SSH 复合宿主认证的安全性分析

本报告的其他语言版本：English。

draft-ietf-sshm-composite-sigs-00 · ProVerif+ Agent 报告

文稿	Post-Quantum Composite Signatures in SSH （协议定义）; Post-Quantum/Traditional Hybrid Key Exchange with the Module-Lattice-Based Key-Encapsulation Mechanism for Use in SSH （辅助）; The Secure Shell (SSH) Transport Layer Protocol （辅助）
检查的性质	13 条：13 条成立，0 条不成立
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

本草案中的 SSH 复合宿主认证让客户端用一把同时组合了 ML-DSA-44 与 Ed25519 的宿主密钥来核实服务器的身份，而连接密钥则同时用 X25519 和 ML-KEM-768 协商。客户端 C 在一条可信的 known-hosts 条目中持有服务器的复合宿主密钥 K_S；服务器 S 持有与之匹配的复合私钥。该交换先跑一轮 KEXINIT，然后是混合密钥交换的 init 与 reply，其中 S 用两个分量对交换哈希签名，C 对两者都进行验证。

交换结束时，客户端确信自己与之通信的是被固定下来的复合宿主密钥的持有者，确信自己所验证的交换哈希覆盖了两个算法名列表、两个标识字符串、宿主密钥、两个密钥交换八位串以及共享秘密，并且确信会话密钥来自一个网络上任何人都不知道的秘密。服务器只相信它自己计算了哈希并对其签名；在所建模的运行中，后续没有任何环节使用新密钥。

各方：

- **C**：客户端，验证复合宿主签名。持有：S 的 known-hosts 条目：复合宿主密钥 K_S（mldsa44_pk, ed25519_pk）。
- **S**：服务器，用两个分量对交换哈希签名。持有：复合宿主私钥（mldsa_seed, ed25519_seed）。

图 1 展示了文稿所规定的交换流程。

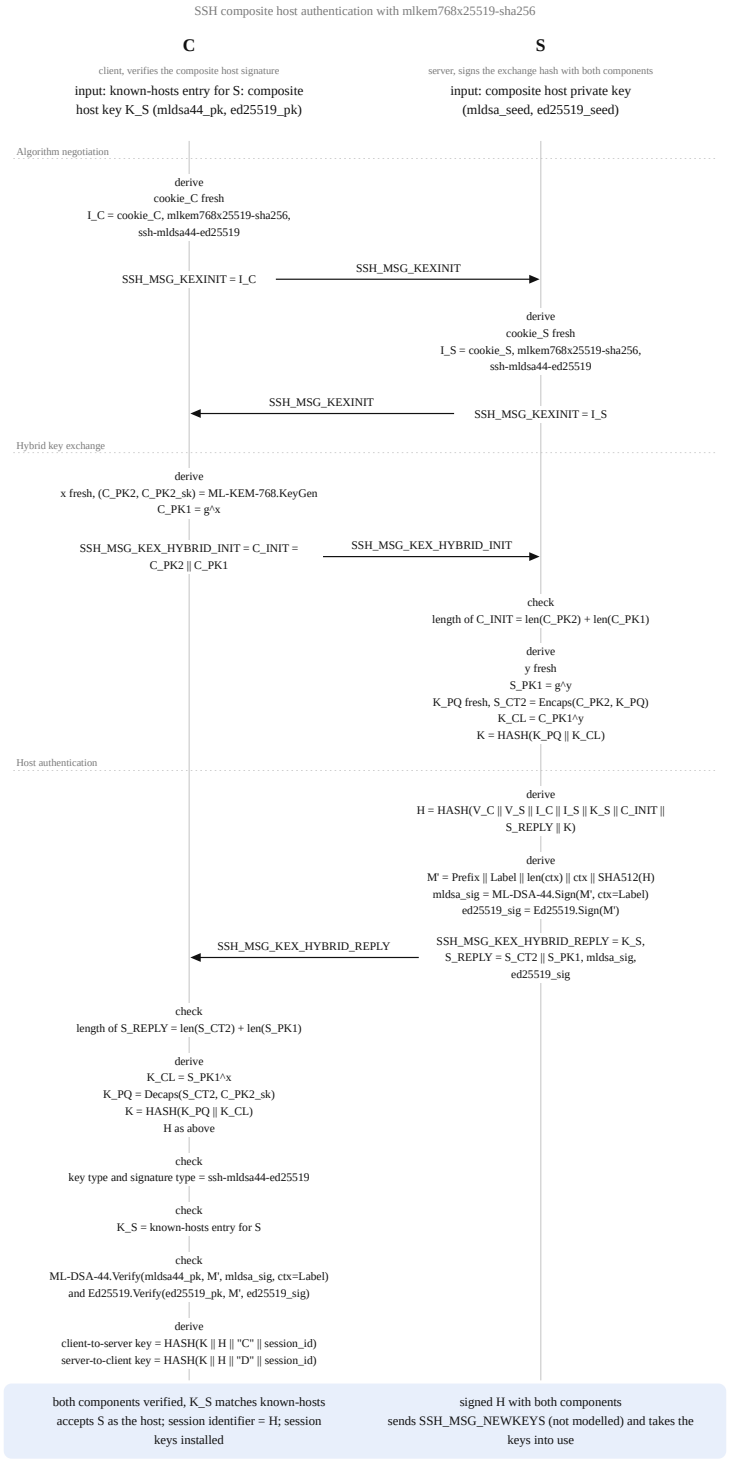


图 1. 文稿所规定的协议。

本次分析的范围见附录 D：分析了什么、省略了什么，以及它依据的是文稿的哪些章节。

2 发现

没有发现。 本次分析没有发现任何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。

无发现：在所建模的流程中，草案的复合宿主认证给出了它本应给出的东西，也没有出现任何文稿尚未说明的情况。客户端获得了关于服务器身份、关于交换哈希及其所有字段（包括两个算法名列表）、关于宿主密钥与本会话的绑定以及关于会话密钥的保证；每一次接受都对应服务器的一次不同运行。宿主私钥、会

话密钥和交换哈希的保密性成立，并且在连接结束后从存储中读取宿主私钥的情况下依然成立。攻破两个分量算法之一并不能获得宿主认证，这正是草案所承诺的双算法保障。这些是关于符号模型的证明：被攻破的分量被建模为一种伪造能力，X25519 被建模为一条 Diffie-Hellman 定律，而 rekey、用户认证以及混合消息的长度检查则被省略。十三个目标中有十二个得到了完整回答，只有一个仅得到部分回答，即密钥分离中密钥复用那一半，它是一条密钥生成的规则，而不是某次运行的性质。

