM 私钥）都被泄露。	host_key_after	proved	c01	checked
C01	会话结束后，被泄露的主机私钥不能给出该会话的 K。	host_key_after	proved	c01	checked
C04	会话完成，且仅其 X25519 临时私钥被泄露。		reachable	reach_c04x	
C04	会话完成，且仅其 ML-KEM-768 封装私钥被泄露。		reachable	reach_c04sk2	
C04	当仅该会话的 X25519 临时私钥被泄露时，K 保持机密；ML-KEM 部分仍保护它。	eph_x25519_known	proved	c04_x	checked
C04	当仅该会话的 ML-KEM-768 封装私钥被泄露时，K 保持机密；X25519 部分仍保护它。	mlkem_sk_known	proved	c04_sk2	checked
C08	会话完成，且其 K_CL 被交给攻击者。		reachable	reach_c08kcl	
C08	会话完成，且其 K_PQ 被交给攻击者。		reachable	reach_c08kpq	
C08	当攻击者已知 K_CL 时，K 保持机密，因为 ML-KEM 部分仍保护它。	kcl_known	proved	c08_kcl	checked

目标	主张	攻击者	回答	查询	对照
C08	当攻击者已知 K_PQ 时，K 保持机密，因为 X25519 部分仍保护它。	`kpq_known`	proved	c08_kpq	checked
D01	客户端接受携带 K_S、S_REPLY 和签名的回复。		reachable	reach_accept	
D01	客户端接受的 K_S、S_REPLY 和签名恰好是某次服务器运行为该会话发送的值，因此回复中没有任何内容被改动或伪造。	`network`	proved	a02	checked
A07	客户端接受主机密钥和签名。		reachable	reach_accept	
A07	被接受的 K_S 正是其私钥产生了对 H 的签名的那把密钥，并且它就是客户端为其所通信的服务器所持有的主机密钥。	`network`	proved	a02	checked
K04	会话完成，且两个会话密钥均已派生。		reachable	reach_secrets	
K04	两个会话绝不共享客户端到服务器会话密钥：相同的密钥意味着相同的会话。	`network`	proved	k04b	checked
K04	一个会话的客户端到服务器密钥与服务器到客户端密钥绝不相等。	`network`	proved	k04a	checked
K01.14	会话到达 K 及派生密钥存在的阶段。		reachable	reach_secrets	
K01.14	攻击者无法推导出 K = HASH(K_PQ ‖ K_CL)。	`network`	proved	k01_k	checked
K01.14	攻击者无法推导出由 K 派生的会话加密/完整性密钥。	`network`	proved	k01_kcs	checked
A03.15	客户端在对各字段的自身副本进行哈希后接受。		reachable	reach_accept	
A03.15	客户端所哈希的标识字符串、KEXINIT 载荷、主机密钥、C_INIT、S_REPLY 和 K 都是服务器运行自己的值。	`network`	proved	a02	checked
A02.16	客户端接受主机密钥和签名。		reachable	reach_accept	
A02.16	当客户端接受主机密钥和签名时，某次服务器运行参与了正是这次交换（相同的 H），因此认证绑定到该会话。	`network`	proved	a02	checked
X01	发生一次服务器封装 S_CT2（会话到达封装步骤）。		reachable	reach_encap	
X01	除非两次服务器封装封装了相同的秘密 K_PQ，否则它们不会产生相同的 S_CT2；ML-KEM 密文及其随机性绝不会跨会话重用。	`network`	proved	g1	checked
X02	不将 K_S 与预先已知的主机密钥进行比较的客户端会接受任何主机密钥；会话到达该客户端的接受事件。	`network`	reachable	reach_nocheck	
X02	若不将 K_S 与已知主机密钥进行比较，持有自己主机密钥对的一个主动攻击者会被客户端当作服务器接受：被接受的回复从未由任何服务器运行发送过。	`network`	attack (holds_in_source)	g2（a02 的对照）	
X03	客户端在检查主机密钥和签名后接受一个回复，因此存在 H 的一个值。		reachable	reach_accept	
X03	交换哈希 H（会话标识符）在交换期间无法被网络攻击者推导出。	`network`	proved	h_secret	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，并把每条性质交给 ProVerif；第 3 节的每个结果都是证明器的回答。

