基于 CMP KEM 的 MAC 保护的安全性分析

本报告的其他语言版本：English。

rfc9810 · ProVerif+ Agent 报告

文稿	Internet X.509 Public Key Infrastructure -- Certificate Management Protocol (CMP) （协议定义）
检查的性质	13 条：9 条成立，4 条不成立（6 个攻击）
发现	1

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

CMP 可以用基于 KEM 共享秘密派生的 MAC 密钥来保护 PKI 管理消息，这样一来，持有 KEM 密钥对的 PKI 实体无需签名即可认证它发送的消息。Alice 持有 KEM 密钥对；Bob 持有她的可信公开 KEM 密钥，例如来自一份经过验证的 KEM 证书。这里建模的交换是一次 PKI 管理操作，而不是 CMP 原本承载的证书签发交换。

Bob 用 Alice 的公钥进行封装，并把密文连同 transactionID、他的 senderNonce 以及可选的 kemContext 一起发送。Alice 进行解封装，以 transactionID 和 kemContext 作为上下文从共享秘密派生 MAC 密钥，并对消息计算 MAC。Bob 派生出相同的密钥并验证 MAC，通过这次验证，他认证了 Alice 在本操作中的身份。

参与方：

- **Alice**：持有 KEM 密钥对，保护她发送的消息。持有：sk（KEM 私钥）、pk。
- **Bob**：持有 Alice 的可信公开 KEM 密钥，负责封装和验证。持有：pk（Alice 的 KEM 密钥，来自经过验证的证书）。

图 1 展示了文稿所规定的交换流程。

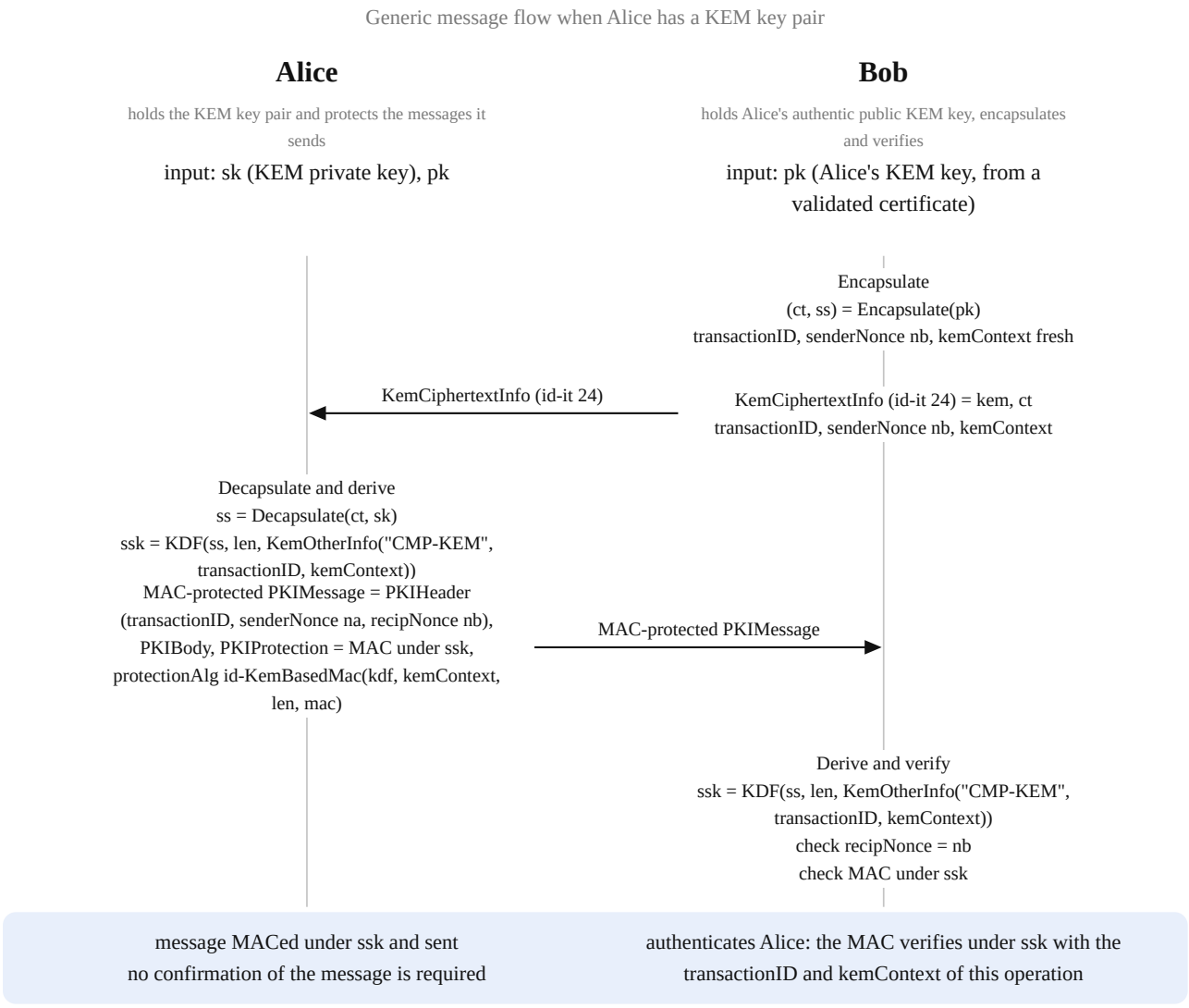


图 1. 规范规定的协议。

分析的范围见附录 D：分析了什么、省略了什么，以及构建分析所依据的文稿章节。

2 发现

本次分析有 1 项发现：以下内容在文稿中并未写明。其中 1 项是否已在文稿中写明，尚不明确。

基于 KEM 的 MAC 保护在网络攻击者面前达到了文稿所要求的保证：Bob 的接受与 Alice 的一次真实运行绑定，与他自己的 transactionID、nonce 和 kemContext 绑定，与消息字段绑定；并且无论进行多少次操作，共享秘密和 MAC 密钥都保持机密。新出现一个缺口：在单次操作内部，nonce 检查无法区分消息，因此从网络记录的一条受保护消息若作为同一操作的下一条消息重放，会被第二次接受。文稿称 nonce 可防止重放，并要求每次操作都进行新的封装，但从未要求接收方记录它已接受的 senderNonce，而且在整个操作中复用同一个 MAC 密钥。接收方记录已接受的 senderNonce 即可消除该攻击；该拟议修改已在展开模型上检查过，其余所有结果保持不变。前向保密结果以及分析所复现的 KEM 私钥泄露的影响，文稿本身已有说明。所有结果均针对理想的符号化 KEM、KDF 和 MAC，以及单向的一对发送方与接收方成立。

攻击图使用图 1 的列。红色表示攻击者的行为。黄色标记的是参与方发送了规范交换在该位置本不应有的消息，以及攻击所破坏的结果。

2.1 同一操作内重放的受保护消息被接受两次

类型	协议规范本身的弱点
是否新发现	不确定：文稿只写了一部分，或没能检索
价值	中
涉及的性质	A04, R02 （见第 3 节）
原文	s0014, s0075
依据	ProVerif attack queries `a04_b`, `r02_b`, `a04_inj`; Figures 2 and 3

发生了什么。 Bob 为每次管理操作派生一个 MAC 密钥，并在该操作中 Alice 发送的每条受保护消息上使用它；而这些消息都携带相同的 recipNonce，即 Bob 的 senderNonce。接收方只比较这个 nonce，而只要 Bob 不再发送消息，它就不会改变，并且 MAC 在复用的密钥下仍能通过验证。因此，从网络记录的一条消息若作为该操作的下一条消息再次发送，会同时通过这两项检查，其 PKIBody 所请求的动作会被第二次执行。

攻击者需要什么。 只需要网络：攻击者记录某次操作中由 Alice 发送的多条消息中的一条，扣留其后的一条，再重放所记录的那条。无需任何密钥，也无需攻破任何原语。

攻击者能得到什么。 本应只处理一次的消息被处理两次，Bob 无法将重放与操作中真实的后续消息区分开。损失的是第 5.1.1 节归于 nonce 的重放保护；KEM 秘密和 MAC 密钥不受影响。