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给出的答案；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
A03	**交换哈希与共享密钥的一致。** 每当客户端接受服务器主机认证（验证组合签名通过并接受会话密钥），就存在服务器的一次真实运行，双方在客户端/服务器身份、主机公钥 K_S、exchange hash 各字段（含协商的算法名列表）与共享秘密 K 上一致。	无	成立	中	是
A02	**服务器对客户端的认证。** 每当客户端接受主机认证，就存在一个以该服务器身份运行的协议实例。	无	成立	中	是
A08	**协商算法绑定于签名。** 客户端接受的主机签名所覆盖的 exchange hash 中，算法名列表、客户端/服务器标识、主机公钥与本次协商结果一致，且签名验证使用协商选定的主机密钥算法。	无	成立	中	是
A04	**每个接受的会话对应不同的服务器运行。** 客户端每次接受主机认证对应服务器一次不同的运行（单射），旧会话的签名交换不能被重放为新会话。	无	成立	中	是
K01	**宿主密钥与会话密钥的保密性。** 组合主机私钥（mldsa_seed、ed25519_seed 或 d）与由 K、H 派生的会话密钥，对未持有它们的攻击者不可推导。	无	成立	中	是
C08	**合成伪造需两个分量。** 在 ML-DSA 分量被攻破（攻击者持有 mldsa_sk）或经典分量被攻破（攻击者持有 ed25519_sk/ecdsa_sk）之一的情形下，攻击者仍不能伪造通过验证的组合主机签名，即客户端不会接受一个服务器未签发的 exchange hash。	服务器组合主机签名密钥的单个分量 (签名验证之前，攻击者已持有该分量私钥)	成立	中	是
K04	**密钥按用途与方向区分。** 不同组合主机密钥之间、组合与非组合密钥之间不复用分量私钥；客户端到服务器与服务器到客户端方向的加密/完整性密钥不同。	无	成立	低	是
D01	**无中间人宿主替换。** 客户端接受的 exchange hash 与主机公钥确实由所声称服务器签发，未被中间人替换。	无	成立	中	是
U01	**单分量被破不破坏宿主认证。** 在组合主机签名的一个分量私钥泄露时，客户端接受的主机认证仍对应服务器真实运行，攻击者不能伪造组合签名。	服务器组合主机签名密钥的单个分量 (签名验证之前)	成立	中	是
A08.10	**宿主密钥绑定于会话。** 客户端接受的主机签名覆盖本次会话的 exchange hash（含 V_C、V_S、I_C、I_S、K_S、C_INIT、S_REPLY、K），主机公钥与本次会话绑定。	无	成立	中	是
U02	**两个签名分量均须验证。** 客户端接受主机认证时，组合签名的两个分量都验证成功，且协商算法与签名格式一致，攻击者不能通过替换绕过任一分量。	无	成立	中	是

ID	性质	假定的泄露	结果	价值	文稿是否已写
X01	**交换哈希的保密性。** 仅能访问网络的攻击者无法推导出客户端已接受宿主的某次连接的交换哈希 H，即 H 对他而言始终未知。	无	成立	低	是
X02	**每个会话的会话标识不同。** 两个被接受的会话从不共享同一个交换哈希 H（会话标识），因此绑定到某个会话的消息无法被重放进另一个会话。	无	成立	低	是

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: 一分量签名密钥泄露时主机认证	否：在 ML-DSA 分量被攻破、或经典分量被攻破的情况下，客户端都不会接受任何并非由某次服务器运行产生的宿主签名，因为两个分量都必须验证通过，所以仅丢失一个分量并不会让攻击者获得宿主认证。	C08, U01, A02
Q2: host key 与会话标识/exchange hash 绑定	是：宿主密钥与标识字符串、两个 KEXINIT 载荷、两个密钥交换值以及共享秘密一起进入交换哈希，因此被接受的密钥绑定于本会话及其会话标识，无法取自另一个会话。	A08.10, A03, X02, A04
Q3: 算法协商/编码替换不会绕过另一签名分量	否：被改写的名列表、密钥类型或签名类型名会改变交换哈希，从而使回复被拒绝；而只携带一个有效分量的回复也会被拒绝，因为两个验证都必须成功。	A08, U02

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和检查给出了什么、未能给出什么，其中密码学被视为理想原语，而不涉及任何实现。

分析者指出，本次分析不覆盖以下内容：

- 未证明 ML-DSA-44、Ed25519 和 ML-KEM 的计算安全性；模型将它们视为理想原语，并把被攻破的分量表示为针对该分量的显式伪造能力。
- X25519 用单一法则 g^xy = g^yx 建模，ML-KEM 建模为自由封装函数；曲线的其他代数性质以及编码均未表示。
- RFC 10042 针对长度扩展尝试所要求的、对 C_INIT 和 S_REPLY 的长度检查未被建模，因为符号配对不存在歧义的拼接。
- 未建模强不可伪造性：签名是确定性项，因此不会为同一个交换哈希产生第二个有效签名，文稿记载的 SUF-CMA 失效也无法复现。
- 未建模任何 rekey、用户认证和加密数据通道，因此对 re-exchange、用户认证方法以及密钥如何使用均未作任何说明。
- 假定宿主密钥在连接前已固定于 known-hosts；接受未经验证的宿主密钥不在分析范围内。

宿主记录的内容：

- A03、A04、K01、X01、X02 的答案依赖于模型所简化的一项操作（附录 D）。

每条性质仍有待解决的问题，以及什么能解决它，见附录 A。

附录 A：逐条性质的解读

以下分析者本人对证明器答案的解读，而非证明器的答案：这些答案对协议意味着什么、解读依据是什么、能否按原样使用，以及它有多大价值。

A03: 交换哈希与共享密钥的一致

每当客户端接受服务器主机认证（验证组合签名通过并接受会话密钥），就存在服务器的一次真实运行，双方在客户端/服务器身份、主机公钥 K_S、exchange hash 各字段（含协商的算法名列表）与共享秘密 K 上一致。

每当客户端接受主机认证，服务器确实以相同的宿主密钥、相同的交换哈希和相同的共享秘密运行了该交换。哈希覆盖了两个算法名列表、两个标识字符串、宿主密钥、两个密钥交换八位串以及共享秘密，因此攻击者只要改动其中任何一项，客户端就会拒绝该回复。

- 结果：成立；是否回答了所问：是；可用但有注意事项
- 价值 中：RFC 4253 和 RFC 10042 指出，exchange hash 用于认证密钥交换并列出其字段；结果确认混合变体保留了这一点。
- 文稿中已说明：是。文稿中的位置：rfc4253:1428, s0217, s0144
- 依据：X25519 由一个 Diffie-Hellman 法则表示，ML-KEM 由一个自由封装函数表示，哈希由一个理想函数表示。；exchange hash 将 KEXINIT 载荷作为不透明字符串绑定；模型不解析其中的名称列表。；假定已知宿主固定，且未建模用户认证和密钥重交换。
- 下一步：已建模流程没有未决事项；要讨论名称列表中选定了哪个算法，需要一个解析名称列表的模型。

A02: 服务器对客户端的认证

每当客户端接受主机认证，就存在一个以该服务器身份运行的协议实例。

接受主机认证的客户端是在与为该服务器固定的组合主机密钥持有者通信，而不是与网络上的攻击者通信。只有同时持有两个分量私钥的一方才能对客户端的新鲜 exchange hash 签名。