ProVerif+ Agent 会话 session-20261114T183559，2026-11-14。状态：已完成。模型：deepseek-v4-1-flash-261017。步数：48，证明器 37 s，实际耗时 19 min，费用 0.141 USD。

ProVerif：运行了 144 个证明，141 个已判定，3 个无答案，32 s，记录了 70 个回答。

分析了什么

协议	SSH：ML-KEM/X25519 混合 KEX 与主机认证
版本	rfc10042
分析范围	固定 mlkem768x25519-sha256；KEXINIT、混合密钥交换、host key 检查与 NEWKEYS。用户认证与重密钥单独扩展。
不在范围内	完整 transport/host key 语义来自 SSH 基线，不能只给扩展稿；不将服务器认证当作客户端用户认证。仅证明明确模型及攻击者条件下的性质，不能把缺失检查、完整安全信道或全诚实 NF 当作待证结论的替代。
原文	131 个编号原文单元（30345 字节），来自下面列出的各节。本报告中的 s0001 之类的引用就是这些原文单元之一

分析所依据的文稿各节：

文稿	章节	标题
main	2	PQ/T 混合密钥交换
main	2.1	PQ/T 混合密钥交换方法抽象
main	2.2	PQ/T 混合密钥交换消息编号
main	2.3	PQ/T 混合密钥交换方法名称
main	2.3.1	mlkem768nistp256-sha256
main	2.3.2	mlkem1024nistp384-sha384
main	2.3.3	mlkem768x25519-sha256
main	2.4	共享密钥 K
main	2.5	密钥派生
main	5	安全考虑
rfc4253	7	密钥交换
rfc4253	7.1	算法协商
rfc4253	7.2	密钥交换的输出
rfc4253	7.3	启用密钥
rfc4253	8	Diffie-Hellman 密钥交换
rfc4253	8.1	diffie-hellman-group1-sha1
rfc4253	8.2	diffie-hellman-group14-sha1

文稿	章节	标题
rfc4253	9	密钥重交换

源文包定位到的协议目的：固定 mlkem768x25519-sha256；KEXINIT、混合密钥交换、host key 检查与 NEWKEYS。用户认证与重密钥单独扩展。