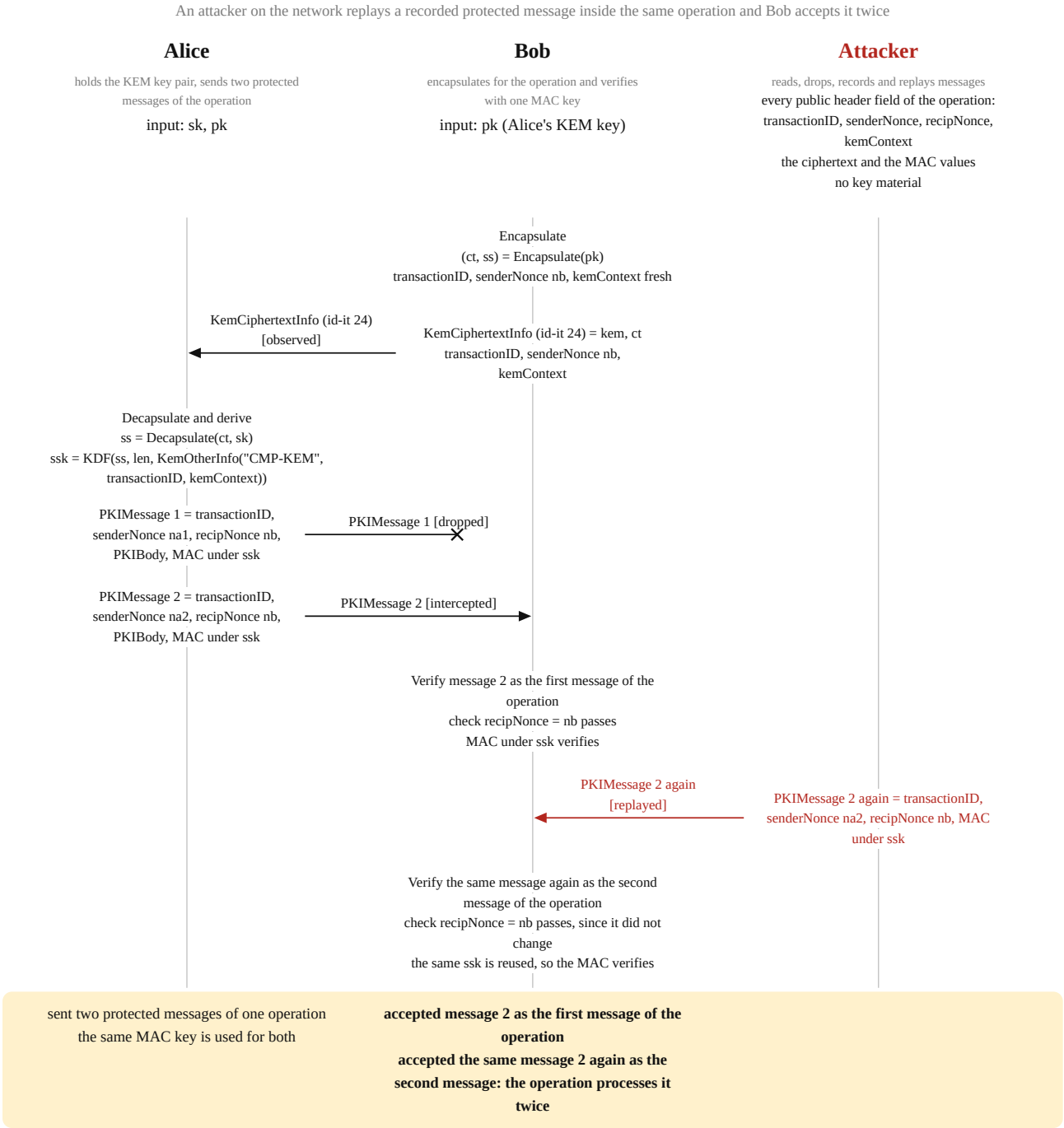


图 2. 网络上的攻击者在同一操作内重放记录到的受保护消息，Bob 接受了两次。性质 A04；攻击查询 a04_b、a04_inj。

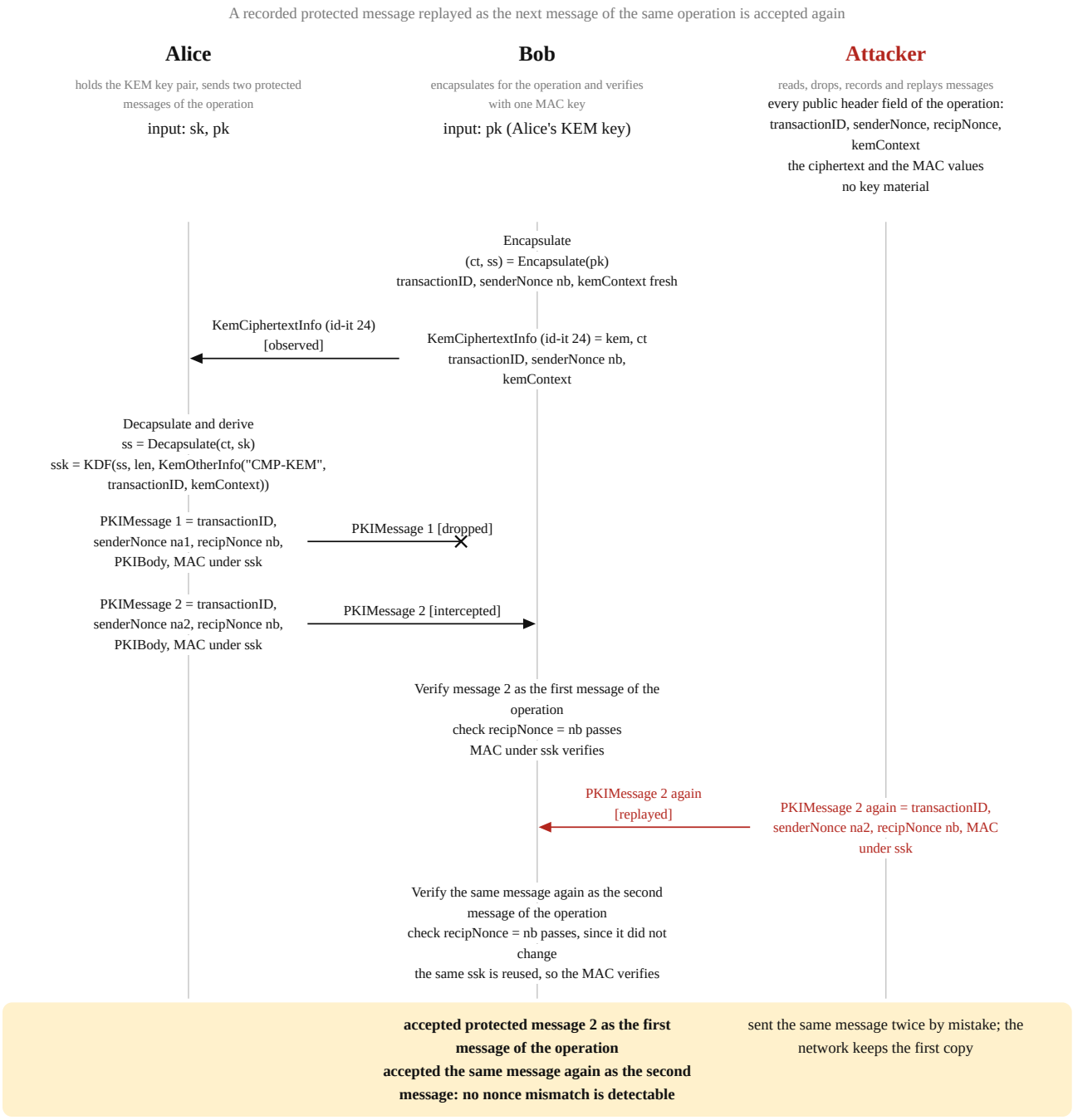


图 3. 记录到的受保护消息作为同一操作的下一条消息重放时被再次接受。性质 R02；攻击查询 r02_b。

拟议修改。 要求 MAC 保护的 PKIMessage 的接收方在同一事务内保留已接受的 senderNonce 值，并拒绝 senderNonce 已被接受过的消息，failInfo 为 badSenderNonce。该拟议修改已在展开的两消息模型上实施：两个攻击查询在该模型上均得到验证，且没有其他已记录答案发生变化。

