软件学报 ISSN 1000-9825, CODEN RUXUEW
2025,36(10):4430−4443 [doi: 10.13328/j.cnki.jos.007393] [CSTR: 32375.14.jos.007393]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于 Kyber 公钥加密的高效认证密钥交换协议*

米瑞琪[1,2], 江浩东[3], 张振峰[1,2]

[1](中国科学院 软件研究所, 北京 100190)

[2](中国科学院大学, 北京 100049)

[3](河南省网络密码技术重点实验室, 河南 郑州 450001)

通信作者: 米瑞琪, E-mail: ruiqi2017@iscas.ac.cn

摘　要: Kyber 是一个基于格上困难问题的密钥封装机制, 2023 年被美国国家标准与技术研究院宣布为第 1 批标准化对象. Kyber-AKE 是 Kyber 的设计者基于 Kyber 构造的弱前向安全的认证密钥交换, 通过使用 3 个 IND-CCA 安全的密钥封装机制在两轮内协商会话密钥. 介绍 Kyber-PFS-AKE, 这是一种新的认证密钥交换协议. Kyber-PFS-AKE 只使用了 3 个 IND-CPA 安全的公钥加密, 并通过 FO 变换中的重加密技术处理 IND-CPA 安全公钥加密中的解密错误, 从而简化了后量子 Kyber-AKE 的设计. 严格证明 Kyber-AKE 协议中某些操作是冗余的, 去除这些冗余后, 协议变得更加简单和高效. 在 eCK-PFS-PSK 模型下证明 Kyber-PFS-AKE 的会话密钥不可区分性质, 以及完美的前向安全性等安全性质. 使用量子安全为 165-bit 的 Kyber-768, PKE 实现了 Kyber-PFS-AKE. 实验结果表明, Kyber-PFS-AKE 相比于 Kyber-AKE, 发起者计算时间降低了 38%, 响应者计算时间降低了 30%.

关键词: 公钥加密方案; Kyber; 认证密钥交换

中图法分类号: TP309

中文引用格式: 米瑞琪, 江浩东, 张振峰. 基于Kyber公钥加密的高效认证密钥交换协议. 软件学报, 2025, 36(10): 4430–4443. http://www.jos.org.cn/1000-9825/7393.htm

英文引用格式: Mi RQ, Jiang HD, Zhang ZF. Efficient Authenticated Key Exchange Protocol Based on Kyber Public-key Encryption. Ruan Jian Xue Bao/Journal of Software, 2025, 36(10): 4430–4443 (in Chinese). http://www.jos.org.cn/1000-9825/7393.htm

Efficient Authenticated Key Exchange Protocol Based on Kyber Public-key Encryption

MI Rui-Qi[1,2], JIANG Hao-Dong[3], ZHANG Zhen-Feng[1,2]

[1](Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

[2](University of Chinese Academy of Sciences, Beijing 100049, China)

[3](Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450001, China)

Abstract: Kyber, a key encapsulation mechanism based on lattice problems, was the first to be standardized by the National Institute of Standards and Technology (NIST) in 2023. Kyber-AKE, a weak forward-secure authenticated key exchange (AKE) protocol, was constructed by the designers of Kyber and derives session keys in two rounds using three IND-CCA secure key encapsulation mechanisms. This study introduces Kyber-PFS-AKE, a newly proposed authenticated key exchange protocol. In Kyber-PFS-AKE, only IND-CPA secure public-key encryption is utilized, and decryption errors within IND-CPA secure encryption are addressed using the re-encryption technique within the FO transformation, thus simplifying the design of post-quantum Kyber-AKE. A rigorous proof demonstrates that certain operations in the Kyber-AKE protocol are redundant. By eliminating these redundancies, the protocol achieves a simpler and more efficient design. The session key indistinguishability and perfect forward security of Kyber-PFS-AKE are formally proven within the eCK-PFS-PSK model. The proposed Kyber-PFS-AKE is implemented using Kyber-768. PKE with 165-bit quantum security. Experimental results show

* 基金项目: 国家重点研发计划 (2021YFB3100100)
本文由“抗量子密码与区块链应用”专题特约编辑翁健教授、祝烈煌教授、赵运磊教授推荐.
收稿时间: 2024-07-01; 修改时间: 2024-09-05; 采用时间: 2024-12-30; jos 在线出版时间: 2025-01-20
CNKI 网络首发时间: 2025-06-11

that compared to Kyber-AKE, the computation time for the initiator is reduced by 38%, while the computation time for the responder is reduced by 30%.

Key words: public-key encryption (PKE) schema; Kyber; authenticated key exchange (AKE)

认证密钥交换 (authenticated key exchange, AKE) 是网络安全系统中最重要的密码学模块之一. 大多数 AKE 协议依赖 Diffie-Hellman (DH) 密钥交换建立会话密钥, 通过数字签名、长期 Diffie-Hellman 密钥交换或公钥加密进行身份认证. AKE 拥有非常广泛的应用, 如传输层安全协议 (transport layer security, TLS), 虚拟专用网络 (virtual private network, VPN) 协议等. 然而随着量子计算机的发展, 拥有量子计算机的攻击者 (以下简称为量子攻击者) 可以在短时间内攻破 DH 密钥交换并获取会话密钥, 换句话说, DH 密钥交换不具备抵抗量子攻击者的能力. NIST 呼吁使用抗量子的公钥加密 (以下简称 PKE)/密钥封装机制 (以下简称 KEM), 以替代目前的密钥交换标准.

对于抗量子的认证密钥交换协议, 目前存在的主要的设计思路是: 1) 通过抗量子的数字签名 (如基于格上困难问题、基于编码上的困难问题的数字签名) 完成身份认证, 并通过抗量子的公钥加密建立会话密钥; 2) 仅使用抗量子的公钥加密完成身份认证并建立会话密钥. 由于后量子的数字签名方案通常具有较长的公钥和签名, 通过公钥和密文长度较小的公钥加密构造认证密钥交换计算效率更高, 传输的代价更小.

2016 年, 美国国家标准与技术研究院 (NIST) 面向全球征集抗量子密码算法, 包括公钥加密和数字签名算法, 旨在通过建立抗量子密码标准, 以应对量子计算机带来的安全威胁[1], Kyber[2]是目前唯一被 NIST 选作密钥封装机制标准的算法[3]. Kyber 基于格上的 MLWE 安全性假设, 拥有较小的公钥和密文, 在大多数应用中展现出了卓越的性能, 其性能与安全性之间的调整也相对简单. NIST 推荐在绝大多数应用场景下优先使用 Kyber. 是否对基于其他安全性假设的方案 (如, 基于编码、多变量困难问题) 进行标准化仍然在讨论中. 如基于编码的 Classic McEliece 密钥封装机制目前存在很多安全问题[4], 因此, 设计基于 Kyber 的认证密钥交换是较为合理的选择.

Kyber 提出了其认证密钥交换方案 Kyber-AKE[2], 其基本思想是发起者与响应者各自拥有长期的 IND-CCA 安全的密钥封装机制的公私钥对用于身份认证. 在握手时, 发起者使用 IND-CCA 安全的密钥封装机制生成临时公私钥对 (epk_i, esk_i), 并用响应者的长期公钥封装出密文和共享秘密 (c_1, k_1). 响应者使用收到的 epk_i 和发起者的长期公钥分别封装出密文和共享秘密作为密钥材料. 双方各自用持有的私钥将收到的密文解密, 并将这些密钥材料输入密钥导出函数, 导出会话密钥. Kyber-AKE 来自 Fujioka 等人[5]提出的通用构造, 在该构造中, 用于生成临时公私钥对 (epk_i, esk_i) 的可以是 IND-CPA 的 KEM. Kyber-AKE 握手过程如图 1 所示, 发起者与响应者均拥有长期公私钥对. 发起者公私钥对为 (spk_i, ssk_i), 响应者公私钥对为 (spk_r, ssk_r).

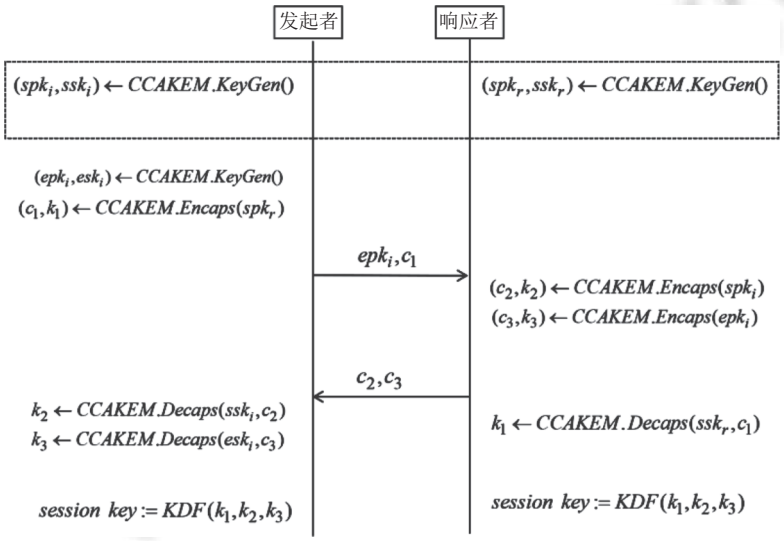

图 1 Kyber-AKE 协议握手过程

Kyber-AKE 会话密钥的安全性可以在 Canetti-Krawczyk (CK) 模型下证明[6], 但是只能保证会话密钥的弱前向安全性, 研究表明, 对于两轮的密钥交换, 难以达到完美前向安全性[6]. 前向安全性指的是即使会话参与方的长期密钥遭到泄露, 也不会影响到泄露前生成的会话密钥的安全性. 弱前向安全性要求敌手不能主动参与会话 (完美前向安全性游戏中敌手可以主动参与会话). 此外, Kyber-AKE 全部使用 IND-CCA 安全的密钥封装机制, 可能存在进一步提升其计算性能的空间. 这是由于 Kyber-AKE 的会话密钥导出过程中, 对已经经过杂凑运算的会话密钥进行了进一步杂凑运算, 该设计可能会导致冗余运算, 因此, 通过 IND-CPA 安全的 PKE 设计 AKE 可能会更加简洁. 采用 IND-CPA 安全的 PKE 设计 3 轮 AKE 需要克服的主要难点在于: (1) IND-CPA 安全的 PKE 中存在一定的解密错误概率, 需要解决由于解密错误导致的安全性问题. (2) 如何保证协议具有较强的安全性质, 如: 完美的前向安全性, Kyber-AKE 并不具备该安全性质. (3) 如何让协议有较好的计算性能.