- 结果：成立；是否回答了所问：是；可用
- 价值 中：RFC 4253 指出密钥交换使用显式服务器认证，且客户端对照本地数据库验证主机密钥；RFC 4251 指出传输层执行服务器主机认证。
- 文稿中已说明：是。文稿中的位置：rfc4253:1428, rfc4251:652, s0154, s0209
- 依据：两个签名分量均为理想方案，且主机密钥在连接前已固定。；模型为每个服务器只有一个组合密钥；未表示多个服务器或一个名称对应多个密钥。
- 下一步：没有未决事项。

A08: 协商算法绑定于签名

客户端接受的主机签名所覆盖的 exchange hash 中，算法名列表、客户端/服务器标识、主机公钥与本次协商结果一致，且签名验证使用协商选定的主机密钥算法。

客户端接受的签名覆盖双方交换的 KEXINIT 载荷，因此攻击者若改写服务器的名称列表，或将交换转移到另一个主机密钥算法，就会使两个 exchange hash 不同，应答被拒绝。客户端还会拒绝密钥类型或签名类型名称不是协商选定组合算法的应答。

- 结果：成立；是否回答了所问：是；可用但有注意事项
- 价值 中：RFC 10042 将 I_C 和 I_S 列为混合 exchange hash 的字段，RFC 4253 指出 exchange hash 覆盖两个 KEXINIT 载荷；结果确认该绑定对组合主机签名成立。
- 文稿中已说明：是。文稿中的位置：rfc4253:1428, s0144, s0169
- 依据：两个 KEXINIT 载荷被建模为进入哈希的不透明字符串；协商选择未从名称列表计算。；未建模猜测的初始包（first_kex_packet_follows）。
- 未决：协商选择本身（而非整个 KEXINIT 载荷）是否与密钥类型比较，属于实现问题；模型检查类型名称但不检查选择算法。
- 下一步：显式建模名称列表选择，以讨论两个组合算法之间的降级。

A04: 每个接受的会话对应不同的服务器运行

客户端每次接受主机认证对应服务器一次不同的运行（单射），旧会话的签名交换不能被重放为新会话。

每次接受主机认证都匹配服务器的一次不同运行，因为客户端为密钥封装和 X25519 共享贡献了新鲜值，且两者都在被签名的哈希内。因此，在旧会话中记录的应答不会在新会话中被接受。

- 结果：成立；是否回答了所问：是；可用但有注意事项
- 价值 中：RFC 4253 使第一个 exchange hash 成为唯一会话标识符，RFC 4251 指出公钥认证交换不能在其他会话中重放。
- 文稿中已说明：是。文稿中的位置：rfc4251:849, s0187
- 依据：按文稿要求，假定客户端临时 X25519 秘密和 ML-KEM 密钥的新鲜性。；排除密钥重交换，因此不检查重交换后的会话标识符。
- 下一步：没有未决事项。

K01: 宿主密钥与会话密钥的保密性

组合主机私钥（mldsa_seed、ed25519_seed 或 d）与由 K、H 派生的会话密钥，对未持有它们的攻击者不可推导。

组合主机私钥和会话密钥对网络上的攻击者保持未知。即使在连接后从服务器存储中读取整个组合主机私钥，会话密钥仍保持未知，因为它们派生自临时密钥交换而非主机密钥；只有泄露自身会话状态的服务器才会暴露它们。

- 结果：成立；是否回答了所问：是；可用但有注意事项
- 价值 中：RFC 4253 从 K 和 H 派生每个加密和完整性密钥，RFC 4251 要求会话密钥的前向保密，并警告收集流量以供日后解密。
- 文稿中已说明：是。文稿中的位置：rfc4251:1021, s0189, s0197, s0138
- 依据：临时 X25519 秘密和 ML-KEM 解封装结果的保密性是共享秘密的唯一保护。；会话密钥被建模为每个方向一次哈希派生；未建模长密钥的扩展过程。；主机密钥泄露置于已建模会话结束之后，这是前向保密的解读。
- 未决：未询问共享秘密在会话中途泄露一个临时值时的保密性。
- 下一步：Tamarin 及其等式理论可以检查临时密钥泄露和 ML-KEM 隐式拒绝。

C08: 合成伪造需两个分量

在 ML-DSA 分量被攻破（攻击者持有 mldsa_sk）或经典分量被攻破（攻击者持有 ed25519_sk/ecdsa_sk）之一的情形下，攻击者仍不能伪造通过验证的组合主机签名，即客户端不会接受一个服务器未签发的 exchange hash。

攻破两个分量签名算法之一不会让攻击者冒充服务器：客户端要求两个验证都通过，因此当 ML-DSA 可伪造时，被接受的应答仍需要攻击者无法生成的 Ed25519 签名，对称情形走同一代码路径。这是草案的双算法承诺，且对所建模的交换成立。

- 结果：成立；是否回答了所问：是；可用但有注意事项
- 价值 中：草案指出攻击者必须同时伪造 ML-DSA 和椭圆曲线签名，且只要至少一个分量未被攻破，组合就保持 EUF-CMA；结果对 SSH 交换机械地复现了这一点。
- 文稿中已说明：是。文稿中的位置：main:447, main:455, s0096, s0098, s0083
- 依据：被攻破的分量被建模为该分量的通用伪造能力，这比持有其私钥更强；经典情形是同一查询的对称解读。；ML-DSA 或 Ed25519 是 EUF-CMA 的计算陈述未被证明；模型只显示当两半都失败时组合失败。；未表示攻破的攻击者类别（经典或量子）。
- 未决：底层规范描述的按攻击者类别的退化是计算性的，超出模型范围。
- 下一步：ProVerif 中没有进一步事项；该主张需要归约证明，而非符号证明。

K04: 密钥按用途与方向区分

不同组合主机密钥之间、组合与非组合密钥之间不复用分量私钥；客户端到服务器与服务器到客户端方向的加密/完整性密钥不同。

连接的两个方向始终使用不同的派生密钥，因此知道一个方向不会给出另一个方向。目标的另一半，即分量密钥材料绝不在其组合密钥之外复用，是关于密钥如何生成和部署的规则，协议模型无法显示；模型仅强制接受固定的组合密钥类型。

- 结果：成立；是否回答了所问：部分；可用但有注意事项
- 价值 低：草案禁止复用分量密钥材料，RFC 4253 从不同字母派生两个方向；结果仅确认派生方面。
- 文稿中已说明：是。文稿中的位置：main:451, s0097, s0190, s0191
- 依据：密钥生成和部署在模型之外；复用禁止作为假设。；密钥分离仅对两个建模的方向密钥显示，未对完整性密钥或多个密钥长度显示。
- 未决：组合密钥与独立密钥之间的密钥材料复用无法向协议证明器询问。
- 下一步：实现或策略审查可以解决复用这一半。

D01: 无中间人宿主替换

客户端接受的 exchange hash 与主机公钥确实由所声称服务器签发，未被中间人替换。

在 known-hosts 中固定了服务器组合主机密钥的客户端，永远不会接受网络上替换的主机密钥或 exchange hash。对于没有分量私钥的攻击者，无法生成对客户端新鲜哈希的签名。