对照文稿定性质

在写出查询之前，分析者把草拟的性质清单对照文稿本身对协议目的的表述进行了核对。名称、性质所涉及的对象以及文稿的说法都是分析者的解读；所引位置在文稿中确实存在。

ID	简称	对象	文稿怎么说	出处
K01	混合共享密钥保持保密	K = HASH(K_PQ ‖ K_CL)，分量机密 K_CL 和 K_PQ，以及由 K 派生的加密/完整性密钥，对网络攻击者保密，仅客户端和服务器知晓	main s0034-s0035 将 K 定义为两个分量机密的哈希，s0045 要求每个连接使用新的临时密钥对，rfc4253 s0092-s0100 从 K 和 H 派生会话密钥	s0034, s0035, s0045, s0092
A02	host 密钥属于真实服务器	客户端确信：证明持有 K_S 对应 host 私钥的对端，是参与了本次交换的一次服务器运行	rfc4253 s0107-s0112：密钥交换使用显式服务器认证，客户端验证 K_S 确实是 S 的 host key，并验证 H 上的签名	s0057, s0108, s0112
A03	exchange hash 输入达成一致	客户端哈希进 H 的值 V_C、V_S、I_C、I_S、K_S、C_INIT、S_REPLY 和 K，就是服务器所使用的值	main s0040-s0041 给出了 H 覆盖的确切拼接；rfc4253 s0112 要求客户端根据自己的副本重新计算 H 并检查签名	s0040, s0041, s0112
A04	每次接受均为独立运行	一次服务器运行的 S_REPLY 和签名不能被客户端接受两次，两个被接受的会话不能共享其会话值	rfc4253 s0090 首次密钥交换的 exchange hash 是唯一的会话标识符；rfc4251:857 警告具有相同会话 id 的两个会话可能被相互重放；main s0045 要求每个连接使用新的临时密钥	s0045, s0090
A05	无第三方获得 K	K 及其派生的密钥，对该会话的客户端和服务器之外的任何一方保密	rfc4251:652 传输协议提供机密信道；main s0034-s0035；rfc4253 s0108 共享机密不能由任一方单独确定	s0034, s0035, s0108
A08	协商参数绑定到 H	I_C 和 I_S 中携带的密钥交换方法和 host key 算法，因而 mlkem768x25519-sha256 和 SHA-256 的选择，绑定到被签名的 exchange hash	main s0016 和 s0019 固定了方法名称列表，s0032 为该方���固定 SHA-256，s0040-s0041 将 I_C 和 I_S 放入 H；rfc4253 s0061 定义了 KEXINIT 载荷	s0016, s0019, s0032, s0040, s0041
R05	接受的方法满足两项策略	一方最终使用的密钥交换方法和哈希，以及对端要求的强度排序	main s0016 列出了允许的方法名称；rfc4251:1046 建议按密码强度对名称列表排序，最强在前；rfc4253 s0065 固定了方法的选取方式	s0016
C01	针对 host 密钥的前向保密	已完成会话的 K 和会话密钥，对在会话之后获得服务器长期 host 私钥的攻击者保密	rfc4251:1021-1031 定义了完全前向保密，并指出 SSH 密钥交换提供该性质：即使长期密钥材料后来泄露，会话仍然安全；main s0045 要求每个连接使用临时密钥对	s0034, s0045
C04	单次临时泄露后 K 仍保密	K 和会话密钥，对仅获知一次会话的 X25519 临时私钥、或仅获知 ML-KEM-768 封装私钥的攻击者保密	main s0034-s0035 将两个分量机密组合成 K，s0047 指出该混合构造在某些假设下可证明；rfc4251:1037 指出泄露两个 DH 私有参数会泄露会话密钥	s0034, s0035, s0047

ID	简称	对象	文稿怎么说	出处
C08	单组件泄露后 K 仍保密	K 和会话密钥，对仅获知一次会话的 K_CL、或仅获知 K_PQ 的攻击者保密	main s0034-s0035 将两个分量机密哈希进 K，s0047 讨论了该组合的 IND-CPA/IND-CCA2 证明，s0048 讨论了 X25519 的选择	s0034, s0035, s0047
D01	回复内容真实	客户端接受的 K_S、S_REPLY 和签名，即服务器所发送且未被修改	main s0008 定义了回复字段，rfc4253 s0112 和 s0122 要求签名计算在 H 上，而 H 覆盖 K_S 和 S_REPLY	s0008, s0040, s0112, s0122
A07	接受的 host 密钥属于服务器	K_S 的主体以及产生签名的私钥，即客户端所要检查的内容	rfc4251:204-207 host key 用于验证客户端确实在与正确的服务器通信，这需要事先知道该密钥；rfc4253 s0112	s0041, s0112
K04	每方向每会话密钥互异	客户端到服务器和服务器到客户端的加密/完整性密钥，以及不同会话的密钥	rfc4253 s0090-s0100 给出了每个连接唯一的会话标识符，并为每个方向的密钥分配了不同的字母；rfc4251:673 指出每个会话密钥通过会话特定的随机数据绑定到会话	s0090, s0092, s0100
K01.14	攻击者无法推导 K 或密钥	K 及其派生的密钥，对网络攻击者保密	用户问题 Q1；main s0034-s0035 和 s0039	s0034, s0035, s0039
A03.15	exchange hash 绑定所有输入	客户端最终哈希的 H 的字段，与服务器自己的值相对照	用户问题 Q2；main s0040-s0041	s0040, s0041
A02.16	服务器认证绑定到会话	客户端对服务器身份的确认，绑定到本次会话的 H	用户问题 Q3；rfc4253 s0057-s0058 显式服务器认证，s0112	s0057, s0112
X01	每会话新鲜 ML-KEM 封装	服务器的 ML-KEM 密文 S_CT2 及其封装的机密 K_PQ	main s0045：实现在生成 ML-KEM 密文时不得重用随机性，并且必须为每个连接生成临时 ML-KEM 密钥对	main:396
X02	host 密钥验证阻止冒充	客户端对服务器的认证，以及客户端所依赖的 host key 关联	rfc4253 s0112：客户端也可以不经验证就接受 host key，但这样做会使协议在面对主动攻击时不安全；rfc4251:984：在不建立 host 与其 host key 之间可靠关联的情况下使用该协议本质上是不安全的	rfc4251:984, s0112
X03	exchange hash 保持保密	H，即 rfc4253 s0090 中的会话标识符	rfc4253 s0121：exchange hash 应当保密；rfc4253 s0090 使其成为会话标识符	s0090, s0121