针对上述问题, 本文从 IND-CPA 安全的公钥加密 (PKE) 出发, 提出了一个安全、简洁且计算性能较高的后量子 AKE, 并且在 eCK-PFS-PSK 模型[7]下给出了该协议的安全性证明. 由于 eCK-PFS-PSK 模型保证了当双方共享一个没有被敌手窃取的预共享密钥, 或者至少有一方的长期或临时私钥没有被敌手窃取, 则会话密钥保持伪随机性, 这意味着会话密钥拥有完美前向保密 (perfect forward secrecy, PFS) 性质. 该协议被命名为 Kyber-PFS-AKE, 并具有更高的计算效率. Kyber-PFS-AKE 构造的主要思想如下.

(1) 从 IND-CPA 安全的 PKE 出发构造认证密钥交换: 认证密钥交换的会话密钥都是由一系列密钥材料 (如图 2 中 $C_1,C_2,C_3,\ldots$) 按顺序链式产生. 以 Kyber-AKE 为例, 双方会按照协议流程产生 KEM 临时公钥、KEM 共享密钥或 KEM 的密文等, 这些被统称为密钥材料 (keying material), 用于生成双方的会话密钥. 在握手过程中, 发起者 (响应者) 将协议过程中每一步生成的密钥材料连同上一步密钥导出函数 (key derivation function, *KDF*) 的输出共同作为密钥导出函数的输入, 最终握手完成后, 密钥导出函数将用于生成传输消息的会话密钥.

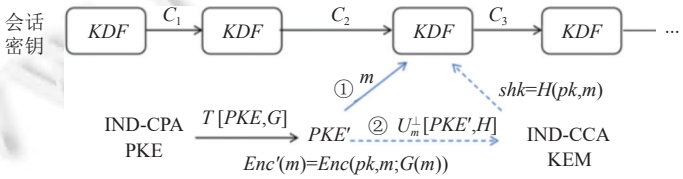

图 2 会话密钥的派生过程

NIST 征集的 IND-CCA 安全的 KEM 都是对 IND-CPA 安全的 PKE 进行 Fujisaki-Okamoto 变换得到 (以下简称 FO 变换). FO 变换由两个变换组成, 分别称为 U 变换和 T 变换, 记为 $FO=U\circ T$. G 和 H 为密钥导出函数, 给定公钥加密方案 PKE. T 变换对 PKE 进行去随机化和重加密校验, 记作 $PKE'=T[PKE,G]$. 具体来说, PKE' 的加密算法定义为 $PKE'.Enc'(pk,m):=Enc(pk,m;G(m))$. PKE' 的解密 $PKE'.Dec'(sk,ct)$ 定义为: 首先通过 $PKE.Dec(sk,ct)$ 将密文 ct 解密为 m', 并对 m' 进行重加密校验, 若 $Enc(pk,m';G(m'))\neq ct$, 则拒绝该密文. U 变换记作 $KEM=U[PKE',H]$, 其封装算法 $KEM.Encaps(pk)$ 首先从消息空间均匀随机选择 m, 计算密文 $ct\leftarrow PKE'.Enc'(pk,m)$, 并计算共享密钥 $shk\leftarrow H(pk,m)$. 解封装算法首先进行解密, 计算 $m:=Dec(sk,ct)$, s 为私钥中保存的随机种子. 解封装过程为: 如果 $m\neq\perp$, 则 $KEM.Decaps(sk,c)=H(pk,m)$, 否则 $KEM.Decaps(sk,c)=H(s,ct)$.

Kyber-AKE 直接将 IND-CCA 安全的密钥封装机制 (KEM) 的会话密钥输入链式密钥中, 链式会话密钥在这一步骤中形如 $KDF(H(pk,m))$. 会话密钥的生成过程如图 2 中的②③所示. 实际上, 链式密钥派生过程中的密钥派生函数 (*KDF*) 可以扮演与 U 变换中的密钥派生函数相同的作用. 因此, Kyber-PFS-AKE 的设计中, 从一个 IND-CPA 安全的 PKE 出发, 对其进行 T 变换, 随后用链式密钥计算中的密钥派生函数代替 U 变换, 这样做的好处是简化了协议设计, 提高了效率并减小了代码规模. Kyber-PFS-AKE 生成会话密钥的过程如图 2 中的①所示.

(2) 去除对完整公钥的杂凑运算: 为了进一步提升协议的计算性能, 在会话密钥导出过程中, 发起者与响应者均需要使用对方的公钥计算 IND-CCA 安全的 KEM 的共享密钥, 并将共享密钥作为密钥派生函数的输入, 计算公钥 pk 的杂凑值是为了提供多用户安全性, 即当存在多个拥有不同公钥接收者的情况下, 敌手可能将同一个消息 m

用不同公钥加密. 研究表明, 在特定的 RSA 参数设定下, 敌手可以解密出该消息[8]. 为了抵抗这种攻击, Kyber 在设计中添加对完整公钥的杂凑运算. 然而, 对整个公钥进行密钥导出非常影响性能, 基于格的密钥封装机制 (如: Kyber 和 Saber[9]) 的公钥长达 1–3 KB, 对完整的公钥做导出会占据较大的计算资源, 特别是当连接请求较多的时候. 为了提高计算效率, 我们移除了将双方的整个公钥包含到最终会话密钥计算中的步骤. 对 Kyber-768 来说, 对整个公钥进行杂凑计算的时间占据 Kyber-768 的密钥封装的总时间的 50% 以上. 根据 Duman 等人[10]的工作, 将公钥的不可预测部分 (例如前 32 个字节) 包含到最终会话密钥计算中已足以达到与包含整个公钥相同的安全目标.

总结前方的所有构造思想, 本文提出的 Kyber-PFS-AKE 构造如图 3 所示, 发起者与响应者均拥有长期公私钥对. 发起者公私钥对为 (spk_i, ssk_i), 响应者公私钥对为 (spk_r, ssk_r). 发起者与响应者可以协商预共享密钥 psk. 需要注意的是, 下面的构造并不是完整的 Kyber-PFS-AKE 协议. 为了实现更强的安全性质 (如: 完美的前向安全性), 需要添加密钥确认消息 (*conf* 消息). 完整的 Kyber-PFS-AKE 协议详见第 3 节.

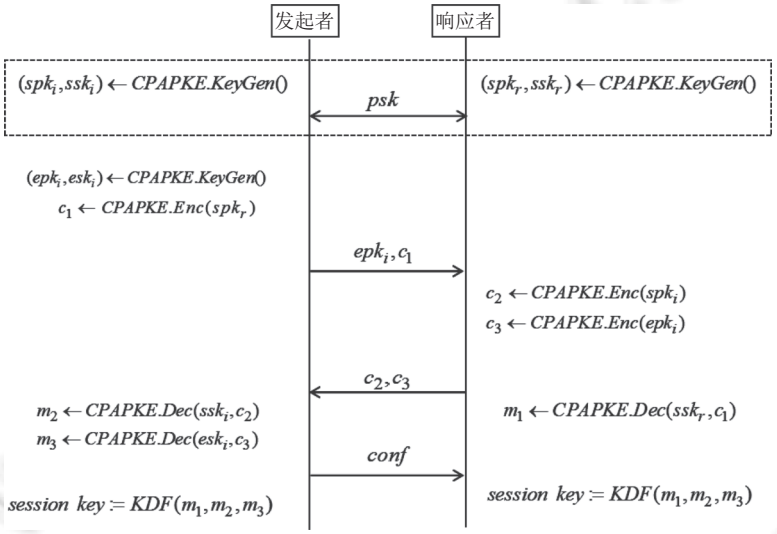

图 3 Kyber-PFS-AKE 协议设计简化示意图

● Kyber-PFS-AKE 的安全性质与安全证明: Kyber-PFS-AKE 能够达到的安全性质为: 1) 完美前向安全性: 如果双方各自至少有一个私钥没有被敌手窃取, 或者双方共享一个没有被敌手窃取的预共享密钥, 则会话密钥保持伪随机性. 这也意味着会话密钥拥有完美前向安全性. 前向安全性通常指的是在某个时间点敌手窃取了用户的私钥, 保证在该时间点前协商的会话密钥仍然保持伪随机性; 2) 会话密钥唯一性: 构造出的会话密钥以极大的概率不会重复出现, 成立的前提是敌手不主动更改用户公私钥对; 3) 实体认证 (entity authentication): Kyber-PFS-AKE 提供实体认证, 但是需要除去两种非平凡的情况. 如果敌手拿到了 Alice 的长期私钥以及 Alice 和 Bob 之间所有预共享的密钥, 或者反过来敌手拿到了 Bob 的长期私钥以及 Alice 和 Bob 之间所有预共享的密钥, 那么认证就无法实现. 此外, Kyber-PFS-AKE 还可以抵抗 KCI 攻击、MEX 攻击、UKS 攻击等. KCI 攻击指的是敌手拿到 Bob 的全部长期私钥后, 尝试在 Bob 面前扮演 Alice.