- 结果：成立；所问是否得到回答：是；可用
- 价值中：RFC 4251 描述了这种中间人情形，并指出当宿主密钥事先经过验证时，攻击者无法得逞，因为他无法对会话特定的数据签名。
- 文稿中已说明：是。文稿中的位置：rfc4251:905, s0209, s0219
- 依据：known-hosts 条目在连接建立前已被信任且存在；接受新的宿主密钥不在模型范围内。；签名和哈希是理想的。

- 下一步：无未决事项。

U01: 单分量被破不破坏宿主认证

在组合主机签名的一个分量私钥泄露时，客户端接受的主机认证仍对应服务器真实运行，攻击者不能伪造组合签名。

一个分量私钥泄露并不能让攻击者获得宿主认证：组合校验还需要另一个分量的签名。ML-DSA 泄露和经典分量泄露这两种解读由同一个查询回答，因为模型独立检查两个分量并要求两者都通过。

- 结果：成立；所问是否得到回答：是；可用，但有注意事项
- 价值中：草案指出，只要一个分量未被攻破，组合签名就仍然不可伪造；本条是这一陈述在宿主认证上的形式。
- 文稿中已说明：是。文稿中的位置：main:447, s0096, s0101
- 依据：泄露被建模为能够伪造该分量，而不是持有密钥；这是更强的能力，因此答案覆盖了密钥泄露这一解读。；未建模分量密钥的跨协议重用，而草案禁止这种做法。
- 未决：经典分量这一解读由模型两项检查的对称性回答，而不是由第二个查询回答。
- 下一步：若模型包含两个不同的组合算法，就能用两个独立查询分别回答这两种解读。

A08.10: 宿主密钥绑定于会话

客户端接受的主机签名覆盖本次会话的 exchange hash（含 V_C、V_S、I_C、I_S、K_S、C_INIT、S_REPLY、K），主机公钥与本次会话绑定。

客户端接受的组合宿主密钥位于本次会话的 exchange hash 之内，与客户端的新鲜密钥交换值一起参与哈希，因此在另一个会话中记录的宿主密钥或主机签名不能在这里使用。

- 结果：成立；所问是否得到回答：是；可用
- 价值中：RFC 10042 把 K_S 放入混合 exchange hash，RFC 4253 使第一个 exchange hash 成为会话标识符。
- 文稿中已说明：是。文稿中的位置：rfc4253:1428, s0144, s0187
- 依据：按线格式的要求，宿主密钥被建模为应答中携带并位于哈希内的一对公钥。；exchange hash 被视为首次密钥交换的会话标识符；不包含 rekey。
- 下一步：无未决事项。

U02: 两个签名分量均须验证

客户端接受主机认证时，组合签名的两个分量都验证成功，且协商算法与签名格式一致，攻击者不能通过替换绕过任一分量。

只有当 ML-DSA-44 和 Ed25519 的验证在其协商的组合密钥类型下都成功时，客户端才接受组合主机签名。只携带一个有效分量的应答，或指定其他密钥类型或签名类型的应答，都会被拒绝；对照查询恰好显示了这一点：只检查 ML-DSA 的客户端会接受用被攻破分量伪造的应答。

- 结果：成立；所问是否得到回答：是；可用
- 价值中：草案要求两个分量验证都成功，并要求任一分量失败即组合失败；结果确认，按此实现的客户端不会被替换绕过。
- 文稿中已说明：是。文稿中的位置：main:440, s0083, s0088, s0077
- 依据：两个分量方案是理想的，带前缀和标签的组合器由围绕 exchange hash 的一个函数表示。；线格式由密钥和签名 blob 中的算法类型名称表示，客户端将其与协商的名称比较。

- 下一步：无未决事项。

X01: 交换哈希的保密性

只掌握网络的攻击者无法推导出客户端已接受宿主的连接的 exchange hash H，即 H 对他保持未知。

已接受连接的 exchange hash 对网络上的攻击者保持未知，因为它包含共享秘密和双方的新鲜密钥交换值。两份文稿在这里存在分歧：RFC 4251 允许会话标识符公开，而 RFC 4253 要求 exchange hash 保密，对真实连接成立的是保密这一解读。

- 结果：成立；所问是否得到回答：是；可用，但有注意事项
- 价值低：RFC 4253 说 exchange hash 应保密，RFC 4251 说会话标识符可以公开，而这里只问 RFC 4253 的解读。
- 文稿中已说明：是。文稿中的位置：rfc4253:218, rfc4251:849, s0218
- 依据：哈希是理想的，因此哈希隐藏其输入；对于结构泄露的哈希，该主张会不成立。；哈希即会话标识符，上层以模型未表示的方式使用它。
- 未决：文稿本身未说明意图是这两种解读中的哪一种。
- 下一步：对 RFC 4253 和 RFC 4251 做编辑性澄清，即可确定会话标识符是否应当保密。

X02: 每个会话的会话标识不同

两个已接受的会话从不共享同一个 exchange hash H（会话标识符），因此绑定到某个会话的消息不能重放到另一个会话。

每个已接受的会话都对应服务器的一次不同运行和客户端自己的新鲜密钥交换值，因此一个会话的 exchange hash 不会被另一个已接受的会话重用。模型以单射一致性检查这一点，因为其查询语言无法直接陈述两个事件携带不同的值。

- 结果：成立；所问是否得到回答：部分；可用，但有注意事项
- 价值低：RFC 4253 使 exchange hash 成为唯一标识符，RFC 4251 说两个会话共享会话标识符将允许它们之间的重放。
- 文稿中已说明：是。文稿中的位置：rfc4251:857, s0187
- 依据：唯一性依赖于客户端临时值的新鲜性以及哈希理想且单射；未建模哈希碰撞。；该陈述以宿主密钥、exchange hash 和共享秘密上的单射一致性检查，而不是直接比较两个会话标识符。
- 未决：所用查询语言无法表达直接的无碰撞陈述。
- 下一步：支持显式相等查询或计数查询的证明器（例如带限制的 Tamarin）可以直接陈述碰撞性质。

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，以及每项攻击和分析者的判定。

目标	级别	界	查询	对照	攻击	主张
A03	proved		agree	checked		1 proved
A02	proved		agree	checked		1 proved
A08	proved		agree	checked		1 proved
A04	proved		inj	checked		1 proved
K01	proved		secret_sym	checked		3 proved
C08	proved		agree	checked		2 proved
K04	proved		dirsep	checked		1 proved
D01	proved		agree	checked		1 proved
U01	proved		agree	checked		2 proved
A08.10	proved		agree	checked		1 proved
U02	proved		agree	checked		1 proved
X01	proved		secret_h	checked		1 proved
X02	proved		inj	checked		1 proved