用 ProVerif 复核。 拟议修改被加入模型的副本：MAC 保护的 PKIMessage 的接收方在同一事务内保留它接受的第一条受保护消息的 senderNonce，并且必须拒绝同一事务中 senderNonce 重复已接受值的后续消息（failInfo badSenderNonce）。在修改后的模型上，ProVerif 证明了 `a04_b`：该攻击在该模型中不再可能。在其他已记录的答案中，0 条仍成立，1 条发生变化：`r02_b`（R02，原为攻击）现在得到验证。这是对模型的初步检查：它表明该拟议修改在该模型中消除了这一攻击，而非表明它不会引入其他问题。修改后的模型：`records/fixes/0028-fix-nonce.pv`。

用 ProVerif 复核。 拟议修改被加入模型的副本：MAC 保护的 PKIMessage 的接收方在同一事务内保留它接受的第一条受保护消息的 senderNonce，并且必须拒绝同一事务中 senderNonce 重复已接受值的后续消息（failInfo badSenderNonce）。在修改后的模型上，ProVerif 证明了 `r02_b`：该攻击在该模型中不再可能。在其他已记录的答案中，0 条仍成立，1 条发生变化：`a04_b`（A04，原为攻击）现在得到验证。这是对模型的初步检查：它表明该拟议修改在该模型中消除了这一攻击，而非表明它不会引入其他问题。修改后的模型：`records/fixes/0028-fix-nonce.pv`。

对于该发现中的这些攻击（A04: r02_b, R02: a04_b, R02: a04_inj），未给出图。

3 检查过的性质

每一行是一条提交给 ProVerif 的性质。结果是 ProVerif 在分析者的协议模型上给出的答案；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	**Bob 认证 Alice。** Bob 每次成功验证 Alice 的 MAC 保护消息时，都存在 Alice 的一次真实运行，双方对角色、身份以及本次管理操作的数据（transactionID、nonce、kemContext 等）一致。	无	成立	中	是
A04	**每次接受对应不同的 Alice 会话。** Bob 的每次 MAC 验证成功都对应 Alice 的一次不同运行；同一受保护消息不能被 Bob 接受两次。	无	不成立（2 个攻击）	中	不确定
D01	**MAC 绑定消息字段。** Bob 接受的每条受 MAC 保护的 PKIMessage，其 PKIHeader、PKIBody、transactionID、nonce 与 kemContext 都对应 Alice 的真实发送，未被攻击者篡改。	无	成立	中	是
K01	**KEM 秘密保持机密。** 攻击者不能从公开的 ct、受保护消息和公钥推出 ss、ssk 或 Alice 的 KEM 私钥 sk。	无	攻击成立（文稿已写明）	中	是
K04	**操作之间的密钥分离。** 不同 PKI 管理操作、不同方向派生的 ssk 互不相同；同一操作内复用 ssk 是允许的。	无	成立	中	是
R02	**操作内无重放。** 在单次管理操作范围内，Bob 对同一逻辑消息至多接受一次；重放的旧消息因 nonce 不匹配而被拒绝。	无	不成立（1 个攻击）	中	不确定
R01	**新鲜的 KEM 证据与随机数。** Bob 接受的 nonce 与 KEM 封装证据是本次管理操作新产生的，旧证据不能被当作本次有效认证。	无	成立	中	是
A06	**向 Bob 的密钥确认。** Bob 在 MAC 验证成功时，确认 Alice 已导出并持有同一 ssk；该确认针对本次会话与上下文。	无	成立	中	是
A08	**操作上下文一致性。** Bob 接受的受保护消息所携带的 transactionID、kemContext 与算法参数，与本次管理操作中 Bob 实际发送的封装上下文一致。	无	成立	中	是
A03.10	**Bob 认证 KEM 密钥持有者。** Bob 每次成功验证 MAC 时，都存在持有 Alice KEM 私钥的 Alice 的真实运行，且双方对身份与会话数据一致。	无	成立	中	是

ID	性质	假定的泄露	结果	价值	文稿是否已写
D01.11	**头部与主体绑定。** Bob 接受的受保护消息中，PKIHeader、PKIBody、transactionID 与 nonce 都对应 Alice 的真实发送，未被替换。	无	成立	中	是
U01	**跨操作无 ss/ct 复用。** 不同管理操作之间 ss 与 ct 不重用；Bob 拒绝属于其他事务或已接受过的消息。	无	成立	中	是
X01	**MAC 密钥的前向保密。** 攻击者在某次 PKI 管理操作结束后才获得 Alice 的 KEM 私钥时，无法推导出该操作的 ssk，即记录下来的密文和受保护消息不会泄露过去的密钥。	Alice 的 KEM 私钥 sk，在操作完成后获得	攻击成立（文稿已写明）	低	是

“攻击成立（文稿已写明）”：证明器找到了这个攻击，而文稿本身已经写明它存在。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.

所提出的问题，以及分析者的回答：

问题	回答	依据
Q1: Bob 对持有 Alice KEM 私钥者的认证	是：面对网络上的攻击者，Bob 验证成功意味着持有 Alice KEM 私钥的一方以相同的 transactionID、nonce 和 kemContext 运行过一次；唯一的例外是同一份真实消息在一次操作内可能被计数两次。	A03, A03.10, A04, R02
Q2: MAC 对 PKIHeader/PKIBody、transactionID 和 nonce 绑定	MAC 对它们进行了绑定：Bob 只接受 transactionID、recipNonce 和 kemContext 均为自己的、且主体与 senderNonce 均为 Alice 所发送的消息，因此被修改的字段会被拒绝。	D01, D01.11, A08
Q3: ss/ct 不跨 management operation 重用，跨事务重放检查	跨操作时 ss 和 ct 均不会被重用，携带其他 transactionID 的消息会被拒绝；缺口在于一次操作内的重放，同一份消息会被第二次接受。	U01, K04, R02, A04

4 本次分析的局限

这里的结果是关于协议模型的结果：它说明的是所规定的消息和检查能提供什么、不能提供什么，其中密码学被视为理想原语，而不涉及任何具体实现。

按照分析者的说明，本次分析不覆盖以下内容：

- 只对基于 KEM 的通用 MAC 消息保护流程建模。证书签发、持有证明、RA 注册以及其他 CMP 消息类型不在范围之内。
- KEM、KDF 和 MAC 均为理想的符号原语。关于 IND-CCA2 安全性、KDF 的强度或 MAC 的不可伪造性，均未作任何证明。
- kemContext 被建模为 Bob 选择并随密文发送的一个值；原文将其内容交给算法决定，并未说明由谁固定，因此从对端获取该值的部署不在覆盖范围内。
- Bob 被赋予 Alice 的真实公钥 KEM 密钥，且模型中只存在这一对参与方，因此不考察两个发送者或两个 KEM 证书之间的混淆。
- 只对 Alice 到 Bob 的一个方向上的受保护消息建模；该流程还允许响应方在另一方向上用自己的 KEM 密钥对执行该过程。
- 一次操作内的重放是在展开为两条受保护消息、且 PKIBody 为不透明值的操作上分析的；重复消息的实际影响取决于 CMP 消息类型。
- 可用性、时序、消息大小以及随机数生成器的质量均未建模。

宿主记录的内容：

- A04、R02 的回答依赖于模型所简化的某次操作（附录 D）。

每条性质还有哪些未决之处、以及什么能解决它们，见附录 A。

附录 A：逐条性质的解读

这是分析者本人对证明器回答的解读，而非证明器的回答：这些回答对协议意味着什么、解读依据什么、能否按原样使用，以及它有多大价值。

A03: Bob 认证 Alice

Bob 每次成功验证 Alice 的 MAC 保护消息时，都存在 Alice 的一次真实运行，双方对角色、身份以及本次管理操作的数据（transactionID、nonce、kemContext 等）一致。

面对网络上的攻击者，Bob 只有在 Alice 以相同的 transactionID、senderNonce、recipNonce、kemContext 和主体运行过时才会接受受保护消息。没有 Alice 的 KEM 私钥就无法生成 MAC 密钥，因此接受即指明是 Alice 和本次操作。

