面向多服务器智慧医疗环境的后量子口令认证与快速切换密钥协商协议

王雄，张震，郭豪，张迪，刘长天
（北京电子科技学院网络空间安全系，北京 100070）

摘要：针对智慧医疗中医生跨多服务器综合诊治的需求，以及现有后量子认证方案缺乏高效切换机制问题，提出一种面向多服务器智慧医疗环境的后量子口令认证与快速切换密钥协商协议。协议以ML-KEM为核心，设计初始认证与快速切换认证相结合的机制，实现首次完整认证并建立主会话密钥和业务会话密钥，多服务器访问时快速切换认证并独立协商新业务会话密钥。安全性方面，ROR模型下证明协议满足会话密钥语义安全，并能够抵抗常见攻击。性能方面，当用户连续访问5个服务器时，相比Wen和Chen两种后量子方案重复执行完整认证的方式，协议的累计计算开销降低70.34%以上，累计通信开销降低27.53%以上。该性能差距随服务器数量增加而扩大，说明本文协议在大规模部署场景下具有显著的效率优势和良好的可扩展性。

关键词：后量子密码；ML-KEM；口令认证；多服务器认证；快速切换认证
中图分类号：TP309
文献标志码：A

A Post-Quantum Password Authentication and Fast-Switching Key Agreement Protocol for Multi-Server Smart Healthcare Environments

Wang Xiong, Zhang Zhen, Guo Hao, Zhang Di, Liu Changtian
Dept. of Cyberspace Security, Beijing Electronic Science and Technology Institute, Beijing 100070, China

Abstract: To meet the requirement of doctors' comprehensive diagnosis and treatment across multiple servers in smart healthcare, and to address the lack of efficient switching mechanisms in existing post-quantum authentication schemes, a post-quantum password authentication and fast-switching key agreement protocol for multi-server smart healthcare environments was proposed. With ML-KEM as the core cryptographic primitive, a mechanism combining initial authentication and fast switching authentication was designed. In the initial authentication phase, full authentication was performed, and a master session key and a service session key were established. During multi-server access, fast switching authentication was performed, and a new service session key was independently negotiated with the target server. In terms of security, the session key semantic security of the proposed protocol was proved under the ROR model, and resistance to common attacks was analyzed. In terms of performance, when a user continuously accesses five servers, compared with the repeated full-authentication approach in Wen's and Chen's post-quantum schemes, the proposed protocol reduces the cumulative computational overhead by more than 70.34% and the cumulative communication overhead by more than 27.53%. The performance gap increases as the number of servers grows, which shows that the proposed protocol has significant efficiency advantages and good scalability in large-scale deployment scenarios.

Key words: post-quantum cryptography, ML-KEM, password authentication, multi-server authentication, fast switching

收稿日期：XXXX–XX–XX；修回日期：XXXX–XX–XX
通信作者：张震，3608430243@qq.com
基金项目：中央高校基本科研业务费资金（No.3282024056, No.3282024022）
Foundation Items: The Fundamental Research Funds for the Central Universities(No.3282024056, No.3282024022)

authentication

0 引言

随着物联网、云计算等技术在医疗领域的应用，智慧医疗系统逐渐由单一服务模式向多服务器协同服务模式发展。在远程会诊、移动查房等场景中，医生可能连续访问电子病历、医学影像等多个医疗服务节点。与此同时，医疗数据具有长期敏感性，传统基于整数分解和离散对数等困难问题的公钥密码体制在量子计算环境下面临潜在威胁[1]。美国国家标准与技术研究院（national institute of standards and technology, NIST）发布后量子密码标准后，模格密钥封装机制（module lattice based key encapsulation mechanism, ML-KEM）等后量子密钥封装机制成为公开信道上实现后量子安全密钥建立的重要基础[2]。然而，现有后量子认证方案多面向固定服务器、云辅助医疗等场景，对智慧医疗多服务器连续访问过程中的认证效率考虑不足。

在多服务器认证场景中，重复执行完整认证能够保持较清晰的安全边界，但累计计算开销和通信开销会随访问服务器数量增加而线性增长；文献[3-5]提出的单点登录或授权机制能够使用户完成一次认证后访问多个服务节点，减少重复认证次数，但统一票据可能导致权限边界过宽和凭证泄露影响范围扩大等问题。因此，如何在保证后量子安全和目标服务器独立参与认证的前提下，降低用户跨服务器访问时的重复认证开销，成为智慧医疗多服务器认证中需要解决的问题。

1 相关工作

1.1 现存工作

口令认证因部署成本低、使用便捷，是远程身份认证中常用的认证方式。早期口令认证研究[6-8]主要面向两方认证场景，难以直接满足用户跨多个医疗服务节点连续访问的需求，因此多服务器认证协议逐渐受到关注。Chuang等人[9]提出面向多服务器环境的匿名认证方案，Mishra等人[10]指出其存在匿名性不足和服务器伪装攻击等问题并提出改进方案。后续研究[11-13]围绕智能卡、生物特征、动态身份和三因子认证等方向对多服务器认证协议进行了改进。Barman等人[14]、Ali等人[15]和Luo等人[16]进一步面向移动多服务器、云服务和电子医疗等场景优化了用户匿名性、会话密钥安全和离线注册中心等性能。王雄等人[17, 18]进一步针对无线医疗传感器网络和多网关医疗认证场景提出轻量级匿名认证协议。为降低用户跨服务器访问时的重复认证开销，汪文明等人[19]提出了自适应快速切换认证方案。