我们在 eCK-PFS-PSK 模型[11]下给出了 Kyber-PFS-AKE 的安全证明. 证明中的一个技术难点是, 如何处理 IND-CPA 安全的公钥加密中出现的解密错误 (尽管解密错误的概率很低). 由于我们保持了转换 $T[PKE,G]$ (去随机化和重加密). $U[PKE',H]$ (哈希) 的角色由密钥派生过程中的密钥派生函数 *KDF* 扮演. 因此, 通过将 *KDF* 建模为随机预言机, 我们仍然可以通过类似 FO 变换中构造解密预言机的证明技术, 将挑战者返回的值嵌入游戏中.

● 计算性能: 我们的构造在计算性能上相比 Kyber-AKE 有明显的提升: 发起方时间计算时间减少了 38%, 响应者计算时间减少了 30%.

● 与通用 AKE 构造[7]的对比: 通用的 3 轮 AKE 构造基于 DH 密钥交换, DH 运算天然具有对称的特性, 发起

者和响应者各自拥有长期的 DH 公私钥对. 在握手过程中, 双方将各自的公钥和对方的长期私钥结合, 如果双方能够计算出相同的 DH 值则可以认证对方的身份, 同时, 双方在握手中各自生成临时的 DH 公私钥对, 并且将各自的临时 DH 公钥发送给对方, 将对方的临时 DH 公钥与自己的临时 DH 私钥结合, 直观上说, 如果双方导出相同的 DH 值, 则可以保证协议的前向安全性.

通过公钥加密和密钥封装构造 AKE 与通过 DH 密钥交换构造 AKE 具有较大区别, 由于公钥加密并不具备和 DH 运算类似的对称性质, 因此在身份认证和会话密钥的导出上存在较大的区别. 具体来说, 发起者和响应者各自有用于认证身份的长期公私钥对. 在握手过程中, 发起者使用响应者的公钥进行加密, 并发送给响应者. 如果响应者可以解密该密文, 代表响应者拥有相应的私钥, 从而向发起者认证自己的身份. 响应者使用发起者的公钥进行加密, 并发送给发起者, 如果发起者可以解密该密文, 代表发起者拥有相应的私钥, 从而向响应者认证自己的身份. 此外发起者在握手过程中会生成临时公私钥对, 并将临时公钥发送给响应者, 响应者使用该临时公钥加密后将密文发送给发起者, 如果发起者能够解密该密文可以保证了握手的前向安全性. 因此, Kyber-AKE 将这三者各自的共享密钥 (其计算过程为 $KDF(pk,m)$, 其中 pk 为公钥, m 为明文) 进行密钥导出 (即对共享密钥进行 KDF 运算), 作为双方的会话密钥. 同时, Kyber-AKE 是一个两轮协议, 无法达到完美的前向安全性. 而本文的 Kyber-PFS-AKE 的构造中, 添加了密钥确认消息, 取消了共享密钥的导出过程, 将明文 m 和部分公钥直接进行密钥导出, 对使用长期密钥的解密运算, 通过重加密技术解决了 IND-CPA 安全的 PKE 的解密错误, 在保证完美前向安全性的同时, 使得协议拥有更好的计算性能.

在考虑到网络延迟的情况下, Kyber-PFS-AKE 的运行时间通常会高于两轮的 Kyber-AKE. 根据文献 [12] 中的研究, 广域网上的网络延迟通常要超过 20 ms, 甚至高达数百毫秒, 而后量子的公钥加密 (PKE/KEM), 以及杂凑、AEAD 的运行时间通常不会超过 10 ms, 因此 3 轮的 AKE 协议在传输时间上会显著慢于两轮的 AKE. 然而, Kyber-PFS-AKE 能够达到完美的前向安全性, 因此这是一个取舍的问题, 在不需要达到完美前向安全性的场景下 Kyber-AKE 具有性能上的优势, 在需要前向安全性的场合, Kyber-PFS-AKE 显然有更强的安全性质. 因此需要根据安全性和性能需求选择合适的协议.

PQ-WireGuard[11]可以看成 WireGuard 的后量子版本, 基于 IND-CCA 安全的 Classic McEliece 密钥封装机制和 IND-CPA 安全的密钥封装机制 Saber 构造的认证密钥交换. 作者同样在 eCK-PFS-PSK 模型下给出了安全性证明. 在设计上, Kyber-PFS-AKE 与之有很多相似之处. 然而, 两者在困难问题、计算效率和使用场景上差别较大. 具体来说: (1) PQ-WireGuard 基于 IND-CCA 安全的 Classic McEliece 和 Saber, Classic-McEliece 是一个基于编码困难问题的密钥封装机制, 该方案是否成为 NIST 标准目前仍在第 4 轮讨论中; 此外, PQ-WireGuard 选用 IND-CPA 版本的 Saber, Saber 已经不被 NIST 考虑成为标准. 另外, PQ-WireGuard 使用的 Saber 参数并非是 NIST 提案中的参数, 而是作者自己选择的新参数 (被命名为 Dagger), 其解密错误率被升高到 2^{-25}, 是否能够使用 Dagger 仍然需要讨论. (2) 关于计算性能: Classic McEliece 密钥封装机制在封装运算和解封装运算中均显著慢于 Kyber, 以 NIST 规定的安全级别 3 为例, Classic McEliece 的封装时间与解封装时间分别为 Kyber-768 的加解密时间的 2.5 倍和 6 倍, Saber 的封装时间与解封装时间与 Kyber-768 类似. 因此在计算效率方面, Kyber-PFS-AKE 显然更具有计算性能上的优势. (3) 使用场景: PQ-WireGuard 是一个基于 UDP 协议的 VPN (虚拟专用网络), 双方无需传输证书, 此外, PQ-WireGuard 的数据可以在一个 IP 包内传输完成而无需分片, 而 Kyber-PFS-AKE 需要 IP 分两片完成传输, 此时 PQ-WireGuard 比 Kyber-PFS-AKE 更有优势. 但是 Classic-McEliece 的公钥长度为 524 160 字节, 在很多需要传输证书的场景中 (如: 国密 SSL 协议), 如此长的公钥传输是需要耗费相当长的时间的. 而 Kyber 公钥与密文长度仅为 1 184 字节, 在这种场景下, 选择 Kyber-PFS-AKE 是更有优势的.

本文第 1 节介绍后量子认证密钥交换构造相关工作与研究现状. 第 2 节介绍本文所需的基础知识. 第 3 节介绍 Kyber-PFS-AKE 协议构造. 第 4 节给出协议在 eCK-PFS-PSK 模型下的安全性证明 (内容梗概, 完整证明发布在公开平台 CSDN 上: https://blog.csdn.net/weixin_38238086). 第 5 节通过对比实验验证我们的协议对比 Kyber-AKE 有更高的效率. 最后总结全文. 在完整的安全性证明中, 我们给出协议在 eCK-PFS-PSK 模型下的完整安全性证明, 包括完整的安全模型、敌手能力、证明过程等.

1 后量子认证密钥交换构造相关工作与研究现状

抗量子 AKE 有两种主要的设计思路.

第 1 种设计思路是通过抗量子的数字签名进行身份认证, 并通过抗量子的公钥加密 (KEM/PKE) 替代 DH 密钥交换建立会话密钥. 这种思路的代表性工作是 Paquin 等人[12]在 TLS 中进行的实验, 该实验中测试了全部 NIST 后量子算法征集中的签名算法与公钥加密组合的表现, Sosnowski 等人[13]做了类似的工作, 他们发现在同一证书链中混合使用不同的签名可以提升握手性能.

第 2 种 AKE 设计思路是仅使用抗量子的公钥加密 (PKE/KEM) 完成身份认证并建立会话密钥. 由于后量子的数字签名方案通常具有较长的公钥和签名长度, 而公钥加密或者密钥封装机制的公钥和密文长度通常较小, 因此构造的认证密钥交换计算效率更高, 且传输的代价更小. 第 2 种设计思路可以更加细分为: 1) 使用密钥封装机制 (KEM) 构造 AKE; 2) 通过被动安全的公钥加密方案 (PKE) 构造 AKE. 使用密钥封装机制 (KEM) 构造 AKE 的代表性工作为 Schwabe 等人[14]提出的 KEMTLS, 该工作是在 TLS 1.3 协议的基础上, 通过 IND-CCA 安全的 KEM 构造的后量子 TLS 协议, 认证通过密钥封装机制完成, 其基本思想是, 能够成功解密对方发来的密文即可证明自己是私钥的持有者. 作者测试了 NIST 算法征集中的所有 KEM 在 KEMTLS 的表现. 此外, 作者还给出了 KEMTLS 的安全性证明. 另一个代表性工作是 Hülsing 等人[11]提出的 PQ-WireGuard 协议, 这是一个虚拟专用网络 (VPN) 协议, 作为 WireGuard[15]协议的后量子版本. PQ-WireGuard 使用了 IND-CCA 安全的 Classic McEliece 替代使用长期 DH 密钥交换用于身份认证, 使用了 Saber 变体[9]的 IND-CPA 安全的 Dagger KEM 替代临时 DH 密钥交换. PQ-WireGuard 实现了 WireGuard 的所有安全属性, 例如享受单程密钥交换的好处, 固定的密码套件, 并在性能上优于 IPsec 和 OpenVPN. PQ-WireGuard 比 WireGuard 慢不到 60%. 此外, 李子臣等人[16]提出了基于环上的 LWE 假设 (RLWE) 的认证密钥交换, 并在 eCK 模型下给出了安全性证明. 这是一个两轮的密钥交换, 根据文献 [6] 分析, 两轮的 AKE 无法做到完美的前向安全性.