逐条主张，按分析者对每条主张的措辞；每条主张的答案来自证明器：

目标	主张	攻击者	回答	查询	对照
A03	每当客户端接受宿主认证时，它所接受的组合宿主密钥 K_S、交换哈希 H（绑定 V_C、V_S、I_C、I_S、K_S、C_INIT、S_REPLY 和 K）以及共享秘密 K，都来自服务器的一次运行；攻击者除网络之外一无所有。	`network`	proved	agree	checked
A02	每当客户端接受宿主认证时，都存在服务器的一次运行，持有为其固定的组合宿主密钥，即客户端正在与指定的服务器通信，而不是与网络攻击者通信。	`network`	proved	agree	checked
A08	被接受的宿主签名下的交换哈希覆盖了实际交换的名称列表 I_C 和 I_S、宿主密钥以及客户端和服务器的密钥交换值，并且客户端只接受其协商得到的组合算法名称；被改写的名称列表或密钥类型会产生不同的 H 并被拒绝。攻击者除网络之外一无所有。	`network`	proved	agree	checked
A04	每次接受宿主认证都对应服务器的不同运行，因此录制的密钥交换回复不能被重放到新会话中。	`network`	proved	inj	checked
K01	由 K 和 H 派生的会话密钥（客户端到服务器和服务器到客户端的密钥）对仅有网络的攻击者保持未知；攻击者只有在服务器泄露它们的会话中才能获知它们。	`network`	proved	secret_sym	checked
K01	即使攻击者在连接之后从服务器存储中读取整个组合宿主私钥，会话密钥仍保持秘密：会话密钥不依赖宿主密钥（前向保密）。	`host_key_after_session`	proved	fs_sym	checked

目标	主张	攻击者	回答	查询	对照
K01	组合宿主私钥，包括 ML-DSA 分量（mldsa_seed）和 Ed25519 分量（ed25519_seed），在密钥交换期间对仅有网络的攻击者保持未知。	`network`	proved	secret_priv	checked
C08	ML-DSA-44 分量被攻破（攻击者可以为任意交换哈希生成有效的 ML-DSA 分量签名）：客户端仍然不接受任何未经服务器运行签名的交换哈希上的组合宿主签名，因此宿主认证在该分量丢失后依然成立。	`broken_component`	proved	agree	checked
C08	经典分量同理：模型用同一代码路径检查两个分量并要求两者都通过，因此只能伪造 Ed25519 分量签名的攻击者处于与上述对称的位置，同样无法冒充服务器通过认证。	`broken_component`	proved	agree	checked
K04	同一会话的客户端到服务器密钥和服务器到客户端密钥始终是不同的值，因此知道一个方向并不能得到另一个方向，两个方向由其派生输入分隔。	`network`	proved	dirsep	checked
D01	从 known-hosts 持有服务器组合宿主密钥的客户端，绝不会接受网络攻击者替换的宿主密钥或交换哈希。	`network`	proved	agree	checked
U01	组合宿主签名的一个分量被攻破时，攻击者仍然无法让客户端在没有该服务器运行的情况下接受对该服务器的宿主认证（ML-DSA 被攻破；经典情形是对同一查询的对称解读）。	`broken_component`	proved	agree	checked
U01	经典分量被攻破而 ML-DSA 分量仍然成立：同一查询覆盖此情形，因为两个分量检查相互独立且都必需，所以攻击者无法冒充服务器通过认证（对称解读）。	`broken_component`	proved	agree	checked
A08.10	客户端接受的组合宿主密钥绑定到本会话的交换哈希和会话标识符，因此取自另一会话的宿主密钥不能用于本会话。	`network`	proved	agree	checked
U02	只有当 ML-DSA-44 和 Ed25519 验证在其协商的组合密钥类型下都成功时，客户端才接受组合宿主签名，因此只携带一个有效分量、或指明其他密钥或签名类型的回复会被拒绝。	`network`	proved	agree	checked
X01	客户端接受的连接的交换哈希 H（同时也是会话标识符）无法被仅有网络的攻击者推导出来；它绑定到共享秘密 K。	`network`	proved	secret_h	checked
X02	没有两个被接受的会话共享会话标识符：每次接受都对应不同的服务器运行和客户端自己的新鲜密钥交换值，因此绑定到一个会话的消息在另一个会话中无效。	`network`	proved	inj	checked

附录 D：分析是怎么跑的

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

ProVerif+ Agent 会话 session-20261114T224652，2026-11-14。状态：已完成。模型：deepseek-v4-1-flash-261017。步骤：38，证明器 10 秒，实际耗时 12 分钟，成本 0.099 美元。

ProVerif：运行 50 个证明，50 个已判定，0 个无答案，6 秒，记录 18 个回答。

分析了什么

协议	SSH composite host authentication
版本	draft-ietf-sshm-composite-sigs-00
分析范围	固定草案中 Ed25519+ML-DSA 组合主机签名，以及 MLKEM768X25519 KEX；两分量都验证，完整 exchange hash、角色和算法协商绑定；可信 known-hosts 预置，排除用户认证及 rekey；算法标识按实际草案处理，不发明值。
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	228 个编号原文单元（37431 字节），来自下列章节。本报告中诸如 s0001 的引用即为这些原文单元之一

分析所依据的文稿章节：

文稿	章节	标题
main	4	Algorithm specifications
main	4.1	Key generation
main	4.1.1	ssh-mldsa44-ed25519
main	4.1.1.1	Composite raw public key
main	4.1.1.2	Composite raw private key
main	4.1.2	ssh-mldsa87-p384
main	4.1.2.1	Composite raw public key
main	4.1.2.2	Composite raw private key
main	4.1.3	SSH Public Key Format
main	4.1.4	SSH Private Key Format
main	4.2	Signatures
main	4.2.1	Signature contexts
main	4.2.2	Signature Generation
main	4.2.2.1	ssh-mldsa44-ed25519
main	4.2.2.2	ssh-mldsa87-p384
main	4.2.2.3	SSH Signature Format
main	4.2.2.4	Signature Verification

文稿	章节	标题
main	4.2.2.4.1	ssh-mldsa44-ed25519
main	4.2.2.4.2	ssh-mldsa87-p384
main	5	Security Considerations
rfc10042	2	PQ/T Hybrid Key Exchange
rfc10042	2.1	PQ/T Hybrid Key Exchange Method Abstraction
rfc10042	2.2	PQ/T Hybrid Key Exchange Message Numbers
rfc10042	2.3	PQ/T Hybrid Key Exchange Method Names
rfc10042	2.3.1	mlkem768nistp256-sha256
rfc10042	2.3.2	mlkem1024nistp384-sha384
rfc10042	2.3.3	mlkem768x25519-sha256
rfc10042	2.4	Shared Secret K
rfc10042	2.5	Key Derivation
rfc4253	7	Key Exchange
rfc4253	7.1	Algorithm Negotiation
rfc4253	7.2	Output from Key Exchange
rfc4253	7.3	Taking Keys Into Use
rfc4253	8	Diffie-Hellman Key Exchange
rfc4253	8.1	diffie-hellman-group1-sha1
rfc4253	8.2	diffie-hellman-group14-sha1
rfc4253	9	Key Re-Exchange

源文包步骤在文稿中定位到的协议目的：固定草案中 Ed25519+ML-DSA 组合主机签名，以及 MLKEM768X25519 KEX；两分量都验证，完整 exchange hash、角色和算法协商绑定；可信 known-hosts 预置，排除用户认证及 rekey；算法标识按实际草案处理，不发明值。