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 98 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 20 段，没选 78 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:392-394	没有陈述目标	这是引用：它把 ECDH 部分交给 RFC5656 和 RFC8731 的安全考虑，这两份文稿在下面单独列出（P81-P92、P95-P98）
P02	main:396-406	K01, X01, `network`	
P03	main:408-423	证明器问不了	定长编码是为了抵御侧信道和时序攻击，而基于迹的证明器无法权衡这一点
P04	main:425-438	K01, C08	

段落	出处	定为	不选的原因
P05	main:440-449	证明器问不了	比较的是曲线实现和采用情况，也就是原语的强度和品质，不是协议性质
P06	rfc4251:648-650	没有陈述目标	这是对汇集起来的安全考虑的引言，本身没有目标
P07	rfc4251:652-655	K01, A02, K04, `network`	
P08	rfc4251:657-662	不在范围内	客户端用户认证，任务把它留给单独的扩展
P09	rfc4251:664-669	不在范围内	连接协议（多路复用、shell、转发），在传输密钥交换之外
P10	rfc4251:673-684	K04, X01	
P11	rfc4251:690-694	证明器问不了	实现可用的熵的数量，任何迹性质都无法表达
P12	rfc4251:698-701	不在范围内	显示文本时对终端控制字符的过滤，在密钥交换之外
P13	rfc4251:707-721	不在范围内	批量密码的选择和相对强度，即记录层，任务没有建模
P14	rfc4251:723-726	不在范围内	仅用于调试的 none 密码，记录层
P15	rfc4251:728-736	不在范围内	记录层加密的 CBC 模式弱点
P16 至 P31	rfc4251:743-818	不在范围内	CBC/IV 和 SSH_MSG_IGNORE 的讨论，全部是记录层加密的细节
P32 至 P34	rfc4251:822-835	不在范围内	记录层数据完整性、none MAC 以及 MAC 的序列号回绕
P35 至 P38	rfc4251:839-887	A04, K04, `network`	
P39 至 P40	rfc4251:891-903	A02, X02, `network`	
P41	rfc4251:905-932	A02, A07, X02, `network`	
P42 至 P43	rfc4251:934-962	A02, X02, `network`	
P44	rfc4251:964-982	不在范围内	第三种情况：会话建立后对数据包的操作，即记录层中的 MAC 安全
P45	rfc4251:984-989	A02, X02, `network`	
P46 至 P48	rfc4251:993-1009	证明器问不了	拒绝服务以及连接建立的成本，迹性质无法度量
P49	rfc4251:1013-1017	证明器问不了	通过填充和 ignore 消息建立的隐蔽信道，无法表达为迹性质
P50	rfc4251:1021-1035	C01, `host_key_after`	
P51	rfc4251:1037-1042	C04, `eph_x25519_known`, `mlkem_sk_known`	
P52	rfc4251:1046-1051	R05	
P53	rfc4251:1063-1071	证明器问不了	流量分析和消息长度，超出可以向符号证明器提出的问题
P54 至 P56	rfc4251:1075-1090	不在范围内	认证协议：客户端用户认证、其方法和本地策略
P57	rfc4251:1094-1100	不在范围内	在弱传输上应禁用哪些认证方法，属于用户认证规则