- 结果：成立；回答了所问：是；可在有保留的情况下使用
- 价值 中：该流程规定 Bob 通过验证 MAC 来认证 Alice。
- 文稿中有此陈述：是。文稿中的位置：s0074
- 依据：KEM、KDF 和 MAC 为理想的符号原语；只有一对发送者-接收者：对端由 Bob 持有谁的公钥 KEM 密钥确定；只有一个方向
- 下一步：对具有不同 KEM 证书的两个发送者建模，以检验 Bob 的接受能说明它指的是哪一个身份。

A03.10: Bob 认证 KEM 密钥持有者

Bob 每次成功验证 MAC 时，都存在持有 Alice KEM 私钥的 Alice 的真实运行，且双方对身份与会话数据一致。

Bob 验证成功意味着持有 Alice KEM 私钥的一方以相同的 transactionID、nonce 和 kemContext 运行过一次。网络上的攻击者没有该密钥就无法生成这样的消息。

- 结果：成立；回答了所问：是；可在有保留的情况下使用
- 价值 中：文稿将 Alice 定义为拥有 KEM 密钥对、提供基于 MAC 保护的那个实体。
- 文稿中有此陈述：是。文稿中的位置：s0047, s0074
- 依据：理想的 KEM 和 MAC；对端的身份由 Bob 持有哪一份公钥 KEM 密钥确定
- 下一步：与认证目标一样，包含两个密钥持有者的模型可以解决身份混淆问题。

A04: 每次接受对应不同的 Alice 会话

Bob 的每次 MAC 验证成功都对应 Alice 的一次不同运行；同一受保护消息不能被 Bob 接受两次。

跨操作时，每次接受都对应 Alice 的一次不同运行，因为 senderNonce 和共享密钥对每次操作都是新的。在一次操作内，同一受保护消息会被接受两次，正如展开模型所示：recipNonce 保持不变，MAC 密钥被重用。

- 结果：不成立（2 个攻击）；回答了所问：是；可在有保留的情况下使用
- 价值 中：第 5.1.1 节说 nonce 能防重放，第 8.8 节要求每次操作都进行新的封装，但文稿并未说明一次操作内重复的消息会被接受。
- 文稿中有此陈述：不明确。所依据的原文：s0014, s0097

- 依据：一次操作内的重放是在两条消息的展开、且 PKIBody 为不透明值的情况下分析的；模型为 Bob 的每次操作给出新的 transactionID 和新的封装
- 未决：重复消息的危害取决于 PKIBody 的动作，而该动作未被建模
- 下一步：对一次操作中实际的 CMP 消息类型建模，以便能指明被重复的动作。

D01: MAC 绑定消息字段

Bob 接受的每条受 MAC 保护的 PKIMessage，其 PKIHeader、PKIBody、transactionID、nonce 与 kemContext 都对应 Alice 的真实发送，未被攻击者篡改。

被接受消息的每个字段都通过 MAC 与 Alice 的发送绑定，且 Bob 会将自己的操作与 transactionID、recipNonce 和 kemContext 进行核对。途中被修改的字段，或被移到另一操作的消息，都无法通过验证。

- 结果：成立；回答了所问：是；可在有保留的情况下使用
- 价值 中：文稿说明受保护的头部信息与消息一同受到保护，且 PKIProtection 携带对整条消息的 MAC。
- 文稿中有此陈述：是。文稿中的位置：s0002, s0048, s0074
- 依据：理想 MAC：字段改变则 MAC 值改变；PKIBody 被建模为一个不透明值，因此不考察主体内部的逐字段结构
- 下一步：对已建模的字段而言，无。

D01.11: 头部与主体绑定

Bob 接受的受保护消息中，PKIHeader、PKIBody、transactionID 与 nonce 都对应 Alice 的真实发送，未被替换。

被接受的消息携带 Alice 的 PKIBody 和 senderNonce 以及 Bob 的 transactionID、recipNonce 和 kemContext，全部由 MAC 覆盖。被修改的头部字段或主体会被拒绝。

- 结果：成立；回答了所问：是；可在有保留的情况下使用
- 价值 中：文稿说明保护覆盖整条消息，且 transactionID 参与密钥派生。
- 文稿中有此陈述：是。文稿中的位置：s0002, s0082
- 依据：理想 MAC；不透明的主体是一个值，而不是字段结构
- 下一步：对已建模的字段而言，无。

A06: 向 Bob 的密钥确认

Bob 在 MAC 验证成功时，确认 Alice 已导出并持有同一 ssk；该确认针对本次会话与上下文。

Bob 的接受确认了 Alice 为同一操作和上下文导出了相同的 MAC 密钥，因为 MAC 只在该密钥下才能验证通过。该确认的强度取决于密钥的保密性。

- 结果：成立；回答了所问：是；可在有保留的情况下使用
- 价值 中：文稿说明 Bob 用共享密钥验证 MAC，并以此认证 Alice。
- 文稿中有此陈述：是。文稿中的位置：s0074, s0075
- 依据：不存在单独的密钥确认消息；该确认就是 MAC 验证成功；理想 MAC
- 下一步：无。

A08: 操作上下文一致性

Bob 接受的受保护消息所携带的 transactionID、kemContext 与算法参数，与本次管理操作中 Bob 实际发送的封装上下文一致。

Bob 只接受携带他自己的 transactionID 和 kemContext 的消息，而 transactionID 参与密钥派生，因此来自另一操作或带有另一上下文的消息无法通过验证。

- 结果：成立；回答了所问：是；可在有保留的情况下使用
- 价值 中：文稿要求 transactionID 为携带密文的那条消息的值，并将其用于派生密钥的域分离。
- 文稿中有此陈述：是。文稿中的位置：s0080, s0082, s0083
- 依据：kemContext 被建模为 Bob 选择并随密文发送的一个值；原文对其内容未作限定；理想 KDF 和 MAC
- 下一步：kemContext 来自对端的部署应单独建模。

K01: KEM 秘密保持机密

攻击者不能从公开的 ct、受保护消息和公钥推出 ss、ssk 或 Alice 的 KEM 私钥 sk。

网络上的攻击者无论进行多少次操作，都无法从密文、受保护的消息和公钥中推导出共享密钥、MAC 密钥或 Alice 的 KEM 私钥。若攻击者在事后取得 Alice 的 KEM 私钥，就能恢复某次已记录操作的共享密钥和 MAC 密钥，文稿将此表述为此类泄露的后果。

- 结果：攻击成立（文稿中已说明）；是否回答了所问：是；可谨慎使用
- 值 中：文稿指出，KEM 私钥泄露可能导致所有受该密钥保护的内容被披露。
- 文稿中已说明：是。文稿出处：rfc9629:568
- 依据：KEM 是理想的：没有私钥，密文不泄露关于共享密钥的任何信息；MAC 仅保护完整性，因此 PKIBody 在链路上可读；密钥泄露发生在操作之后，属于后续阶段
- 下一步：无。

K04: 操作之间的密钥分离

不同 PKI 管理操作、不同方向派生的 ssk 互不相同；同一操作内复用 ssk 是允许的。

只有当两个操作携带相同的 transactionID 时，它们才会共享 MAC 密钥，因为每次操作都重新封装，且 transactionID 参与密钥派生。密钥无法从一个操作转移到另一个操作。

- 结果：成立；是否回答了所问：是；可谨慎使用
- 值 中：文稿指出 transactionID 确保派生密钥在操作之间实现域分离，并且每次操作都需要重新封装。
- 文稿中已说明：是。文稿出处：s0082, s0097
- 依据：理想 KDF：不同输入产生不同密钥；模型为 Bob 的每次操作提供新的 transactionID，如同一个对给定 ID 至多使用一次事务的客户端
- 下一步：无。