对照文稿定性质

在编写查询之前，分析者将起草的性质列表与文稿自身对协议目的的说明进行了对照。名称、性质所涉内容以及文稿所述内容均为分析者的解读；所提及的位置在文稿中确实存在。

ID	简称	对象	文稿怎么说	出处
A03	交换哈希与共享密钥的一致	客户端 C 确信它接受的宿主密钥 K_S、包含所有字段（V_C、V_S、I_C、I_S、C_INIT、S_REPLY）的交换哈希 H 以及共享秘密 K 均来自服务器 S 的同一次运行	RFC 4253 指出交换哈希用于认证密钥交换，并列出其覆盖的字段；RFC 10042 给出 H 的混合变体	s0217, s0218, s0144, s0205
A02	服务器对客户端的认证	客户端 C 确信名为 S 的服务器的一个协议实例确实运行过，即它正在与为 S 固定的合成宿主密钥的持有者通信	RFC 4253 指出密钥交换使用显式服务器认证，客户端用本地数据库验证宿主密钥；RFC 4251 指出传输层执行服务器宿主认证	rfc4253:1428, s0154, s0209, s0205

ID	简称	对象	文稿怎么说	出处
A08	协商算法绑定于签名	客户端 C 确信其接受的宿主签名所对应的交换哈希覆盖了它交换的算法名称列表 I_C 和 I_S 以及它选择的宿主密钥算法，并且验证使用该选定算法	RFC 10042 将 I_C 和 I_S 列入混合交换哈希的字段；RFC 4253 指出签名算法随 KEXINIT 协商，交换哈希覆盖两个 KEXINIT 载荷	s0143, s0144, s0169, s0157
A04	每个接受的会话对应不同的服务器运行	客户端 C 的每一次宿主认证接受都对应服务器 S 的一次不同运行，因此录制的回复无法被重放到新会话中	RFC 4253 指出首次密钥交换的交换哈希是唯一的会话标识，后续重新交换不会改变它；RFC 4251 指出公钥认证交换绑定于会话，无法在其他会话中重放	rfc4251:849, s0187, s0161
K01	宿主密钥与会话密钥的保密性	合成宿主私钥（mldsa_seed, ed25519_seed）除服务器 S 外无人知晓，由 K 和 H 派生的会话密钥对网络攻击者保密	RFC 4253 从 K 和 H 派生所有加密和完整性密钥；RFC 4251 要求会话密钥具备前向保密，并警告现在收集流量、以后解密的做法必须不可行	rfc4251:1021, s0189, s0197, s0138, s0015
C08	合成伪造需两个分量	客户端 C 确信为服务器 S 接受的合成宿主签名是用两个分量私钥共同生成的；只能伪造一个分量的攻击者无法冒充 S	main 指出攻击者必须同时伪造 ML-DSA 和椭圆曲线签名才能伪造合成签名，且只要至少一个分量未被攻破，合成签名就保持 EUF-CMA	main:447, main:455, s0096, s0098, s0088
K04	密钥按用途与方向区分	服务器 S 和客户端 C 仅在此合成密钥内部使用分量密钥材料，连接的两个方向使用不同的派生密钥	main 禁止在合成密钥与非合成密钥之间、或不同合成密钥之间复用分量密钥材料；RFC 4253 用不同字母派生两个方向的密钥	main:451, s0097, s0190, s0191
D01	无中间人宿主替换	从 known-hosts 持有 S 的合成宿主密钥的客户端 C 确信，网络中没有任何攻击者在交换中替换了宿主密钥或交换哈希	RFC 4253 要求客户端验证宿主密钥确实是 S 的密钥；RFC 4251 指出在关联已验证的情况下中间人无法得逞，因为他无法对会话特定数据签名	rfc4251:905, s0209, s0219
U01	单分量被破不破坏宿主认证	客户端 C 确信泄露或攻破服务器 S 的单个分量私钥不会让攻击者冒充 S	main 陈述双算法目标：合成密钥至少与未攻破的分量一样安全，攻击者必须伪造两个分量	main:447, main:469, s0096, s0101, s0083
A08.10	宿主密钥绑定于会话	客户端 C 确信它接受的宿主密钥 K_S 绑定于本会话的交换哈希和会话标识，而非取自其他会话	RFC 10042 将 K_S 放入混合交换哈希；RFC 4253 指出交换哈希是上层使用的会话标识	s0144, s0187, s0209
U02	两个签名分量均须验证	客户端 C 确信接受的合成宿主签名在协商算法下同时通过了 ML-DSA-44 和 Ed25519 验证，因此替换签名类型或密钥类型、或丢弃某个分量均不会被接受	合成签名规范要求两个分量验证都成功，任一失败则合成签名失败，并给出签名和密钥类型名称的线上格式	s0083, s0088, s0077, s0032
X01	交换哈希的保密性	连接的交换哈希 H（同时也是会话标识）对网络攻击者保密	RFC 4253 指出交换哈希应（SHOULD）保密；RFC 4251 则相反，指出会话标识可以公开而不损害安全，因此两份文稿意见相左，此处追问的是保密性解读。	rfc4253:218, rfc4251:849

ID	简称	对象	文稿怎么说	出处
X02	每个会话的会话标识不同	每个连接的会话标识，以及计算它的客户端和服务器	RFC 4253 指出首次密钥交换的交换哈希是连接的唯一标识；RFC 4251 指出若两个会话具有相同的会话 id，则一个会话的报文可被重放到另一个会话上。	rfc4251:857, main:187

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 133 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 32 段，没选 101 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	main:444-445	没有陈述目标	指向复合签名规范安全考虑部分的指针；其中陈述的目标已录入下文 rfc10073 第 9 节
P002	main:447-449	C08, U01, `broken_component`	
P003	main:451-453	K04	
P004	main:455-458	C08, U01, `broken_component`	
P005 至 P006	main:462-466	没有陈述目标	页眉和页脚
P007	main:469-474	C08, U01, `broken_component`	
P008	main:476-478	证明器问不了	SUF-CMA 是关于为同一消息产生第二个不同签名的计算性陈述；这里所用的符号签名模型具有确定性签名，无法表达可延展性
P009	main:480-482	不在范围内	当组件签名在独立的受信密钥下校验时，不可分离性才会成为问题，而 P003 的密钥复用禁令排除了这种情况；所分析的协议始终在来自 known-hosts 的复合密钥下校验两个组件
P010	rfc10042:392-394	K01, A03	
P011	rfc10042:396-406	A04	
P012	rfc10042:408-423	证明器问不了	编码长度侧信道不在 Dolev-Yao 模型的范围内
P013	rfc10042:425-438	K01	
P014	rfc10042:440-449	没有陈述目标	关于哪条椭圆曲线更易于实现的建议
P015	rfc10073:2083-2084	没有陈述目标	指向 ML-DSA 安全考虑部分的指针
P016 至 P017	rfc10073:2088-2109	C08, U01	
P018	rfc10073:2111-2126	没有陈述目标	迁移灵活性和代码库复用：这是部署层面的论证，没有必须成立的内容
P019	rfc10073:2130-2142	K04	