对于通过 PKE 构建 AKE 的研究, 代表性工作包括 Xue 等人[17]提出的 AKE, 该工作从所谓的“2-key” KEM 出发, 提供了一个通用的构造方法. 直接从弱安全 (IND-CPA) 的 PKE 构造 AKE, 因此可以去除一些冗余操作 (例如哈希函数). 另一项具有类似思路的工作是由 Hövelmanns 等人[18]提出的一种通用的方法, 主要技术是将一些 FO 变换的机制移入 AKE, 这是一个两轮的密钥交换. 作者并未对这种通用框架进行实例化, 此外, 该构造的安全性证明使用的模型要弱于 eCK-PFS-PSK 模型, 无法达到完美的前向安全性.

2 基础知识

Kyber-PFS-AKE 主要基于公钥加密 (PKE), 下面就相关概念和基本安全性定义予以介绍.

给定 $n \in \mathbb{N}$, $[n]$ 表示 $\{1,2,\ldots,n\}$. 对于有限集 S, $|S|$ 表示其基数. $x \xleftarrow{\$} S$ 表示从 S 中均匀随机选择元素 x. $x \xleftarrow{\$} D$ 表示从分布 D 中随机采样元素 x. $\stackrel{?}{=}$ 为布尔表达式, 如果等式成立则值为 1, 否则为 0. 对于一个确定性算法 $\mathcal{A}$, 其输入为 x, 输出为 y, 记作 $y \leftarrow \mathcal{A}(x)$. 对于一个概率算法 $\mathcal{A}$, 其输入为 x, 输出为 y, 记作 $y \xleftarrow{\$} \mathcal{A}(x)$. 如果敌手 $\mathcal{A}$ 可以访问预言机 $\mathcal{O}$, 记作 $\mathcal{A}^{\mathcal{O}}$. 如没有特殊说明, 算法均默认为概率算法.

定义 1. 公钥加密方案. 密钥空间为 $\mathcal{K}$ 的公钥加密方案 $PKE = (KeyGen, Enc, Dec)$ 由以下算法组成: (1) 密钥生成算法 $KeyGen$: 给定安全参数 1^n 和公共参数 pp, 概率算法 $KeyGen$ 输出一对公私钥 (pk, sk). 密钥生成过程表示为 $(pk, sk) \leftarrow KeyGen(1^n; pp)$. (2) 公钥加密算法 Enc: Enc 输入公共参数 pp、公钥 pk 以及待加密明文 m, 输出其加密结果 ct. 公钥加密过程表示为 $ct \leftarrow Enc(pp; pk, m)$. (3) 解密算法 Dec: 确定性算法 Dec 输入私钥 sk 和密文 ct, 输出解密结果 m'. 解密过程表示为 $m' \leftarrow Dec(sk, ct)$.

● 公钥加密方案的正确性. 解密过程存在一定的错误概率, 如果对于明文空间中任意 $m \in \mathcal{M}$, 满足 $\Pr[m = Dec(sk, ct)] \geqslant 1 - \delta$, 其中 $(pk, sk) \leftarrow KeyGen(1^n; pp)$, $ct \leftarrow Enc(pp; pk, m)$, 概率来自于 $KeyGen$ 和 Enc 中的随机采样, 则称公钥加密方案是 δ-正确的.

定义 2. 公钥加密方案在选择明文攻击下的不可区分安全性 (IND-CPA 安全性). 公钥加密方案 PKE 满足选择明文不可区分安全敌手 $\mathcal{A}$ 和挑战者之间进行如算法 1 所示游戏, 敌手的获胜优势为:

$$Adv^{\text{IND-CPA}}_{\text{PKE},\mathcal{A}} = \left|\Pr[G^{\text{IND-CPA}}_{\text{PKE},\mathcal{A}} \Rightarrow 1] - \frac{1}{2}\right| \leqslant negl(n).$$

算法 1. $G^{\text{IND-CPA}}_{\text{PKE},\mathcal{A}}$.

输入: 安全参数 λ;
输出: 敌手是否取得成功 (即 $b' \stackrel{?}{=} b$), 成功则输出 0, 失败则输出 1.

1. $(pk, sk) \leftarrow Gen(1^{\lambda})$
2. $b \stackrel{\$}{\leftarrow} \{0,1\}$
3. $(m_0, m_1) \leftarrow \mathcal{A}^{Enc}(pk)$
4. $c^* \leftarrow Enc(pk, m_b)$
5. $b' \leftarrow \mathcal{A}^{Enc}(pk, c^*)$
6. **return** $b' \stackrel{?}{=} b$

本文中, 密钥导出函数 KDF 用于导出链式会话密钥. KDF 具有两个输入, 我们用 $KDF(X,Y)=Z$ 表示密钥导出函数, 其中输入为 X, Y, 输出为 Z, Z 中包含 3 个块 $Z=Z_1||Z_2||Z_3$, $KDF(X,Y)$ 输出的第 i 个分块记作 $KDF_i(X,Y)$. 每个分块长度为 32 字节.

定义 3. 带关联数据的认证加密 (authenticated encryption with associated data, AEAD)[7]. $AEAD=\{KeyGen, Enc, Dec\}$, 其安全参数为 λ, 定义空间 $\mathcal{K}$, $\mathcal{N} \in \{0,1\}^l$, $\mathcal{M} \in \{0,1\}^*$, $\mathcal{H} \in \{0,1\}^*$: (1) 密钥生成算法 $KeyGen$ 的输入为 1^{λ}, 输出 $k \in \mathcal{K}$, 该过程记作 $k \leftarrow KeyGen(1^{\lambda})$. (2) 加密算法 Enc 的输入为 $k \in \mathcal{K}$, 一次性随机数 $N \in \mathcal{N}$, 头 $H \in \mathcal{H}$, 消息 $M \in \mathcal{M}$, 输出为密文 $C \in \{0,1\}^*$, 该过程记作 $C := Enc(k,N,H,M)$. (3) 解密算法 Dec 的输入为密钥 $k \in \mathcal{K}$, 一次性随机数 $N \in \mathcal{N}$, 头 $H \in \mathcal{H}$, 密文 C, 若解密成功, 则返回 M', 若解密失败则返回 $\perp$.

对于概率多项式时间的敌手 $\mathcal{A}$, 其输入为安全参数 λ, $\mathcal{A}$ 可以访问加密预言机 $Enc(\cdot,\cdot,\cdot)$, 加密预言机的输入为 (N,H,M), 预言机会随机选择 $k \in \mathcal{K}$, 输出 $Enc(k,N,H,M)$. 如果敌手能够输出 (N,H,C), 满足 $M := Dec(k,N,H,C) \neq \perp$ 且 (N,H,M) 从未询问过加密预言机 $Enc(\cdot,\cdot,\cdot)$, 则称敌手 $\mathcal{A}$ 伪造密文成功. 如果敌手伪造密文成功的概率 $Adv^{\text{aead-auth}}_{\text{AEAD},\mathcal{A}}(\lambda)$ 可以忽略, 我们称 AEAD 为 aead-auth 安全的.

3 Kyber-PFS-AKE 协议设计

Kyber-AKE 握手过程的 3 次封装和解封装计算分别起到不同的作用. 发起者和响应者各自拥有用于认证身份的长期公私钥对, 分别记作 (spk_i, ssk_i) 和 (spk_r, ssk_r), 响应者有能力解密发起者使用 spk_r 加密的密文, 代表响应者可以向发起者认证自己的身份. 发起者有能力解密响应者使用 spk_i 加密的密文, 代表发起者可以向响应者认证自己的身份. 发起者能够解密响应者通过临时公钥加密的密文可以保证了握手的前向安全性. 因此, Kyber-AKE 将这三者各自的共享密钥进行密钥导出, 作为双方的会话密钥.

然而, 这种设计中存在无法抵御的攻击. 如: (1) MEX (maximal exposure) 攻击[5]. 在 MEX 攻击中, 敌手可以获得握手过程中生成的所有临时密钥和随机数. 这类攻击通常发生在运行环境或使用的随机数生成器被破坏的情况下. 此外, eCK-PFS-PSK 模型允许敌手获取发起者或者响应者的长期私钥. 在基于 DH 设计的 AKE 中, 可以通过双方的长期私钥进行 DH 密钥交换处理. 但是在使用公钥加密的情况下, 无法将双方的长期私钥通过公钥加密或者密钥封装结合在一起. 因此一种可能的处理方式[11]是双方各自保持长期的随机数 σ 和短期的 (定期更新的) 随机数 r, 即: 发起者持有长期随机数 σ_i 和短期随机数 r_i, 响应者持有长期随机数 σ_r 和短期随机数 r_r. 发起者通过密钥派生函数计算 $KDF(\sigma_i, r_i)$ 作为其 PKE 中被加密的对象 m_1, 响应者通过密钥派生函数计算 $KDF(\sigma_r, r_r)$ 作为其 PKE 中被加密的对象 m_3. 通过这种方式将双方各自的长期/临时随机数放入会话密钥中, 以抵抗 MEX 攻击.