段落	出处	定为	不选的原因
P58	rfc4251:1102-1108	A02, X02, `network`	
P59	rfc4251:1114-1118	不在范围内	针对中间人运行认证协议时泄露的凭据
P60	rfc4251:1122-1139	不在范围内	用户认证期间的调试消息
P61 至 P66	rfc4251:1143-1182	不在范围内	已认证用户的本地安全策略和授权，在传输密钥交换之外
P67 至 P68	rfc4251:1186-1194	不在范围内	口令认证
P69	rfc4251:1198-1201	不在范围内	基于宿主的用户认证
P70 至 P71	rfc4251:1219-1232	不在范围内	连接协议的端点安全假设
P72 至 P74	rfc4251:1236-1251	不在范围内	代理转发
P75 至 P78	rfc4251:1255-1287	不在范围内	X11 转发
P79	rfc4253:1428-1431	K01, A02, K04, `network`	
P80	rfc4253:1433-1434	没有陈述目标	引用 SSH 架构文稿以获取完整的安全考虑
P81	rfc5656:708-712	没有陈述目标	引用 SSH 全局和椭圆曲线的安全考虑，这里在 rfc4251 和 rfc4253 下审阅
P82 至 P91	rfc5656:714-795	证明器问不了	曲线和哈希的选择与强度，即原语的强度
P92	rfc5656:797-803	C04, `eph_x25519_known`, `mlkem_sk_known`	
P93 至 P94	rfc6234:6499-6506	没有陈述目标	SHA 文稿对其函数在特定用途下的安全性不作任何断言
P95	rfc8731:165-166	没有陈述目标	引用继承来的安全考虑
P96 至 P98	rfc8731:168-190	证明器问不了	Curve25519 和 Curve448 的安全级别以及 mpint 侧信道，即原语强度和侧信道

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 29 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 5 条，没检查 24 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	不在范围内	这里分析的传输密钥交换不承载也不保护任何应用载荷
K03	强保密／秘密值不可区分性	证明器问不了	K 是计算出来的值，不是自由的私有名称，所以 ProVerif 无法对它提出不可区分性问题；密钥组合的 IND-CCA2 是密码学假设（main s0047），不是迹性质
A01	对端曾参与性（Aliveness）	A02	

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	不适用	SSH 传输密钥交换没有密钥确认消息；rfc4253 s0105 说明 NEWKEYS 的存在是为了仍能报告失败
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	不在范围内	密钥交换不执行任何业务操作
R04	消息接受顺序／窗口安全	不在范围内	接受窗口和顺序是记录层性质，在任务之外
C02	抗密钥泄露冒充（KCI）	不适用	传输层没有客户端长期凭据；唯一的长期密钥是服务器宿主密钥，其泄露是 C01
C03	已知密钥安全／泄露隔离	K04	
C05	泄露后安全恢复（PCS）	不在范围内	重新密钥和泄露后恢复是任务略去的单独重新交换扩展
C06	注册凭据泄露后的增强认证	不在范围内	用户认证和注册记录不在范围内
C07	抗离线字典攻击	不适用	密钥交换不使用口令或其他低熵秘密
Z01	授权执行正确性	不在范围内	传输密钥交换不授予任何授权
Z02	委托权限不扩张	不在范围内	传输密钥交换中没有委托
Z03	授权与身份／会话／操作绑定	不在范围内	传输密钥交换中没有令牌或授权决定
Z04	撤销与过期生效	不在范围内	传输密钥交换中没有撤销或过期
Z05	跨租户／跨服务安全隔离	不在范围内	传输密钥交换中没有租户或服务
S01	加密 nonce 唯一性	X01	
S02	重传／密钥重安装安全	不在范围内	重传和密钥重安装属于记录层和重新密钥，任务将其略去
S03	抗回滚与状态单调性	不在范围内	已安装密钥的回滚属于密钥重新交换，任务将其略去
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用户认证和重新密钥，属于单独的扩展
S05	终止与旧状态失效	不在范围内	连接终止和旧状态失效在密钥交换之外
P01	身份匿名性	不适用	服务器宿主身份按设计是公开、稳定的名称，在此交换中没有任何参与方是匿名的
P02	会话不可关联性	不适用	会话有意与服务器的长期宿主密钥关联，这正是宿主认证的意义所在
P03	用户／凭据存在性隐藏	不在范围内	隐藏用户或凭据是否存在属于用户认证
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恢复是进展性质，迹证明器无法被问及
V02	无非预期死锁	证明器问不了	无死锁是分支时间性质
V03	受限资源滥用／抗放大	证明器问不了	资源成本和放大在此模型中无法度量