段落	出处	定为	不选的原因
P020	rfc10073:2144-2148	C08, U01, `broken_component`	
P021	rfc10073:2152-2156	C08	
P022	rfc10073:2158-2164	C08, U01	
P023	rfc10073:2166-2167	C08, U01, `broken_component`	
P024	rfc10073:2169-2170	C08, U01	
P025	rfc10073:2172-2173	C08, U01, `broken_component`	
P026	rfc10073:2175-2180	不在范围内	涉及 X.509 证书内的复合签名，而非此处分析的 SSH 宿主认证
P027 至 P031	rfc10073:2184-2212	证明器问不了	强不可伪造性是关于可延展性的计算性陈述；符号模型把签名视为确定性项，无法为同一消息产生第二个有效签名
P032 至 P035	rfc10073:2216-2248	不在范围内	可分离性是指把某个组件签名从复合签名中取出，并在独立的受信密钥下校验；所分析的协议从不信任独立的组件密钥（禁止密钥复用，P003）
P036 至 P037	rfc10073:2252-2273	K04	
P038	rfc10073:2275-2286	不在范围内	CA 的证书吊销检查
P039	rfc10073:2288-2294	不在范围内	涉及其他协议的应用上下文字符串；SSH 将上下文固定为空字符串
P040	rfc10073:2298-2305	不在范围内	独立传统验证器内部的可选防护；SSH 客户端校验的是复合类型名和标签，而不是裸的传统签名
P041 至 P042	rfc10073:2309-2326	不在范围内	弃用算法的策略，而非某次运行的属性
P043	rfc4251:648-650	没有陈述目标	仅说明安全考虑部分收集在何处
P044	rfc4251:652-655	A02, A03, K01	
P045	rfc4251:657-662	不在范围内	用户认证被任务排除
P046	rfc4251:664-669	不在范围内	连接协议被任务排除
P047	rfc4251:673-684	A04	
P048	rfc4251:690-694	证明器问不了	随机数据的质量是模型的假设，不是可以向证明器提出的问题
P049	rfc4251:698-701	不在范围内	终端控制字符，与密码学无关
P050 至 P068	rfc4251:707-818	不在范围内	加密数据通道中批量密码的选择以及 CBC/IV 处理；任务不建模任何应用数据，模型也没有数据通道的加密
P069 至 P071	rfc4251:822-835	不在范围内	加密数据通道的完整性、none-MAC 以及带重新加密的序列号回绕；重新加密被任务排除

段落	出处	定为	不选的原因
P072 至 P073	rfc4251:839-855	A04	
P074	rfc4251:857-862	X02	
P075	rfc4251:864-887	不在范围内	数据通道的序列号和重新加密
P076	rfc4251:891-896	D01, A02	
P077	rfc4251:902-903	D01	
P078	rfc4251:905-932	A02, D01	
P079	rfc4251:934-949	不在范围内	在没有受信关联的情况下分发宿主公钥；任务把受信的 known-hosts 条目固定为假设
P080	rfc4251:955-962	没有陈述目标	关于发布指纹的建议
P081	rfc4251:964-982	不在范围内	会话建立之后对数据包的操纵，即数据通道
P082	rfc4251:984-989	D01	
P083 至 P085	rfc4251:993-1009	证明器问不了	拒绝服务和 CPU 开销不在基于迹的证明器范围内
P086	rfc4251:1013-1017	证明器问不了	隐蔽信道不在符号迹模型的范围内
P087 至 P088	rfc4251:1021-1042	K01, `host_key_after_session`	
P089	rfc4251:1046-1051	A08	
P090	rfc4251:1063-1071	证明器问不了	流量分析取决于消息长度和时序，而模型对此进行了抽象
P091 至 P093	rfc4251:1075-1090	不在范围内	用户认证和连接协议被任务排除
P094 至 P096	rfc4251:1094-1118	不在范围内	涉及在弱传输上运用户认证
P097	rfc4251:1122-1139	不在范围内	上层的调试消息
P098 至 P103	rfc4251:1143-1182	不在范围内	服务器的本地安全策略（rhosts、代理转发等），而非密钥交换
P104 至 P106	rfc4251:1186-1201	不在范围内	口令认证和基于主机的用户认证被任务排除
P107 至 P108	rfc4251:1219-1232	不在范围内	端点自身的安全性
P109 至 P115	rfc4251:1236-1287	不在范围内	代理转发和 X11 转发，即任务略去的系统部分

段落	出处	定为	不选的原因
P116	rfc4253:1428-1431	A02, A03, K01	
P117	rfc4253:1433-1434	没有陈述目标	指向架构文档的指针
P118	rfc5656:708-712	K01	
P119 至 P128	rfc5656:714-795	不在范围内	椭圆曲线和哈希函数的选择与强度：原语的计算参数，模型将其视为理想，且 NIST 曲线并非所分析方法所用的曲线
P129	rfc5656:797-803	证明器问不了	随机数生成器的质量
P130	rfc8731:165-166	没有陈述目标	仅说明继承了其他安全考虑
P131 至 P132	rfc8731:168-185	证明器问不了	曲线规模的强度估计，以及 mpint 编码侧信道
P133	rfc8731:187-190	没有陈述目标	关于优先选择哪条曲线的建议

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 3 项，没设的 20 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	按定义全部失守	这里每个目标都陈述客户端被保证了什么；客户端被攻击者控制时，就没有客户端可被保证了，它持有的值只有自己的会话密钥和公开的宿主公钥
S02	服务器 S	参与方	`host_key_after_session`	
S03	攻击者	参与方	攻击者得不到更多	模型中的攻击者就是已经控制了网络的一方
S04	主机公钥 K_S	值	攻击者得不到更多	宿主公钥是公开的，并在密钥交换中发送
S05	exchange hash H	值	按定义全部失守	exchange hash 正是被询问其保密性的值；持有它就失去了该目标，且不会泄露任何密钥材料
S06	共享秘密 K	值	按定义全部失守	所有会话密钥都由 K 派生；按定义，持有 K 的攻击者就掌握了该会话的流量
S07	会话标识	值	攻击者得不到更多	会话标识可以公开而不损害安全性（RFC 4251），并且标识字符串、名称列表、角色字符串和 cookie 都以明文发送或为公开常量
S08	服务器身份	值	攻击者得不到更多	会话标识可以公开而不损害安全性（RFC 4251），并且标识字符串、名称列表、角色字符串和 cookie 都以明文发送或为公开常量
S09	协商算法名列表 I_C/I_S	值	攻击者得不到更多	会话标识可以公开而不损害安全性（RFC 4251），并且标识字符串、名称列表、角色字符串和 cookie 都以明文发送或为公开常量
S10	角色标识 V_C/V_S	值	攻击者得不到更多	会话标识可以公开而不损害安全性（RFC 4251），并且标识字符串、名称列表、角色字符串和 cookie 都以明文发送或为公开常量