(2) UKS (unknown key share) 攻击, 一种经典的攻击模式为中间人攻击. 抵抗 UKS 攻击的典型措施是将双方的公钥做杂凑运算, 整合到最终会话密钥. (3) 如果能够在第 1 条消息中使用双方的长期 DH 密钥进行 DH 运算, 通过第 1 条消息即可对发起方进行身份验证, 从而能简单缓解 DoS (拒绝服务) 攻击, 因为响应方一旦接收到发起方的第 1 条消息就可以检测到消息来源非法并中止握手. 因此, Kyber-PFS-AKE 的设计中, 引入了发起方和响应方之间的预共享密钥 psk, 可以将 $H(spk_i \oplus spk_r)$ 作为默认值, 其中 H 为密钥导出函数, 并将 psk 哈希到最终会话密钥. 这使得会话密钥计算中涉及的两方的静态公钥, 这种方法最早出现在 PQ-WireGuard 中, 能够在一定程度上抵御 UKS 攻击.

Kyber-AKE 中使用的 KEM 均为 IND-CCA 安全的 KEM. IND-CCA 安全的 KEM 是对 IND-CPA 安全的 PKE 进行 FO 变换得到. FO 变换的简要介绍见引言部分第 8 自然段. 根据 FO 变换的设计特性, 可以将 IND-CCA 安全的 KEM 共享密钥 shk 的链接密钥的密钥导出过程表示为 $C_3, \kappa_3 = KDF(C_2, H(spk_r, m_1))$ ($C_8 = KDF(C_7, H(spk_i, m_3))$).

Kyber-PFS-AKE 的构造保持了 T 变换. 从本质上讲, 链式密钥中的密钥派生函数 $C_3, \kappa_3 = KDF(C_2, *)$ ($C_8 = KDF(C_7, *)$) 可以扮演与密钥导出函数 H 在 U 变换 (“hashing”) 相同的角色. 因此, 我们保留变换 T, C_3, κ_3 和 C_8 的派生形式为 $C_3, \kappa_3 = KDF(C_2, spk_r, m_1)$ ($C_8 = KDF(C_7, spk_i, m_2)$), 让链式密钥中的 $KDF(C_2, *)$ 发挥同样的作用.

此外, T 变换中存在额外的耗时因素. 第 1 个额外耗时因素是 T 变换将整个公钥都作为密钥派生函数的输入, 这是不必要的浪费. 事实上, 将公钥的不可预测部分 (例如前 32 字节) 包含在最终会话密钥计算中足以实现与包括整个公钥在内的会话密钥计算提供的相同安全性[10]. 因此, C_3, κ_3 的计算形式为 $C_3, \kappa_3 = KDF(C_2, \overline{spk_r}, m_1)$, $C_8 = KDF(C_7, \overline{spk_i}, m_2)$, 其中 $\overline{spk_i}$、$\overline{spk_r}$ 表示密钥的一个小的 (例如 32 字节) 不可预测部分. 基于格的 PKE/KEMs 中的公钥大小 (Kyber 和 Saber 中约为 1 KB) 明显大于 32 字节, 这会对速度产生显著影响. 减少哈希函数的输入长度在 Kyber 的密钥生成和封装过程中可以加快 2–3 倍的速度, 并且在 Kyber、Saber 中可以将封装过程提速高达 50% 以上. 因此, 我们只将公钥的一个小的不可预测部分 (例如 32 字节) 包含在最终会话密钥计算中, 而不是包含整个公钥.

Kyber-PFS-AKE 握手协议如图 4 所示, 会话密钥导出的过程 (链式密钥 $C_1, C_2, \ldots, C_{10}$) 在图 5 中给出, 包括种子值 (seed, C_k)、AEAD 密钥 (κ_k) 和杂凑值计算. 对于 分为行的 k 值, 第 1 行表示发起者的计算, 第 2 行表示响应者相应的计算, $\overline{spk_i}$, $\overline{spk_r}$ 表示密钥的小部分 (例如, 32 字节) 的不可预测部分. 图 4 中 $CPAPKE = (KeyGen, Enc, Dec)$ 是一个 IND-CPA 安全的公钥加密方案, KDF 是一个密钥派生函数, 而 MAC 是消息认证码. 发起者和响应者的最终会话密钥分别是 tk_i、tk_r. 我们将 IND-CPA 安全的 PKE 记作 $CPAPKE = (CPAPKE.Gen, CPAPKE.Enc, CPAPKE.Dec)$, 协议的发起者拥有一对长期公私钥对 (ssk_i, spk_i), 响应者拥有一对长期公私钥对 (ssk_r, spk_r). now 表示当前时间, 用 $time$ 表示时间戳, 该时间戳用于保证攻击者无法通过重放攻击破坏当前在发起者和响应者之间的会话. ltk 表示发起者的身份, 包含使用 AEAD 加密的 $H(spk_i)$. 每个会话有其独立的会话标识符 sid, 发起者的会话标识符为 sid_i, 响应者的会话标识符为 sid_r, 用于将后续的消息回应和本条消息绑定在一起.

我们在 eCK-PFS-PSK 模型[11]下完成了安全性证明. 由于 Kyber-AKE 使用了 IND-CCA 安全的密钥封装机制, 其安全性证明可以通过使用随机比特串替换 IND-CCA 安全的密钥封装机制的会话密钥完成. 因此其安全性可以直接归约到密钥封装机制的 IND-CCA 安全性. 由于 Kyber-PFS-AKE 的设计中没有使用 IND-CCA 安全的密钥封装机制, 因此 Kyber-AKE 的安全性证明中归约到密钥封装机制的 IND-CCA 安全性的部分不能被平凡地推广到公钥加密的 IND-CPA 安全性.

然而, 在 Kyber-PFS-AKE 的设计中, IND-CPA 安全的公钥加密经过去随机化和重加密后, 结合外层的密钥导出函数, 仍然具有和 IND-CCA 安全的密钥封装机制类似的结构. 因此, 可以将外层的密钥导出函数建模为随机预言机, 在安全性证明中需要 IND-CCA 挑战者回答解密询问的部分, 可以通过密钥导出函数模拟对解密询问的回应.

此外, 为了添加一定的对拒绝服务攻击 (DDoS) 攻击的缓解手段, 可以通过 cookie 机制缓解: 当服务器过载时, 它不会立即处理握手, 而是返回一个 cookie. 响应者根据网络状态选择是否接受. 响应者维护一个随机秘密值, 该随机秘密值每两分钟更新一次, 响应者使用该随机秘密值作为 MAC 密钥, 计算发起者 IP 的 MAC 值. 在高负载的时候, 响应者会将该 cookie 返回给发起者. 发起者在重新发送消息的时候, 会使用这个 cookie 作为 MAC 密钥, 计算其消息的 MAC 值. 响应者接到发送者再次发来的消息的时候, 如果处于高负载状态, 可以根据是否使用了 cookie 计算正确的 MAC 来选择是否接受和处理这个消息. 这种机制将发起者发送的消息与其 IP 地址绑定, 证明了 IP 的所有权.

图 4 Kyber-PFS-AKE 握手协议

k	seed C_k	Key κ_k	Hash H_k
1	$H(lbl_1)$	–	$H(C_1\|\|lbl_2)$
2	$KDF_1(C_1, epk_i)$	–	$H(H_1\|\|spk_r)$
3	$KDF_1(C_2, \overline{spk_r}, m_1)$ $m_1' = CPAPKE.Dec(ssk_r, ct_1)$ if $m_1' = \perp$ or $ct_1 \neq Enc(spk_r, m_1'; G(\overline{spk_r}\|\|m_1'))$: $KDF_1(C_2, s_r, ct_1)$ else $KDF_1(C_2, \overline{spk_r}, m_1')$	$KDF_2(C_2, \overline{spk_r}, m_1)$ $m_1' = CPAPKE.Dec(ssk_r, ct_1)$ if $m_1' = \perp$ or $ct_1 \neq Enc(spk_r, m_1'; G(\overline{spk_r}\|\|m_1'))$: $KDF_2(C_2, s_r, ct_1)$ else $KDF_2(C_2, \overline{spk_r}, m_1')$	$H(H_2\|\|epk_i)$
4	$KDF_1(C_3, psk)$	$KDF_2(C_3, psk)$	$H(H_3\|\|ltk)$
5	–	–	$H(H_4\|\|time)$
6	$KDF_1(C_4, ct_2)$	–	$H(H_5\|\|ct_2)$
7	$KDF_1(C_6, e)$ $e' = CPAPKE.Dec(esk_i, ct_2)$ $if\ e' = \perp$: $KDF_1(C_6, e)$ else $KDF_1(C_6, e')$	–	–
8	$KDF_1(C_7, \overline{spk_i}, m_2)$ $m_2' = CPAPKE.Dec(ssk_i, ct_3)$ $if\ m_2' = \perp$ or $ct_3 \neq Enc(spk_i, m_2'; G(\overline{spk_i}\|\|m_2'))$ $KDF_1(C_7, s_i, ct_3)$ else $KDF_1(C_7, \overline{spk_i}, m_2')$	–	–
9	$KDF_1(C_8, psk)$	$KDF_2(C_8, psk)$	$H(H_8\|\|\kappa_9)$
10	$KDF_1(C_9, \emptyset)$	$KDF_2(C_9, \emptyset)$	$H(H_9\|\|zero)$
11	–	$tk_i = KDF_1(C_{10}, \emptyset)$ $tk_r = KDF_2(C_{10}, \emptyset)$	–