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 23 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 18 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 C	参与方	攻击者得不到更多	由攻击者控制的客户端只持有它自己运行的会话的临时秘密。传输层对服务器不作任何承诺，恶意客户端无从破坏；也不会由此推出其他会话的秘密

ID	参与方或值	类别	攻击者条件	不设的原因
S02	服务器 S	参与方	`host_key_after`	
S03	K_CL	值	`kcl_known`	
S04	K_PQ	值	`kpq_known`	
S05	K = HASH	值	按定义全部失守	持有 K 的攻击者所持有的，正是 K 的保密性质所禁止的，没有任何目标在该性质下要求更弱的东西
S06	会话加密密钥与完整性密钥	值	按定义全部失守	会话密钥是 K 的函数；持有它们就是持有其保密性质所禁止的东西
S07	服务器主机密钥 K_S	值	攻击者得不到更多	主机公钥在回复中经网络明文发送
S08	主机签名	值	攻击者得不到更多	签名在回复中经网络明文发送
S09	交换哈希 H 的各组成字段	值	攻击者得不到更多	H 的每个字段除 K 外都经网络明文发送
S10	K	值	按定义全部失守	同 S05
S11	临时密钥对	值	`eph_x25519_known`, `mlkem_sk_known`	
S12	会话标识	值	按定义全部失守	持有会话标识恰好就是关于 H 保密的那个目标所禁止的；没有其他目标依赖 H 未知
S13	会话密钥	值	按定义全部失守	同 S06
S14	I_C	值	攻击者得不到更多	I_C 经网络发送
S15	I_S	值	攻击者得不到更多	I_S 经网络发送
S16	协商的方法名	值	攻击者得不到更多	协商的方法名是公开的，并出现在 KEXINIT 负载中
S17	H	值	按定义全部失守	同 S12
S18	K_S	值	攻击者得不到更多	K_S 是主机公钥，经网络发送
S19	S_REPLY	值	攻击者得不到更多	S_REPLY 经网络发送
S20	签名	值	攻击者得不到更多	签名经网络发送
S21	主机私钥	值	`host_key_after`	
S22	各方向会话密钥	值	按定义全部失守	同 S06
S23	H 的各组成字段	值	攻击者得不到更多	同 S09

每条性质在各攻击者条件下的答案

这些攻击者是分析者设定的，在写出查询之前就已确定；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该性质给出的答案；“无答案”指该性质在该攻击者集合下，证明器未对某条主张给出判定，或未记录任何内容；空单元格指分析者未针对该性质设定该攻击者。

攻击者	持有什么	依据	出处
`network`	仅网络：读取、丢弃、修改并发送消息，不持有任何一方的秘密		
`host_key_after`	服务器的长期主机私钥，在会话的 NEWKEYS 交换完成之后获得	rfc4251:1021-1031 把前向保密定义为：会话之后长期私钥泄露而不危及更早的会话，并指出 SSH 密钥交换具有该性质	rfc4251:1021
`eph_x25519_known`	某一会话某一方的 X25519 临时私钥，在该会话期间或之后	rfc4251:1037 指出，若客户端和服务器双方的 DH 私有参数被泄露，则会话密钥被泄露；单组件情形正是混合构造旨在承受的（main s0034-s0035）	rfc4251:1037

攻击者	持有什么	依据	出处
`mlkem_sk_known`	某一会话中属于客户端 C_PK2 的 ML-KEM-768 私钥	main s0031：K_PQ 秘密用客户端的后量子 KEM 私钥解封装；该密钥泄露是 rfc4251:1037 的后量子对应情形	s0031
`kcl_known`	某一会话的经典组件共享秘密 K_CL，例如 X25519 共享被破解或被强制为已知值	main s0034-s0035 组合 K_PQ 与 K_CL；s0047 指出该组合只有在假设下才可证明，而混合构造旨在让一个组件失守时 K 仍保密	main:425
`kpq_known`	某一会话的后量子组件共享秘密 K_PQ	main s0034-s0035、s0047、s0048：经典组件旨在后量子组件失守时保护 K	main:425