R01: 新鲜的 KEM 证据与随机数

Bob 接受的 nonce 与 KEM 封装证据是本次管理操作新产生的，旧证据不能被当作本次有效认证。

Bob 用于接受的 nonce 和封装证据是他为本次操作生成的，因此先前操作的证据不能作为当前认证。当操作携带两条受保护消息时同样如此。

- 结果：成立；是否回答了所问：是；可谨慎使用
- 值 中：文稿指出 nonce 保护消息免受重放，并要求每次操作重新封装。
- 文稿中已说明：是。文稿出处：s0014, s0097
- 依据：在展开模型中每次操作有两条受保护消息；ct 和 nonce 使用新鲜随机数
- 下一步：无。

R02: 操作内无重放

在单次管理操作范围内，Bob 对同一逻辑消息至多接受一次；重放的旧消息因 nonce 不匹配而被拒绝。

跨操作时，重放的消息会失败，因为 Bob 的 senderNonce 和共享密钥对每次操作都是新的。在单次操作内，被记录的消息作为下一条消息重放时会被再次接受，因为 Alice 发送的每条消息都携带相同的 recipNonce 且 MAC 密钥被复用，因此无法检测到 nonce 不匹配。

- 结果：不成立（1 次攻击）；是否回答了所问：是；可谨慎使用
- 值 中：文稿指出 nonce 可防止重放，但未说明操作内重复的消息会被接受，也未说明接收方必须记录已接受的 senderNonce。
- 文稿中已说明：不明确。所依据的原文：s0014
- 依据：操作被展开为两条受保护消息；模型不记录已接受的 senderNonce，因为原文不要求；PKIBody 是不透明的，因此未指明被重复的动作
- 待解决：重复的后果取决于消息类型
- 下一步：对操作的消息类型建模，以指明被重复的动作。

U01: 跨操作无 ss/ct 复用

不同管理操作之间 ss 与 ct 不重用；Bob 拒绝属于其他事务或已接受过的消息。

每次操作都获得新的密文、共享密钥和 MAC 密钥，transactionID 不是 Bob 自己的消息会被拒绝。ss 与 ct 在操作之间均不复用，其他事务的消息无法通过验证。

- 结果：成立；是否回答了所问：是；可谨慎使用
- 值 中：文稿要求每次操作重新封装，并禁止复用 ss 和 ct。
- 文稿中已说明：是。文稿出处：s0064, s0097
- 依据：理想 KDF 和 MAC；按原文要求每次操作使用新鲜随机数
- 下一步：无。

X01: MAC 密钥的前向保密

攻击者若在 PKI 管理操作完成之后才取得 Alice 的 KEM 私钥，则无法推导出该操作的 ssk，即记录的密文和受保护消息不会泄露过去的密钥。

该方案对 MAC 密钥没有前向保密：攻击者在操作之后取得 Alice 的 KEM 私钥，就能解封装记录的密文，并从公开的 transactionID 和 kemContext 重新推导出 MAC 密钥，因此可以验证和伪造该操作的消息。文稿指出 CMP 消息保护不需要前向保密，因此这是预期结果。

- 结果：攻击成立（文稿中已说明）；是否回答了所问：是；可用
- 值 低：CMP 文稿第 8.5 节指出消息保护不需要完美前向保密，RFC 9629 指出 KEM 私钥泄露可能披露受其保护的所有内容。
- 文稿中已说明：是。文稿出处：main:3875, rfc9629:568

- 依据：攻击者在操作之后（模型的后续阶段）获得 KEM 私钥；在符号模型中，使用私钥解封装是精确的
- 下一步：无。

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，以及分析者对每次攻击的判定。

目标	级别	界	查询	对照	攻击	主张
A03	proved		a03	checked		1 proved
A04	proved		a04	checked	a04_b (holds_in_source); a04_inj (holds_in_source)	1 proved, 2 attack
D01	proved		d01	checked		1 proved
K01	proved		k01_ss	checked	x01_ss (holds_in_source); x01_ssk (holds_in_source)	3 proved, 2 attack
K04	proved		k04	checked		1 proved
R02	proved		r02	checked	r02_b (holds_in_source)	1 proved, 1 attack
R01	proved		r01	checked		1 proved
A06	proved		a06	checked		1 proved
A08	proved		a08	checked		1 proved
A03.10	proved		a03	checked		1 proved
D01.11	proved		d01	checked		1 proved
U01	proved		u01	checked		2 proved
X01	none				x01_ssk (holds_in_source)	1 attack

逐条主张，按分析者对每条主张的表述；每条主张的答案是证明器给出的：

目标	主张	攻击者	回答	查询	对照
A03	Bob 的每一次成功 MAC 验证都对应一次真实的 Alice 运行，且 transactionID、senderNonce、recipNonce、kemContext 和 PKIBody 相同（角色与身份如流程中所述，Alice 持有 KEM 私钥）。	`network`	proved	a03	checked
A04	Bob 的每一次接受都对应一次不同的 Alice 运行：同一条受保护消息不会被接受两次。	`network`	proved	a04	checked
A04	在一次管理操作内，同一条受保护消息不会被 Bob 接受两次。	`network`	attack (holds_in_source)	a04_b	
A04	在一条包含两条消息的管理操作中，Bob 的每一次接受都对应 Alice 的一次不同发送（在展开模型上的单射一致性）。	`network`	attack (holds_in_source)	a04_inj	
D01	被接受消息的 PKIHeader 字段、PKIBody、transactionID、senderNonce/recipNonce 和 kemContext 都是 Alice 所发送的、未经修改的值。	`network`	proved	d01	checked
K01	一次操作的 ss 对网络上的攻击者保持保密。	`network`	proved	k01_ss	checked
K01	Bob 导出的 ssk 对网络上的攻击者保持保密。	`network`	proved	k01_ssk	checked
K01	面对在一次操作之后获得 Alice KEM 私钥的攻击者，该次被记录操作的 ss 可以从公开密文中恢复。	`kem_sk_leak`	attack (holds_in_source)	x01_ss	
K01	面对在一次操作之后获得 Alice KEM 私钥的攻击者，该次被记录操作的 ssk 可以从公开密文、transactionID 和 kemContext 中恢复。	`kem_sk_leak`	attack (holds_in_source)	x01_ssk	

目标	主张	攻击者	回答	查询	对照
K01	Alice 的 KEM 私钥 sk 对网络上的攻击者保持保密。	`network`	proved	k01_sk	checked
K04	不同管理操作的 ssk 值互不相同（相同的 ssk 只在相同 transactionID 下出现）。	`network`	proved	k04	checked
R02	重放的受保护消息不会被 Bob 接受：接受在 Alice 的运行上是单射的，并且绑定到 Bob 的新鲜 nonce。	`network`	proved	r02	checked
R02	在一次管理操作内，Bob 对给定的受保护消息最多接受一次；重放的旧消息因 nonce 不匹配而被拒绝。	`network`	attack (holds_in_source)	r02_b	
R01	Bob 接受时所依据的 nonce 和 KEM 封装证据，正是他为这次操作所生成的。	`network`	proved	r01	checked
A06	Bob 的成功 MAC 验证确认 Alice 为这次操作和上下文派生并持有相同的 ssk。	`network`	proved	a06	checked
A08	Bob 接受时所依据的 transactionID 和 kemContext，正是他为这次操作所做的封装所对应的值。	`network`	proved	a08	checked
A03.10	Bob 的成功验证意味着持有 Alice KEM 私钥的一方进行了一次运行，且 transactionID 和 nonce 匹配。	`network`	proved	a03	checked
D01.11	被接受的受保护消息，其 PKIHeader、PKIBody、transactionID 和 nonce 都对应 Alice 的真实发送。	`network`	proved	d01	checked
U01	Bob 拒绝属于另一事务的受保护消息：被接受消息的 transactionID 正是他自己封装所用的那个。	`network`	proved	u01	checked
U01	ss 和 ct 不会在不同管理操作之间复用：两次操作只有在携带相同 transactionID 时才共享同一个 ssk。	`network`	proved	k04	checked
X01	只有在操作结束之后才获得 Alice KEM 私钥的攻击者，无法派生该次操作的 ssk。	`kem_sk_leak`	attack (holds_in_source)	x01_ssk	

附录 D：分析是怎么跑的

一名 LLM 分析者根据文稿建立协议模型，并把每条性质交给 ProVerif；第 3 节中的每个结果都是证明器给出的回答。

ProVerif+ Agent 会话 session-20261114T230140，2026-11-14。状态：已完成。模型：deepseek-v4-1-flash-261017。步数：41，证明器 5 秒，实际耗时 10 分钟，费用 0.082 USD。