图 5 Kyber-PFS-AKE 的会话密钥导出过程

上述 cookie 方案使用在 DTLS[19]和 IKEv2[20]中. 但该 cookie 机制存在 3 个主要缺点. 首先, 为了尽可能避免暴露服务器, 原则上, 不应当给任何不能确认身份的包回应任何消息. 在服务器高负载的情况下, 如果给收到的每个消息都发送 cookie 回复会破坏这个特性. 第 2 个问题是, cookie 不应以明文发送, 因为中间人可能会利用它发送伪造的信息, 引导消息的发送者和中间人握手. 第 3 个问题是, 攻击者可以给发起者发送大量的虚假 cookie, 随后发起者会使用这些虚假的 cookie 计算消息的 *MAC*, 这样计算出来的 *MAC* 值是无效的, 但是会大量消耗发起者的计算资源.

为解决这个问题, 我们采用了双 *MAC* 机制[15] (以下用 $msg.mac_1$ 和 $msg.mac_2$ 指代) 解决了上述 3 个问题. 针对第 1 个问题, 无论是否处于高负载状态下, 发送的每一条消息都必须包含一个 *MAC* 值, 记作 $msg.mac_1$. 发起者使用响应者的公钥作为计算 $msg.mac_1$ 的密钥, 这意味着发起者至少能够确定响应者的身份. 只有发送有效的 $msg.mac_1$ 才能引发响应者的 cookie 响应. 尽管响应者的公钥并非保密的值, 但是在保证服务的隐秘性的要求下, 能够算出正确的 $msg.mac_1$ 的响应者的公钥能够证明消息的发起方知道响应者的存在. 从而可以保证握手确实在发起者和响应者之间进行.

为了解决第 2 个问题, 可以使用 AEAD 加密 cookie. AEAD 的加密密钥为响应者的公钥. 在拒绝服务攻击模型中, 这些大多是公开的值已经足够满足我们的目的. 如果存在中间人利用该 cookie 进行欺诈, 如果接收者身份 (公钥) 不正确则无法对 cookie 解密, 从而可以识别出恶意发送的 cookie. 当响应者负载过高时, 只接受包含 $msg.mac_2$ 的消息. $msg.mac_2$ 的运算使用安全传输的 cookie 作为 *MAC* 的密钥. 总的来说, 响应者在计算 $msg.mac_1$ 与 $msg.mac_2$ 并将其与消息中接收到的 *MAC* 进行比较后, 必须拒绝带有无效 $msg.mac_1$ 的消息, 并且在负载过高时也可能会拒绝带有无效 $msg.mac_2$ 的消息. 如果响应者收到一个带有有效 $msg.mac_1$ 但无效 $msg.mac_2$ 的消息, 并且处于负载状态, 它可以回应一个包含 cookie 的回复消息.

最后, 为了解决第 3 个问题, 即攻击者可能给发起者发送大量的虚假 cookie, 欺骗发起者会使用这些虚假的 cookie 计算消息的 *MAC*, 这样计算出来的 *MAC* 值是无效的, 攻击者可能通过这种方式对发起者进行 DDoS 攻击. 为了解决这个问题, WireGuard 采用的解决方案是使用 AEAD 加密 cookie 的时候, 将 AEAD 的“附加数据”字段设定为 $msg.mac_1$. cookie 回复消息中 cookie 对初始消息的第 1 个 *MAC* ($msg.mac_1$) 进行额外认证. 如果攻击者没有站在中间人位置, 由于 cookie 的计算中, 使用 AEAD 的关联数据部分绑定了初始消息, 攻击者无法向发起者发送大量无效的 cookie 回复消息阻止发起者使用正确的 cookie 进行认证. 对于处于中间人位置的攻击者 (可以任意改变或阻止消息传输), 阻止 cookie 的方法是直接去掉 cookie 回应从而阻止发起者和响应者之间建立连接. 此外, 在 $msg.mac_1$ 和 $msg.mac_2$ 的运算中, 可以不包含对消息中密文的 *MAC* 运算. 发起者给响应者发送的消息中, 包含使用响应者公钥计算的密文 ct_1, 响应者给发起者发送的消息中, 包含使用发起者临时公钥计算密文 ct_2 和发起者公钥计算密文 ct_3. 计算上千字节的杂凑值本身是个非常耗时的行为, 计算密文的杂凑值耗时几乎占据计算整个握手消息的一半. 因此, 这种做法可以极大地提升 Kyber-PFS-AKE 的计算性能. 这种做法不会影响会话密钥的安全性, 这是因为在 Kyber-PFS-AKE 的协议设计中包含对密文的解密和重加密校验. 重加密校验会检出无效的密文, 并拒绝无效的 *zero* 值, 因此无效密文会被拒绝. 验证 *MAC* 的过程可以和重加密校验并行完成, 消息中其他内容 (如: *time*,*ltk*) 长度均非常短 (32 字节), 对其进行 *MAC* 计算耗时很短.

4 算法安全性分析

本节给出 Kyber-PFS-AKE 协议在 eCK-PFS-PSK 模型下的安全性证明概述. 我们参照在 eCK-PFS-PSK 模型下的 AKE 的证明, 基于游戏跳跃 (game hop) 给出了 Kyber-PFS-AKE 的安全性证明. 因篇幅有限, 证明过程中, 情况 (case) 和游戏 (game) 的详细内容可以查阅本文的电子附件 https://blog.csdn.net/weixin_38238086.

Kyber-PFS-AKE 的证明将证明分为 3 种情况讨论, 假设敌手选中用于测试随机性的会话为 π_i^s, 敌手需要判断挑战者返回给他的比特串是均匀随机选取的, 还是会话 π_i^s, 3 种情况的区分是 π_i^s 会话是否具有匹配的会话.

1) 情况 1: π_i^s 为发起者的会话, 不存在与之匹配的会话 (意味着敌手在另一端欺骗发起者完成会话并协商出会

话密钥).

2) 情况 2: π_i^s 为响应者的会话没有与之匹配的会话 (意味着敌手在另一端欺骗响应者完成会话并协商出会话密钥).

3) 情况 3: π_i^s 会话有匹配的会话 (但是敌手依然不能区分真实的会话密钥与均匀随机的字符串).

这几种不同情况的区别在于, 敌手是否尝试伪装成某个参与方, 或者敌手是否能获取关于已建立的会话密钥的, 以及对手是否被允许拿到某个参与方的私钥. 在每种情况下, 都可以通过一系列游戏跳跃证明敌手必须直接突破 AEAD 方案的安全性, 或者区分两个信息论意义上不可区分的比特串以获取关于密钥的任何非平凡的信息, 才能成功进行攻击. 大多数游戏跳跃是使用 prf 或对偶 prf 假设的. 在这些游戏跳跃中, 用于组合两个中间值的 *KDF* 的输出 (其中至少一个是随机的, 具体取决于敌手能够破坏哪一个) 被替换为一个随机值.

证明中的一个技术难点是, 如何在 IND-CPA 安全性假设下, 处理 IND-CPA 安全的 PKE 的解密错误 (尽管解密错误的概率很低). 由于我们在保持了转换 $T[PKE,G]$ ("去随机化"和"重加密"). 转换 $U[PKE',H]$ ("哈希") 的角色由密钥派生过程中的密钥派生函数 *KDF* 扮演. 因此, 通过将 *KDF* 建模为随机预言机, 我们仍然可以通过类似 FO 变换中构造解密预言机的证明技术, 将挑战者返回的值嵌入游戏中. 这些步骤涉及的游戏跳跃主要涉及情况 1 的游戏 5 (Case 1: Game 5), 情况 2 的游戏 5 (Case 2: Game 5), 情况 3.2 的游戏 3 (Case 3.2: Game 3), 情况 3.3 的游戏 3 (Case 3.3: Game 3), 情况 3.4 的游戏 3 (Case 3.4: Game 3) 和情况 3.5 的游戏 3 (Case 3.5: Game 3). 在这些游戏中, 情况 3.2 的游戏 3 是唯一依赖于 IND-CPA 安全性的游戏.

在 eCK-PFS-PSK 模型中, 敌手 $\mathcal{A}$ 和挑战者 $\mathcal{C}$ 之间进行的实验记作 $Exp_{\mathrm{KE},n_P,n_S,\mathcal{A}}^{\text{eCK-PFS-PSK}}(\lambda)$, 其中 KE 表示密钥交换协议. 在该实验中, 挑战者维持着 n_P 个参与方 $P_1,\ldots,P_{n_P}$, 这些参与方通过协议和其他参与方交流. 每个参与方都可以运行最多 n_S 个会话. π_i^s 表示协议中用户 P_i 进行的第 s 个会话. 敌手可以通过向挑战者的不同询问来创建会话, 发送或删除用户之间发送的消息, 同时通过 *CorruptASK*(i) 获取用户 i 的长期 PKE 私钥, 通过 *CorruptEPK*(i,s) 获取会话 π_i^s 中用户 i 的临时 PKE 私钥, 并获取任意两个用户之间商定的预共享密钥.