ID	network	host_key_after	eph_x25519_known	mlkem_sk_known	kcl_known	kpq_known
K01	成立					
A02	成立					
A03	成立					
A04	成立					
A05	成立					
A08	成立					
R05	成立					
C01		成立				
C04			成立	成立		
C08					成立	成立
D01	成立					
A07	成立					
K04	成立					
K01.14	成立					
A03.15	成立					
A02.16	成立					
X01	成立					
X02	不成立					
X03	成立					

原文和目标

来源：131 个原文单元，30345 字节，冻结于 source-delivery-01.json。 目标：13 个已批准（K01、A02、A03、A04、A05、A08、R05、C01、C04、C08、D01、A07、K04）；提出的问题：3 个。 目标列表在事前按草稿原样批准，使用 --approve-targets。

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

- approve_package: 批准源文包 protocol-157741929283d26b083cb3d1ba49

本步骤的自我报告：1 次请求（1 次失败），0.000 USD，输入 25956 个 token，输出 5174 个，耗时 33 秒。

关于证明器和模型的说明

分析者为目标指定的内容：

步骤	目标	特性	理由
36	K01, A02, A03, A04, A05, A08, R05, C01, C04, C08, D01, A07, K04, K01.14, A03.15, A02.16, X01, X02, X03	plain_primitives, many_sessions	每个答案都只依赖朴素原语：HASH 和密钥派生作为不透明单向函数，ML-KEM 作为新鲜秘密的封装，主机签名，以及 X25519 作为满足单一定律 g^xy=g^yx 的 Diffie-Hellman 密钥协商。不涉及异或、其他代数定律，也不涉及任何在会话内被安装并替换的值（传输密钥交换中没有计数器，也没有序列号；重新交换不在范围内），因此这些答案不依赖 changing_state。查询覆盖一次运行中任意数量的客户端和服务器会话，因此这些性质针对无界会话。

分析者的结语

模型：ProVerif 中采用混合方法 mlkem768x25519-sha256（RFC 10042 / RFC 4253）的 SSH 传输密钥交换，正常流程，主机密钥认证。全部 19 个目标（16 个起草，3 个来自文稿补充）在各自指定的攻击者下都有答案，全部 48 个已记录的答案在最终文本上均成立。

已确立（级别为 proved，并配有在移除保护后失败的对照）： - 针对网络攻击者的保密性：K = HASH(K_PQ || K_CL)、两个组件的秘密、两个方向的会话密钥（K01、K01.14、A05），以及交换哈希 H（X03）。 - 主机认证：被接受的 K_S 和签名意味着存在一次服务器运行，具有相同的交换哈希和相同的会话值（A02、A02.16、A03、A03.15、A07、D01、A08），且每次接受都对应一次不同的服务器运行（A04，单射）。 - 按方向、按会话的密钥分离（K04）和新鲜的 ML-KEM 封装（X01）。 - 在失守情形下：会话之后泄露的主机私钥不会给出 K（C01，前向保密）；只泄露一个临时私钥（C04，在 eph_x25519_known 和 mlkem_sk_known 下）或只知道一个组件共享秘密（C08，在 kcl_known 和 kpq_known 下）时，K 仍然保密；一个会话的全部临时状态确实会泄露它（对照 ctl_full）。

攻击：X02，不将所提供的主机密钥与已存储密钥进行比较的客户端，会接受一个持有自己主机密钥对的攻击者（S_PK1 = g，它自己的封装，它自己的签名）。这正是文稿所述及的弱点（rfc4253 s0112；rfc4251:984），因此记录为已知结果，而非发现。无发现：没有任何失败是文稿尚未述及的。R05 仅就协商参数的绑定给出答案；其强度策略部分无法表达，因为模型固定了唯一方法，且未定义最小值。

结果的局限：C_INIT 和 S_REPLY 的长度检查以类型化解析建模；ML-KEM 是 KEM 抽象，不含 FIPS 203 隐式拒绝；X25519 只承载 g^xy = g^yx；哈希、KDF 和签名是理想的；协商、已知主机分发、记录层和重新交换均未建模。答案只依赖朴素原语，没有目标依赖会变化的值，因此无需有界模型或状态审查。具备完整 ML-KEM 和 X25519 模型的证明器（Tamarin+ Agent）可以判定此处所假设的计算步骤。