ProVerif：运行了 106 次证明，106 次有结论，0 次无答案，4 秒，记录了 27 个回答。

分析了什么

协议	CMP KEM-based MAC protection
版本	rfc9810
分析范围	选定 generic KEM-based MAC message-protection flow，Alice 有已认证 ML-KEM 公钥，Bob 产生新鲜 ss/ct，Alice MAC PKIMessage，Bob 验证；transactionID、senderNonce/recipNonce、角色与 kemContext 明确建模。不是完整证书签发或 RA 注册验证。
不在范围内	符号模型不证明算法的计算安全或抗量子安全；攻击轨迹只在明确假设下成立，不直接等同规范漏洞。每个性质必须有可达性/正例控制。
原文	来自下列章节的 100 个编号原文单元（23041 字节）。本报告中的 s0001 之类的引用就是这些原文单元之一

分析所依据的文稿章节：

文稿	章节	标题
main	5.1.1	PKI Message Header
main	5.1.1.1	ImplicitConfirm
main	5.1.1.2	ConfirmWaitTime
main	5.1.1.3	OrigPKIMessage
main	5.1.1.4	CertProfile
main	5.1.1.5	KemCiphertextInfo
main	5.1.3.4	Key Encapsulation
main	5.3.19.18	KEM Ciphertext
main	8.8	Recurring Usage of KEM Keys for Message Protection

源文步骤在文稿中为协议定位的目的：选定 generic KEM-based MAC message-protection flow，Alice 有已认证 ML-KEM 公钥，Bob 产生新鲜 ss/ct，Alice MAC PKIMessage，Bob 验证；transactionID、senderNonce/recipNonce、角色与 kemContext 明确建模。不是完整证书签发或 RA 注册验证。

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对照文稿本身所述协议用途进行了核对。性质的名称、它针对的内容以及文稿所述内容都是分析者的解读；所提到的地方在文稿中确实存在。

ID	简称	对象	文稿怎么说	出处
A03	Bob 认证 Alice	Bob 在 MAC 验证通过时确信：Alice（持有 KEM 私钥 sk 的一方）针对本次管理操作运行了协议，且 transactionID、nonces 与 kemContext 均相同。	流程以“Alice 已由 Bob 认证”结束；Bob 用共享密钥验证 MAC，而该密钥只有 Alice 能从密文推导出来。	s0047, s0061, s0074
A04	每次接受对应不同的 Alice 会话	Bob 的每一次成功接受都对应一次不同的 Alice 运行；任何受保护的消息都不会被接受两次。	Bob 不得将 ss 和 ct 复用于其他操作，每次操作都必须调用一次新的 Encapsulate()。	s0014, s0064, s0097
D01	MAC 绑定消息字段	被接受消息的 PKIHeader 字段、PKIBody、transactionID、nonces 和 kemContext 就是 Alice 发送的原始值，未被修改。	如果 PKI 消息受到保护，那么这些信息也受到保护；PKIProtection 携带对消息的 MAC。	s0002, s0048, s0074, s0080, s0081, s0082, s0083
K01	KEM 秘密保持机密	对于看到公开密文、受保护消息和公钥的攻击者，ss、ssk 以及 Alice 的 KEM 私钥 sk 均保持机密。	KEM 共享秘密只能由持有私钥的一方通过 Decapsulate 得出；基于 MAC 的保护依赖于派生出的 ssk。	s0044, s0045, s0046, s0069, s0072, s0099
K04	操作之间的密钥分离	不同 PKI 管理操作中派生出的 ssk 值各不相同，因此一个 ssk 仅限于其所属操作。	transactionID 用于确保不同操作之间派生共享密钥的域分离，且每次操作都需要新的 ss/ct。	s0064, s0075, s0082, s0083, s0097
R02	操作内无重放	在同一个管理操作内，Bob 对给定的受保护消息至多接受一次；旧消息因其 nonce 不匹配而被拒绝。	recipNonce 从事务中上一条消息的 senderNonce 复制而来；nonces 保护 PKIMessage 免受重放攻击。	s0014
R01	新鲜的 KEM 证据与随机数	Bob 接受时所依据的 nonce 和 KEM 封装证据属于本次操作，而不属于更早的操作。	senderNonce 和 recipNonce 保护 PKIMessage 免受重放；每次操作都使用新的 Encapsulate()。	s0014, s0064, s0097
A06	向 Bob 的密钥确认	Bob 的成功验证确认 Alice 已为本次操作和上下文推导出并持有相同的 ssk。	Bob 使用共享密钥验证所收到消息的基于 MAC 的保护，以此认证 Alice。	s0061, s0074, s0075
A08	操作上下文一致性	Bob 接受消息时所依据的 transactionID 和 kemContext 就是他自己为本次操作执行封装时所用的值。	transactionID 必须是包含密文的消息中的值，并用于域分离；kemContext 是双方共享的算法特定上下文。	s0012, s0013, s0080, s0081, s0082, s0083
A03.10	Bob 认证 KEM 密钥持有者	Bob 的成功 MAC 验证对应持有 Alice 的 KEM 私钥的一方的一次运行，且 transactionID 和 nonces 匹配（用户问题 Q1）。	Alice 被定义为拥有 KEM 密钥对并提供基于 MAC 保护的一方；Bob 对此进行验证。	s0047, s0061, s0074
D01.11	头部与主体绑定	被接受的受保护消息，其 PKIHeader、PKIBody、transactionID 和 nonces 对应 Alice 的真实发送（用户问题 Q2）。	保护覆盖整个 PKIMessage；transactionID 用于派生密钥的域分离。	s0002, s0048, s0074, s0082
U01	跨操作无 ss/ct 复用	ss 和 ct 不会在管理操作之间复用，Bob 会拒绝属于另一事务或已被接受过的消息（用户问题 Q3）。	Bob 不得将 ss 和 ct 复用于其他操作；服务器会拒绝已在使用中的 transactionID。	s0013, s0014, s0064, s0097
X01	MAC 密钥的前向保密	对于在操作完成后才获得 Alice 的 KEM 私钥 sk 的攻击者，已完成操作的 ssk 仍应保密	第 8.5 节指出，CMP 消息保护提供完整性和真实性，不需要完美前向保密；RFC 9629 指出，KEM 私钥泄露可能导致所有用该密钥保护的内容被泄露。	main:3875, rfc9629:568

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 74 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 6 段，没选 68 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:3800-3812	不在范围内	证书签发中的 POP；本任务只对基于 KEM 的 MAC 消息保护流程建模。
P02	main:3816-3830	不在范围内	通过解密证书做 POP；建模流程中没有证书签发。
P03	main:3832-3835	不在范围内	返回解密后 POP 挑战的规则；不属于消息保护。
P04 至 P07	main:3839-3852	不在范围内	通过向 CA/RA 暴露私钥做 POP；建模流程中没有密钥归档或 POP。
P08	main:3856-3871	不在范围内	针对密钥归档/恢复中 DH 密钥交换的小子群攻击；建模流程使用 KEM 而非 DH，也没有密钥恢复。
P09	main:3875-3883	X01, `kem_sk_leak`	
P10	main:3885-3889	不在范围内	CMS EnvelopedData（加密值）和 TLS 传输的前向保密；不在 MAC 保护流程范围内。
P11	main:3893-3896	不在范围内	将 CA 证书签名密钥与 CMP 消息保护密钥分离；建模流程中没有 CA 签名密钥。
P12	main:3899-3899	没有陈述目标	章节标题，未陈述任何性质。
P13	main:3901-3912	证明器问不了	用于生成 nonce 和私钥的 PRNG 质量；属于计算/实现性质。
P14	main:3914-3923	证明器问不了	共享秘密信息的熵、作为共享秘密的口令；KEM 产生高熵的 ss，猜测熵是计算性质。
P15	main:3925-3930	证明器问不了	共享秘密信息相对于交付密钥强度所需的熵；属于计算要求。
P16	main:3932-3943	不在范围内	交付信任锚时的熵要求；信任锚供给不在建模流程范围内。
P17	main:3945-3947	没有陈述目标	指向 RFC 9481 算法配置文件的指引，未陈述任何性质。
P18	main:3951-3954	K04, U01	
P19	main:3956-3958	证明器问不了	假设一个共享秘密密钥保护的总数据量很小；这是数据量/大小假设，不是迹性质。
P20	main:3960-3969	证明器问不了	要求 KEM 在公钥复用时满足 IND-CCA2 安全；属于密码原语的强度。
P21	main:3971-3973	证明器问不了	IND-CCA2 假设在公钥无限复用时带来的推论；属于计算性陈述。
P22 至 P24	main:3977-3995	不在范围内	使用 CMP 消息供给信任锚；不在建模流程范围内。
P25	main:3999-4004	不在范围内	对具有特定 EKU 的证书的授权；属于证书签发策略，不在建模流程范围内。
P26 至 P27	main:4008-4018	不在范围内	CT 日志发布与吊销；不在建模流程范围内。
P28 至 P52	rfc5280:5226-5449	不在范围内	RFC 5280 的证书与路径验证安全考虑；证书验证不属于建模流程。
P53	rfc9481:1050-1056	证明器问不了	所选算法的安全级别；属于原语强度。
P54	rfc9481:1058-1074	证明器问不了	基于口令的保护在 PBMAC1 与 PasswordBasedMac 之间的选择；属于算法选择，与 KEM 流程无关。
P55	rfc9481:1076-1078	证明器问不了	禁止将 AES-GMAC 用作 PBKDF2 的 PRF；属于原语层面的要求。