在研究敌手是否有能力获得会话密钥的时候, 需要排除敌手最简单的打破协议安全性的方法. 敌手 $\mathcal{A}$ 最简单的方法就是在两个会话 π_i^s 与 π_j^t 之间运行协议, 其中下标 i,j 代表用户编号, 上标 s,t 代表会话编号. 敌手只要诚实地将双方发送的消息转发给对方, 最后对会话 π_i^s 进行测试 *Test*(i,s), 挑战者给敌手 $\mathcal{A}$ 返回真实的会话密钥或随机生成的会话密钥. 此时敌手只需要对另一个会话 π_j^t 进行泄露操作 *Reveal*, 泄露操作会向敌手返回会话密钥. 敌手 $\mathcal{A}$ 可以直接将拿到的会话密钥和 *Test* 返回的密钥进行对比. 如果 *Test*(i,s) 返回的密钥和 *Reveal* 返回的密钥进行对比, 如果两者一致, 则敌手 $\mathcal{A}$ 可以判定 *Test*(i,s) 返回的密钥是会话密钥, 否则判定 *Test*(i,s) 返回的是随机生成的密钥. 在这种情况下敌手可以平凡的打破协议的安全性. 因此, 在 eCK-PFS-PSK 模型中, 需要通过贡献式密钥份额 (contributive keyshares) 和清洁谓词 ($clean_{\text{eCK-PFS-PSK}}$) 的定义排除这种情况, 两个定义如定义 4、定义 5 所示.

定义 4. 贡献式密钥份额. 令 $\pi_i^s.kid$ 为会话 π_i^s 在协议执行过程中发送的所有用于生成会话密钥的材料, $\pi_i^s.m_r$ 为会话 π_i^s 在协议执行过程中收到的所有消息的连接. 如果 $\pi_j^t.kid$ 是 $\pi_i^s.m_r$ 的子串, 则称 π_j^t 为 π_i^s 的贡献式密钥份额会话 (contributive keyshares session), 简称两个会话匹配 (match).

清洁谓词 ($clean_{\text{eCK-PFS-PSK}}$) 定义了安全实验中可以向敌手 $\mathcal{A}$ 泄露且不会平凡的破坏协议的安全性的密钥组合. 符合清洁谓词的要求意味着满足完美前向安全性 (PFS) 以及可以抵抗 KCI (key compromise impersonation) 攻击. KCI 攻击指的是敌手拿到 Bob 的全部长期私钥后, 尝试在 Bob 面前扮演 Alice.

定义 5. 清洁谓词 ($clean_{\text{eCK-PFS-PSK}}$) 会话. π_i^s 如果满足以下所有条件, 且 π_i^s 最终接受了会话密钥, 则称该会话是清洁的: 1) 敌手未泄露 π_i^s 会话密钥; 2) 对于所有与 π_i^s 匹配的会话 π_j^t, 敌手未泄露 π_j^t 的会话密钥; 3) 如果用户 i,j 之间的预共享密钥被泄露, 则敌手不能同时通过查询 *CorruptASK*(i) 和 *CorruptEPK*(i,s) 获得用户 i 的长期私钥以及 π_i^s 的临时私钥; 4) 如果用户 i,j 之间的预共享密钥被泄露, 且 π_i^s 与 π_j^t 匹配, 则敌手不能同时通过查询 *CorruptASK*(j) 和 *CorruptEPK*(j,t) 获得用户 j 的长期私钥以及 π_j^t 的临时私钥; 5) 如果不存在与 π_i^s 匹配的会话 π_j^t, 则在 π_i^s 接受会话密钥之前, 敌手不能查询 *CorruptASK*(j) (防止敌手平凡的扮演用户 j).

定理 1. Kyber-PFS-AKE 在满足清洁谓词的情况下, 是 eCK-PFS-PSK 安全的. 即任意敌手 $\mathcal{A}$ 在密钥不可区分游戏中的优势 $Adv^{\text{eCK-PFS-PSK}}_{clean_{\text{eCK-PFS-PSK},n_P,n_S},\mathcal{A}}(\lambda)$ 都被限制在敌手 $\mathcal{A}$ 在对偶-PRF, IND-CPA 以及 auth-aead 游戏中的优势, 具体来说:

$$\begin{aligned}Adv^{\text{eCK-PFS-PSK}}_{clean_{\text{eCK-PFS-PSK},n_P,n_S},\mathcal{A}}(\lambda) \leqslant\ & n_P^2 n_S\left(\frac{n_S}{2^\lambda}+Adv(\mathcal{A}')+6\cdot Adv^{prf}_{KDF,\mathcal{R}}(\lambda)+Adv^{prf^{\text{swap}}}_{KDF,\mathcal{R}}(\lambda)+Adv^{\text{auth-aead}}_{\text{AEAD},\mathcal{R}}(\lambda)\right)\\ &+n_P^2 n_S\left(\frac{n_S}{2^\lambda}+Adv(\mathcal{A}')+3\cdot Adv^{prf}_{KDF,\mathcal{R}}(\lambda)+Adv^{prf^{\text{swap}}}_{KDF,\mathcal{R}}(\lambda)+Adv^{\text{auth-aead}}_{\text{AEAD},\mathcal{R}}(\lambda)\right)\\ &+\max\begin{cases}n_P^2 n_S^2(2\cdot Adv^{prf}_{KDF,\mathcal{R}}(\lambda)+Adv^{prf^{\text{swap}}}_{KDF,\mathcal{R}}(\lambda))\\ n_P^2 n_S^2(Adv^{\text{IND-CPA}}_{CPAPKE,\mathcal{R}}(\mathcal{B})+4\cdot Adv^{prf}_{KDF,\mathcal{R}}(\lambda)+Adv^{prf^{\text{swap}}}_{KDF,\mathcal{R}}(\lambda))\\ n_P^2 n_S^2(Adv(\mathcal{A}')+7\cdot Adv^{prf}_{KDF,\mathcal{R}}(\lambda))\\ n_P^2 n_S^2(Adv(\mathcal{A}')+3\cdot Adv^{prf}_{KDF,\mathcal{R}}(\lambda)+Adv^{prf^{\text{swap}}}_{KDF,\mathcal{R}}(\lambda))\\ n_P^2 n_S^2\left(\frac{n_S}{2^\lambda}+Adv(\mathcal{A}')+7\cdot Adv^{prf}_{KDF,\mathcal{R}}(\lambda)\right)\end{cases},\end{aligned}$$

其中, $Adv^{\text{IND-CPA}}_{\text{PKE}}(\mathcal{B})$ 是任意敌手 $\mathcal{B}$ 对 IND-CPA 安全的 PKE 游戏中获胜的概率. 在多用户设置中 (即存在多个拥有不同公钥的接收者, 将同一个消息 m 使用对应的公钥发送给这些接收者, 随后敌手尝试解密该消息. 在特定的参数设置下, 敌手可以解密出 RSA 密文. 考虑攻击者破解多个用户密文的优势. 在量子随机预言模型 (QROM) 下, 敌手 $\mathcal{A}'$ 的优势为:

$$Adv(\mathcal{A}')\leqslant 2\sqrt{qAdv^{(n,q_C)\text{-IND-CPA}}_{\text{PKE}}(\mathcal{B})}+4q\sqrt{\frac{q_C\cdot n}{|\mathcal{M}|}}+\frac{n^2}{2^l}+4(q_F+1)\sqrt{\frac{n}{2^l}}+16q^2\delta(n)+\frac{q_C^2}{|\mathcal{M}|},$$

其中, $q:=q_F+q_D+1$, n 是用户的数量, q_C 是挑战密文的个数, q_F 是 (Q)ROM 查询的最大次数, q_D 是解密查询的最大次数.

5 算法实现性能评估

我们使用 Kyber-768 进行实验, 用于对比 Kyber-PFS-AKE 和 Kyber-AKE 的性能. Kyber-768 是 Kyber 向 NIST 算法竞赛提交的算法实现代码中, 安全级别为 III (其经典安全强度为 182 比特, 其量子安全强度为 165 比特). 我们参照 NIST 颁布的标准, 使用 Kyber 的第 3 轮提案提供的 AVX2 指令集上实现的代码基础上进行实验. 注意到 Kyber-768 提供了两个版本, Kyber-768 和 Kyber-768-90s, 两者的区别在于随机数发生器不同. 尽管 Kyber-768-90s 拥有更高的计算性能, 但根据 NIST 给出的报告, Kyber-768-90s 不会被标准化, 因而我们只选用了 Kyber-768 的实现代码. 此外, 由于 Kyber-AKE 来自 Fujioka 等人给出的通用构造[5], 其用于生成临时公私钥对 (epk_i, esk_i) 的 KEM 可以为 IND-CPA 安全的 KEM, 在实验中我们采用 IND-CPA 安全的 KEM 进行该部分计算.

协议中存在的其他模块, 对于密钥导出运算 (即 *KDF*), 我们使用 BLAKE2s 算法[21], 对于 AEAD, 我们采用了 ChaCha20-Poly1305 算法[22]. 这两种算法的也在 WireGuard、Noise 中使用. 因此我们采取了与其一致的代码. 所有基准测试均在 Intel Core i7-6700 处理器上进行, 时钟频率为 900 MHz (查询/proc/cpuinfo 获得), 关闭了 TurboBoost 和超线程, 基准测试机器具有 16 GB 的 RAM, 运行 Ubuntu 操作系统, 内核版本为 5.4.0. 报告的所有循环计数均为各自函数的 10000 次执行的中值. 对比结果如表 1 所示.