ID	参与方或值	类别	攻击者条件	不设的原因
S11	会话标识/exchange hash	值	攻击者得不到更多	会话标识可以公开而不损害安全性（RFC 4251），并且标识字符串、名称列表、角色字符串和 cookie 都以明文发送或为公开常量
S12	cookie	值	攻击者得不到更多	会话标识可以公开而不损害安全性（RFC 4251），并且标识字符串、名称列表、角色字符串和 cookie 都以明文发送或为公开常量
S13	组合主机私钥	值	`host_key_after_session`	
S14	会话加密密钥	值	按定义全部失守	按定义，持有某会话的加密密钥或完整性密钥就掌握了该会话的流量；仍然成立的部分在前向保密下询问
S15	完整性密钥	值	按定义全部失守	按定义，持有某会话的加密密钥或完整性密钥就掌握了该会话的流量；仍然成立的部分在前向保密下询问
S16	组合主机签名	值	攻击者得不到更多	组合签名在密钥交换应答中经网络发送
S17	exchange hash	值	按定义全部失守	与 S05 是同一个值：其保密性本身就是目标
S18	分量私钥	值	`host_key_after_session`	
S19	方向密钥	值	按定义全部失守	两个方向密钥都由同一个 K 和 H 派生，因此持有某会话某个派生密钥的攻击者就掌握了该会话；该目标只要求两个方向是不同的值
S20	主机公钥	值	攻击者得不到更多	宿主公钥是公开的
S21	组合签名	值	攻击者得不到更多	组合签名经网络发送
S22	协商算法名	值	攻击者得不到更多	协商出的算法名和签名类型名在 KEXINIT 和应答中以明文发送
S23	签名格式	值	攻击者得不到更多	协商出的算法名和签名类型名在 KEXINIT 和应答中以明文发送

每条性质在各攻击者条件下的答案

这些攻击者由分析者设定，在编写查询之前就已定好；每个答案都记录在其一之下。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”是为该性质设定的攻击者，在该攻击者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、修改并发送消息，不持有任何参与方的秘密		
`broken_component`	能够在不持有分量私钥的情况下，针对任何消息、在服务器的组合宿主公钥下生成有效的 ML-DSA-44 分量签名（一个签名算法被攻破）；对经典分量具备同样的能力属于对称情形	正文指出，攻击者必须伪造两个分量，并且只要至少一个分量未被攻破，组合方案就是 EUF-CMA；底层规范指出，当一个分量被攻破时该保证会降级	main:447, main:469, s0096, s0101, s0098
`host_key_after_session`	在连接结束后从服务器存储中读取到的组合宿主私钥（两个分量私钥）	RFC 4251 要求会话密钥具备前向保密，并警告“先收集、后解密”的收集方式；存储的宿主密钥是长期值，日后可能泄露	rfc4251:1021, main:96, s0187, s0197

ID	`network`	`broken_component`	`host_key_after_session`
A03	成立		
A02	成立		

ID	network	broken_component	host_key_after_session
A08	成立		
A04	成立		
K01	成立		成立
C08		成立	
K04	成立		
D01	成立		
U01		成立	
A08.10	成立		
U02	成立		
X01	成立		
X02	成立		

原文和目标

Source: 228 units, 37431 bytes, frozen in source-delivery-01.json. Targets: 10 approved (A03, A02, A08, A04, K01, C08, K04, D01, U01, U02); the questions asked: 3. The list of targets was approved as drafted, beforehand, with --approve-targets.

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

- approve_package: 批准源文包 protocol-8bf82a0f2d0f8709318672c88f08

该步骤的自我报告：1 次请求（1 次失败），0.000 USD，输入 31993 token，输出 4895 token，耗时 31 s。

关于证明器和模型的说明

- A03：在 ProVerif 中作答，分析者为该目标指定了 dh_algebra
- A04：在 ProVerif 中作答，分析者为该目标指定了 dh_algebra
- K01：在 ProVerif 中作答，分析者为该目标指定了 dh_algebra
- X01：在 ProVerif 中作答，分析者为该目标指定了 dh_algebra
- X02：在 ProVerif 中作答，分析者为该目标指定了 dh_algebra

简化代数（xor、dh_algebra、other_algebra）的 ProVerif 模型给攻击者的能力弱于真实运算：其中的证明是针对更弱攻击者的证明，其中的攻击则是真实的攻击。省略了对会变化的值的检查（changing_state）的 ProVerif 模型让参与方接受更多：其中的证明仍然成立，其中的攻击可能只是缺少了该检查。Tamarin+ 按原样对这些建模。

分析者为各目标指定的内容：

步骤	目标	特性	理由
29	A03, A04, K01, X01, X02	dh_algebra, many_sessions	这些目标依赖混合密钥交换：X25519 用单一 Diffie-Hellman 定律 g^xy=g^yx 建模，不含其他曲线代数，ML-KEM 是带有匹配解封装的自由封装函数，哈希是理想的。ProVerif 覆盖任意数量的会话，但曲线和编码的代数定律被简化了。

步骤	目标	特性	理由
29	A02, A08, A08.10, C08, D01, K04, U01, U02	plain_primitives, many_sessions	这些目标只依赖组合宿主签名：ML-DSA-44 和 Ed25519 是带有匹配验证的理想签名方案，组合合成器及其 M' 前缀由一个函数表示，SHA-512 是理想的。ML-DSA 分量被攻破的情形由该分量的显式伪造函数建模；经典情形是对称的解读。

分析者的收尾说明

该模型覆盖 draft-ietf-sshm-composite-sigs-00 配合 mlkem768x25519-sha256 的正常流程：双向 KEXINIT、SSH_MSG_KEX_HYBRID_INIT、带 K_S、S_REPLY 和两个分量签名的应答、K = HASH(K_PQ || K_CL)、覆盖 V_C、V_S、I_C、I_S、K_S、C_INIT、S_REPLY 和 K 的 exchange hash、由 HASH(K || H || letter || session_id) 得到的方向密钥，以及一个会检查两个分量、密钥和签名类型名及其 known-hosts 条目的客户端。

全部十三个目标在 model.pv 上都有答案，且都是带有失败对照的证明：对宿主密钥、exchange hash 和共享秘密的一致（A03、A08、A08.10、D01、U02，在破分量攻击者下还有 C08 和 U01）、单射一致（A04、X02）、宿主私钥和会话密钥的保密性（K01，包括针对连接结束后从存储中读取宿主密钥的前向保密，以及 exchange hash，X01）、当一个分量签名算法被攻破时的不可伪造性质（C08、U01），以及方向密钥分离（K04）。对照运行在客户端只验证一个分量、服务器读出自身会话状态、两个方向使用同一个派生输入的情况下否证了相同的性质，这表明这些证明依赖于它们所指名的检查。

无发现：没有出现文稿未说明的弱点。更大的阻碍在于：模型对原语做了理想化处理（组件被攻破就等同于具备伪造能力，X25519 只有一条 Diffie-Hellman 定律，ML-KEM 和哈希是理想的），因此计算性主张以及按攻击者类别给出的退化在这里都无法核查；文稿讨论的 SUF-CMA 失败与不可分离性，需要确定性可延展性或密钥重用方面的推理，而符号证明器无法表达这些；此外，RFC 10042 的长度检查、密钥更新、用户认证和加密数据通道都在模型范围之外。K04 只得到了部分回答：禁止密钥重用是一条关于生成密钥的规则，而不是一次运行的 property。