段落	出处	定为	不选的原因
P56	rfc9481:1080-1092	没有陈述目标	关于算法配置文件的迁移建议，未陈述本协议的任何性质。
P57	rfc9481:1094-1098	证明器问不了	为对称 MAC 算法选择合适的 PRF 和足够的密钥熵；属于原语强度。
P58	rfc9629:542-543	没有陈述目标	使 RFC 5652 的安全考虑适用；本身未陈述本协议的任何内容。
P59	rfc9629:545-554	证明器问不了	要求 KEM 在公钥多次使用时安全（IND-CCA2）；属于原语强度。
P60	rfc9629:556-556	证明器问不了	KDF 应至少提供 KEM 的安全级别；属于原语强度。
P61	rfc9629:558-561	证明器问不了	密钥加密算法和 KEK 的大小；属于原语强度，且不用于基于 MAC 的流程。
P62	rfc9629:563-566	A03, D01	
P63	rfc9629:568-572	K01, X01, `kem_sk_leak`	
P64	rfc9629:574-587	K04, U01	
P65	rfc9629:589-598	证明器问不了	密钥的随机生成和 PRNG 质量；属于计算/实现性质。
P66 至 P68	rfc9629:600-620	证明器问不了	有效安全性取两个算法尺寸中较弱者；属于原语强度。
P69	rfc9629:622-629	没有陈述目标	建议算法会随时间变弱、实现应模块化；未陈述协议的任何性质。
P70	rfc9935:406-407	不在范围内	RFC 5280 的安全考虑；证书验证不在建模流程范围内。
P71	rfc9935:409-411	K01, `kem_sk_leak`	
P72	rfc9935:413-421	证明器问不了	生成私钥所用 PRNG 的质量；属于计算性质。
P73	rfc9935:423-431	证明器问不了	ML-KEM 私钥格式的绑定性质（LEAK-BIND、MAL-BIND）；属于 KEM 的计算性质。
P74	rfc9935:433-435	没有陈述目标	指向详细的 ML-KEM 安全考虑的指引，未陈述任何性质。

参与方和受保护的值，逐项是否设为攻击者条件

宿主把起草的性质里点到的参与方和受保护的值逐项列出，共 18 项；分析者对每一项说明了哪个攻击者条件控制这个参与方或持有这个值，或者为什么不设。设了攻击者条件的 1 项，没设的 17 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	Alice	参与方	按定义全部失守	控制 Alice 的攻击者持有 KEM 私钥，能够产生 Alice 能产生的每一条消息，因此 Bob 对 Alice 的认证由攻击者自身即得到满足；没有其他目标问及这一点。
S02	Bob	参与方	攻击者得不到更多	Bob 只持有它自己生成的值（ss、ssk）和 Alice 的公钥；控制 Bob 无法得到 Alice 的秘密，也没有任何目标涉及 Bob 的诚实性。
S03	攻击者	参与方	攻击者得不到更多	网络攻击者已经是基线攻击者。
S04	ssk	值	按定义全部失守	把某次操作的 ssk 交给攻击者，按定义就使该次操作可被伪造；没有目标问及 ssk 保护的其他任何东西。
S05	ss	值	按定义全部失守	ss 通过公开的 KDF 决定 ssk，因此交出 ss 恰好丧失 K01 对该次操作所要求的保密性。
S06	ct	值	攻击者得不到更多	ct 以明文发送给 Alice，网络上的攻击者本来就已持有它。

ID	参与方或值	类别	攻击者条件	不设的原因
S07	transactionID	值	攻击者得不到更多	transactionID 以明文随头部和密文消息传输。
S08	senderNonce/recipNonce	值	攻击者得不到更多	senderNonce 和 recipNonce 以明文随头部传输。
S09	kemContext	值	攻击者得不到更多	kemContext 在保护参数中传递，是公开的上下文数据。
S10	senderNonce	值	攻击者得不到更多	senderNonce 是公开的头部数据。
S11	recipNonce	值	攻击者得不到更多	recipNonce 是公开的头部数据。
S12	PKIHeader 字段	值	攻击者得不到更多	PKIHeader 字段以明文发送；基于 MAC 的保护提供完整性，不提供保密性。
S13	PKIBody	值	攻击者得不到更多	PKIBody 不经过基于 MAC 的保护加密；只涉及其完整性，这正是 D01 所问的。
S14	MAC 值	值	攻击者得不到更多	MAC 值是公开的；知道它并不能得到任何密钥。
S15	Alice 的 KEM 私钥 sk	值	`kem_sk_leak`	
S16	kdf/len/mac 参数	值	攻击者得不到更多	kdf、len 和 mac 参数在 protectionAlg 中传递，是公开的。
S17	staticString	值	攻击者得不到更多	staticString 是固定的公开常量 "CMP-KEM"。
S18	PKIHeader	值	攻击者得不到更多	PKIHeader 以明文发送；其完整性正是 D01 所问的。

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；每个答案都在其中一个攻击者下记录。单元格是该性质在该攻击者下证明器给出的答案；“无答案”表示分析者为该性质设定了一个攻击者，但证明器让某个主张处于未决状态，或没有记录任何内容；空单元格表示分析者没有为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、更改和发送消息，不持有任何一方的秘密		
`kem_sk_leak`	Alice 的 KEM 私钥 sk，在其所针对的操作之后获得（对于 ss 和 ssk 的保密性而言这已足够，因为密文 ct 是公开的）	RFC 9629 指出实现必须保护 KEM 私钥，且 KEM 私钥的泄露可能导致用该密钥保护的所有内容被披露；RFC 9935 指出私钥材料的泄露可能导致冒充。	rfc9629:568, rfc9935:409

ID	`network`	`kem_sk_leak`
A03	成立	
A04	不成立	
D01	成立	
K01	成立	不成立
K04	成立	
R02	不成立	
R01	成立	
A06	成立	
A08	成立	
A03.10	成立	
D01.11	成立	

ID	network	kem_sk_leak
U01	成立	
X01		不成立

原文和目标

来源：100 个原文单元，23041 字节，冻结于 source-delivery-01.json。 目标：10 个已批准（A03、A04、D01、K01、K04、R02、R01、A06、A08、U01）；提出的问题：3 个。 目标清单事先按草稿批准，使用 --approve-targets。

宿主按其记录的默认设置做出的决定（任务 `decisions: auto`），供你核对：

- approve_package: 批准源文包 protocol-3a2c98f241f72293e2214c14eb4c

本步骤的自述：1 次请求（1 次失败），0.000 USD，输入 22983 tokens，输出 4281 tokens，28 秒。

关于证明器与模型的说明

- A04：在 ProVerif 中作答，分析者为此目标命名了 changing_state
- R02：在 ProVerif 中作答，分析者为此目标命名了 changing_state