表 1 实验数据集

性能指标	Kyber-AKE	Kyber-PFS-AKE	计算性能提升百分比 (%)
发起者计算时间	258464	158445	38.6
响应者计算时间	234297	162289	30.7

如表 1 所示, 与 Kyber-AKE 相比, 我们的设计在发起方和响应方设备上均节省了近 30% 的计算工作量. 影响握手时间的另一个重要因素是数据传输, 而在我们的构建和 Kyber-AKE 中传输的数据长度相同, 这意味着两个协

议的数据传输时间大致相同, 握手时间的差异主要来自它们不同的计算性能.

6 总 结

Kyber 是一个基于格上困难问题的密钥封装机制 (KEM), 在 2023 年被美国国家标准与技术研究院 (NIST) 宣布为第 1 个被标准化的 KEM. Kyber-AKE 是 Kyber 的设计者基于 Kyber KEM 构造的弱前向安全的认证密钥交换 (AKE), 通过使用 3 个 IND-CCA 安全的 KEM 在两轮内协商会话密钥,

在本文中, 我们介绍了 Kyber-PFS-AKE, 这是一种新的 AKE 构造方法. Kyber-PFS-AKE 只使用了 IND-CPA 安全的公钥加密 (PKE) 方案, 从而简化了后量子 Kyber-AKE 的设计. 我们严格证明了基于 IND-CCA KEM 的 Kyber-AKE 协议中某些操作是冗余的. 去除这些冗余后, 协议变得更加简化和高效. 通过仅使用具有被动安全性的 PKE, 并通过 FO 变换中的类似技术处理被动安全的 PKE 的解密错误. 我们在 eCK-PFS-PSK 模型下证明了 Kyber-PFS-AKE 的会话密钥不可区分性质, 以及完美的前向安全性等安全性质. 根据 eCK-PFS-PSK 模型的定义, 以及清洁谓词的要求, 保证了 Kyber-PFS-AKE 协议的完美前向安全性. 通用的从 IND-CPA 安全的 AKE 构造框架是无法做到完美的前向安全性的. 我们使用量子安全为 165 比特的 Kyber-768. PKE 实现了 Kyber-PFS-AKE. 我们的实验结果表明, 与 Kyber-PFS-AKE 相比, 我们的构造使得协议发起者的计算时间上降低了 38%, 在响应者计算时间上降低了 30%.

References:

[1] NIST. Call for proposals. 2017. https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/Call-for-Proposals

[2] Bos J, Ducas L, Kiltz E, Lepoint T, Lyubashevsky V, Schanck JM, Schwabe P, Seiler G, Stehlé D. CRYSTALS-Kyber: A CCA-secure module-lattice-based KEM. In: Proc. of the 2018 IEEE European Symp. on Security and Privacy (EuroS&P). London: IEEE, 2018. 353–367. [doi: 10.1109/EuroSP.2018.00032]

[3] Alagic G, Apon D, Cooper D, Dang Q, Dang T, Kelsey J, Lichtinger J, Miller C, Moody D, Peralta R, Perlner R, Robinson A, Smith-Tone D, Liu YK. Status report on the third round of the NIST post-quantum cryptography standardization process. US Department of Commerce, NIST, 2022.

[4] Narisada S, Uemura S, Okada H, Furue H, Aikawa Y, Fukushima K. Solving McEliece-1409 in one day—Cryptanalysis with the improved BJMM algorithm. In: Proc. of the 27th Int'l Conf. on Information Security. Arlington: Springer, 2025. 3–23. [doi: 10.1007/978-3-031-75764-8_1]

[5] Fujioka A, Suzuki K, Xagawa K, Yoneyama K. Strongly secure authenticated key exchange from factoring, codes, and lattices. In: Proc. of the 15th Int'l Conf. on Practice and Theory in Public Key Cryptography. Darmstadt: Springer, 2012. 467–484. [doi: 10.1007/978-3-642-30057-8_28]

[6] Canetti R, Krawczyk H. Analysis of key-exchange protocols and their use for building secure channels. In: Proc. of the 2001 Int'l Conf. on the Theory and Application of Cryptographic Techniques. Innsbruck: Springer, 2001. 453–474. [doi: 10.1007/3-540-44987-6_28]

[7] Dowling B, Paterson KG. A cryptographic analysis of the WireGuard protocol. In: Proc. of the 16th Int'l Conf. on Applied Cryptography and Network Security. Leuven: Springer, 2018. 3–21. [doi: 10.1007/978-3-319-93387-0_1]

[8] Håstad J. Solving simultaneous modular equations of low degree. SIAM Journal of Computing, 1988, 17(2): 336–341. [doi: 10.1137/0217019]

[9] D'Anvers JP, Karmakar A, Roy SS, Vercauteren F, *et al*. Saber: Mod-LWR based KEM algorithm specification and supporting documentation. 2019. https://www.esat.kuleuven.be/cosic/pqcrypto/saber/

[10] Duman J, Hövelmanns K, Kiltz E, Lyubashevsky V, Seiler G. Faster lattice-based KEMs via a generic Fujisaki-Okamoto transform using prefix hashing. In: Proc. of the 2021 ACM SIGSAC Conf. on Computer and Communications Security. Springer, 2021. 2722–2737. [doi: 10.1145/3460120.3484819]

[11] Hülsing A, Ning KC, Schwabe P, Weber FJ, Zimmermann PR. Post-quantum WireGuard. In: Proc. of the 2021 IEEE Symp. on Security and Privacy (SP). San Francisco: IEEE, 2021. 304–321. [doi: 10.1109/SP40001.2021.00030]

[12] Paquin C, Stebila D, Tamvada G. Benchmarking post-quantum cryptography in TLS. In: Proc. of the 11th Int'l Conf. on Post-quantum Cryptography. Paris: Springer, 2020. 72–91. [doi: 10.1007/978-3-030-44223-1_5]

[13] Sosnowski M, Wiedner F, Hauser E, Steger L, Schoinianakis D, Gallenmüller S, Carle G. The performance of post-quantum TLS 1.3. In:

Proc. of the 19th Int'l Conf. on Emerging Networking Experiments and Technologies. Paris: ACM, 2023. 19–27. [doi: 10.1145/3624354.3630585]

[14] Schwabe P, Stebila D, Wiggers T. Post-quantum TLS without handshake signatures. In: Proc. of the 2020 ACM SIGSAC Conf. on Computer and Communications Security. ACM, 2020. 1461–1480. [doi: 10.1145/3372297.3423350]

[15] Donenfeld JA. WireGuard: Next generation kernel network tunnel. In: Proc. of the 24th Annual Network and Distributed System Security Symp. San Diego: The Internet Society, 2017. [doi: 10.14722/ndss.2017.23160]

[16] Li ZC, Xie T, Zhang JM, Xu RH. Post quantum authenticated key exchange protocol based on ring learning with errors problem. Journal of Computer Research and Development, 2019, 56(12): 2694–2701. (in Chinese with English abstract) [doi: 10.7544/issn1000-1239.2019.20180874]

[17] Xue HY, Lu XH, Li B, Liang B, He JN. Understanding and constructing AKE via double-key key encapsulation mechanism. In: Proc. of the 24th Int'l Conf. on the Theory and Application of Cryptology and Information Security. Brisbane: Springer, 2018. 158–189. [doi: 10.1007/978-3-030-03329-3_6]

[18] Hövelmanns K, Kiltz E, Schäge S, Unruh D. Generic authenticated key exchange in the quantum random oracle model. In: Proc. of the 23rd IACR Int'l Conf. on Practice and Theory of Public-key Cryptography. Edinburgh: Springer, 2020. 389–422. [doi: 10.1007/978-3-030-45388-6_14]

[19] Rescorla E, Modadugu N. Datagram transport layer security version 1.2. RFC 6347. 2012.

[20] Kaufman C, Hoffman P, Nir Y, Eronen P. Internet key exchange protocol version 2. RFC 5996. 2010.

[21] Aumasson JP, Neves S, Wilcox-O'Hearn Z, Winnerlein C. BLAKE2: Simpler, smaller, fast as MD5. In: Proc. of the 11th Int'l Conf. on Applied Cryptography and Network Security. Banff: Springer, 2013. 119–135. [doi: 10.1007/978-3-642-38980-1_8]

[22] Nir Y, Langley A. ChaCha20 and Poly1305 for IETF protocols. IETF RFC 8439, 2018. [doi: 10.17487/RFC8439]

附中文参考文献:

[16] 李子臣, 谢婷, 张卷美, 徐荣华. 基于 RLWE 的后量子认证密钥交换协议. 计算机研究与发展, 2019, 56(12): 2694–2701. [doi: 10.7544/issn1000-1239.2019.20180874]

米瑞琪(1995－), 女, 博士生, 主要研究领域为后量子密码的设计与分析.

张振峰(1972－), 男, 博士, 研究员, 博士生导师, 主要研究领域为密码学与数据安全.

江浩东(1989－), 男, 博士, 主要研究领域为抗量子密码研究, 量子可证明安全理论, 格密码学设计与分析, 量子算法与量子查询复杂度理论.