对代数（xor、dh_algebra、other_algebra）做简化的 ProVerif 模型给攻击者的能力弱于真实运算：在其中的证明是针对更弱攻击者的证明，在其中的攻击就是攻击。省略对变化值（changing_state）检查的 ProVerif 模型让各方接受更多：在其中的证明成立，在其中的攻击可能只是缺少了那项检查。Tamarin+ Agent 按原样对这些建模。

分析者的状态审查

这部分是分析者自己对原文的解读，不是证明器的答案。ProVerif 所示一栏中的“是”表示 ProVerif 在展开模型中找到了包含这些步骤的轨迹，因此这些步骤是可能的；至于它们是否破坏某个目标，仅由答案表中的一次攻击说明，它们还带来什么代价则是分析者的解读。“未发现”只说明查看了哪些模式，不证明任何事。

值	由谁保留	何时变化	何时检查	引用
Bob 对该操作消息所期望的 recipNonce（他自己的 senderNonce nb）	Bob	当 Bob 发送该事务的新 PKIMessage 时；在此流程中 Bob 只发送 KemCiphertextInfo 消息，因此对于 Alice 在该操作中发送的每条消息，期望的 recipNonce 始终保持为 nb	对从 Alice 收到的每条受保护消息，这些消息必须携带 recipNonce = nb	s0014, s0068
Alice 发送的每条受保护消息的 senderNonce	Alice	对 Alice 发送的每条受保护 PKIMessage（每条消息使用新的 128 位随机值）	仅间接检查：Bob 在发送该事务的下一条消息时将其复制到 recipNonce；Bob 不会将其与他记住的任何内容进行比较	s0014
管理操作的 ssk	Alice 和 Bob	在操作内部从不变化；新操作会再次调用 Encapsulate，产生新的 ct、ss，因而产生新的 ssk	对收到的每条受保护消息，通过在 ssk 下验证 MAC 来检查	s0064, s0075, s0097
操作内部已接受过的 senderNonce 的记录	任何一方都没有：原文未规定此类记录或计数器	不适用	不适用	s0014

模式	结果	目标	ProVerif 所示	步骤
replay	candidate	R02	是：ProVerif 在 bounded.pv 上否证 `r02_b`	Bob 为该操作调用 Encapsulate(pk_A)，推导出 ss 和 ssk，并发送携带 transactionID、他的 senderNonce nb 和 kemContext 的 KemCiphertextInfo 消息（s0064, s0090, s0091）。/ Alice 解封装 ct，推导出 ssk，并发送一条受保护的 PKIMessage，携带 senderNonce na1 和 recipNonce nb（s0066..s0074, s0014）。/ Bob 检查 recipNonce = nb 以及 ssk 下的 MAC，并接受该消息（s0014, s0074）。/ Alice 发送同一操作的下一条受保护消息，携带新的 senderNonce na2，recipNonce 仍为 nb，ssk 相同（s0014, s0075）。/ 攻击者扣下那条下一条消息，改为重放第一条受保护消息。/ Bob 检查 recipNonce = nb（仍然匹配，因为他的 senderNonce 没有变化）以及 MAC（在复用的 ssk 下仍然验证通过），因此他第二次接受了较早的那条消息（s0014, s0074, s0075）。
repeat_one_time_action	candidate	A04	是：ProVerif 在 bounded.pv 上否证 `a04_inj`	Bob 发送携带 transactionID、senderNonce nb 和 kemContext 的 KemCiphertextInfo 消息，并推导出 ssk（s0064）。/ Alice 发送第一条受保护消息，携带 senderNonce na1、recipNonce nb，MAC 在 ssk 下；Bob 接受它（s0014, s0074）。/ 攻击者截获同一操作的后续受保护消息，改为重放第一条。/ Bob 再次接受同一条受保护消息，因此无论其 PKIBody 请求什么动作，都会在 Alice 的一次发送中被执行两次（s0074, s0075）。
desynchronisation	未发现			Bob 期望的 recipNonce 只在 Bob 发送消息时变化，而 ssk 在该操作中固定不变；在此流程中没有 Alice 可能错过的第二条 Bob 消息，且双方从相同的 ss 和 transactionID 推导 ssk（s0082），因此双方不会被带到不同的密钥上。model.pv 验证了 A08。
update_before_check	未发现			收到消息时不会改变任何保留值：ssk 和期望的 recipNonce 在任何受保护消息到达之前就已固定，且只进行比较（s0014, s0074）。
no_rollback_on_failure	未发现			解封装或 MAC 检查失败会终止该操作（s0068），且不会修改任何保留值，因此不会留下任何被改变的状态；被放弃操作的 ct 和 ssk 直接被丢弃（s0064）。
race_between_sessions	未发现			每个操作都有自己新的 ct、ss、ssk 和 transactionID（s0064, s0083, s0097），因此两个操作之间不共享任何值。model.pv 验证了两个操作只有在相同 transactionID 下才会共享 ssk（K04）。
distinguishable_failure	未发现			对于解封装失败，原文只规定了一个 failInfo 值 badMessageCheck（s0068），因此没有规定可区分的失败应答；模型中没有错误响应，也没有任何事件告诉攻击者哪项检查失败。
wrap_or_window	未发现			nonce 是 128 位随机值，只与 Bob 自己的值作相等比较（s0014）；它们不是计数器，也没有规定接受窗口，因此既不存在回绕，也不存在窗口。

分析者为目标命名的内容：

步骤	目标	特征	理由
20	A04, R02	changing_state	A04 和 R02 要求一条受保护消息不被接受两次，这取决于一个管理操作承载多少条消息，以及接收方在该操作中已经接受了什么。model.pv 对每个操作只有一次接受，省略了按操作记录的信息，因此两者都需要展开模型。
20	A03, A03.10, D01, D01.11, A06, A08, K01, K04, R01, U01, X01	plain_primitives	其余目标依赖 KEM、KDF 和 MAC 作为理想的符号原语（单向封装配以匹配的解封装、单向密钥推导、不可伪造的 MAC）。这些原语的代数定律均未被使用，但它们的计算性质也均未被证明。

分析者的结语

在 ProVerif 中分析了基于 KEM 的通用 CMP MAC 保护流程（RFC 9810 5.1.3.4、5.3.19.18、8.8、5.1.1）。首先构建了正常流程的模型（全部四个发送/接受事件均可到达）；随后全部 13 个目标都得到了答案，其中大多数是对任意数量会话的证明。

针对网络攻击者证明的内容：Bob 的接受意味着存在一次真实的 Alice 运行，其 transactionID、senderNonce、recipNonce、kemContext 和 body 相同（A03, A03.10）；跨操作的单射一致性（A04）；MAC 绑定每个头部字段和 body（D01, D01.11）；MAC 验证成功即为密钥确认（A06）；操作上下文是 Bob 自己的（A08）；nonce 和封装证据是新鲜的（R01）；共享密钥、MAC 密钥和 Alice 的 KEM 私钥保持机密（K01）；密钥在操作之间相互隔离（K04, U01）。所有证明都有被否证的对照查询。

一个新弱点：在一个管理操作内部，MAC 密钥被复用，且 Alice 的每条消息都携带相同的 recipNonce，因此一条被记录的受保护消息若作为同一操作的下一条消息重放，会同时通过两项检查并被第二次接受（A04, R02）。文稿声称 nonce 具有重放保护，却从未要求接收方记录已接受的 senderNonce。接收方侧的 nonce 记录可消除该攻击；此修改已在展开模型上检查，其余所有答案保持不变。

按原文复现、而非作为发现的内容：MAC 密钥不具备针对后续 KEM 私钥泄露的前向保密（新增目标 X01；正文 8.5、RFC 9629 第 7 节），以及 Alice 的 KEM 私钥持有者可恢复 ss 和 ssk（kem_sk_leak 下的 K01）。

限制：KEM、KDF 和 MAC 是理想的符号原语；重放结果依赖于将操作展开到两条消息并使用不透明的 PKIBody；仅涉及一对发送方-接收方和一个方向。Tamarin+ Agent 按原样对原语和消息类型建模，可判定某个具体 CMP 类型的重复消息是否具有有害影响。