然而，上述方案大多基于ECC、无证书公钥或其他传统密码机制。后量子认证协议已逐渐被应用于医疗物联网和云辅助医疗环境。Bahache等人[20]提出面向云医疗应用的抗量子认证与密钥协商框架，Ahmad等人[21]和Chen等人[22]设计了面向云辅助场景的量子安全多因素认证协议。Palaniswamy等人[23]提出了适用于轻量设备的隐私保护认证方案。Wen等人[24]提出的后量子多服务器认证协议与本文研究较为接近，但用户访问不同服务器时仍需执行完整认证过程，未考虑完成初始认证后的跨服务器快速切换认证问题。

综上，现有研究仍缺少一种同时支持后量子安全、多服务器认证、低重复认证开销和目标服务器独立参与认证的智慧医疗认证密钥协商协议。因此，本文提出面向多服务器智慧医疗环境的后量子口令认证与快速切换密钥协商协议。

1.2 本文工作

为解决上述问题，本文提出一种面向多服务器智慧医疗环境的后量子口令认证与快速切换密钥协商协议。主要贡献如下：

（1）提出一种基于ML-KEM的后量子多服务器认证协议，实现用户与医疗服务器之间的双向认证和会话密钥协商，并支持离线注册中心。

（2）设计初始认证与快速切换认证相结合的机制。用户首次访问服务器时执行完整认证并建立主会话密钥，后续访问其他服务器时在当前服务器辅助下完成快速切换认证，从而降低重复完整认证带来的计算和通信开销。

（3）在协议中引入用户与服务器绑定认证值、服务器长期隐藏值和快速切换辅助值，将用户假名、目标服务器身份和切换上下文绑定，增强协议对离线口令猜测、冒充、中间人和非法跨服务器切换等攻击的抵抗能力。

（4）在真实-随机（real-or-random, ROR）模型

下证明协议满足会话密钥语义安全，并分析其对常见攻击的抵抗能力。

2　预备知识与系统模型

2.1　MLWE 问题

模学习错误问题（module learning with errors, MLWE）是格密码中的重要困难问题，也是 ML-KEM 密钥封装机制的安全基础之一。设 $R_q = \mathbb{Z}_q[x]/(f(x))$ 为模 q 上的多项式环，k 为模维度，χ 为误差分布。随机选取矩阵 $A \leftarrow R_q^{k\times k}$、秘密向量 $s \leftarrow \chi^k$ 和误差向量 $e \leftarrow \chi^k$，计算 $b = As + e \in R_q^k$。

决策 MLWE 问题要求敌手区分样本 (A,b) 与均匀随机样本 (A,u)，其中 $u \leftarrow R_q^k$。若任意概率多项式时间敌手都不能以不可忽略优势完成区分，则称 MLWE 问题是困难的。基于 MLWE 问题的密码机制适用于构造后量子安全的认证与密钥协商协议。

2.2　ML-KEM 密钥封装机制

ML-KEM 是基于 Module-LWE 问题构造的后量子密钥封装机制，已被 NIST 标准化。

（1）密钥生成算法：密钥生成算法输入安全参数 1^λ，输出一对公私钥 $(pk,sk) \leftarrow KeyGen(1^\lambda)$，其中，$pk$ 为公钥，sk 为私钥。

（2）封装算法：封装算法输入接收方公钥 pk，输出密文 c 和共享密钥 K，$(c,K) \leftarrow Encaps(pk)$，其中，$c$ 可以通过公开信道传输，K 作为双方后续密钥派生的输入。

（3）解封装算法：解封装算法输入接收方私钥 sk 和密文 c，输出共享密钥 K' 或失败符号 $\perp$，$K' \leftarrow Decaps(sk,c)$。若密文 c 是由合法公钥 pk 通过 $Encaps(pk)$ 生成的，则解封装得到的 K' 应与封装得到的 K 一致。其正确性可表示为：$Pr[(c,K) \leftarrow Encaps(pk), K' = Decaps(sk,c) : K' = K] \geqslant 1 - \delta$。其中，$\delta$ 为可忽略的错误概率。

2.3　系统模型

本文考虑智慧医疗多服务器环境下的用户认证与跨服务器快速切换访问场景，结构如图 1 所示。

系统主要包括注册中心 RC、用户 U_i 和多个医疗服务器 S_j。其中，用户 U_i 可以是医生、患者或移动医疗终端；医疗服务器 S_j 对应智慧医疗系统中的不同业务服务节点，例如电子病历、健康监测和远程诊疗等服务器。注册中心 RC 是系统中的可信实体，主要负责系统初始化、用户注册和服务器注册。

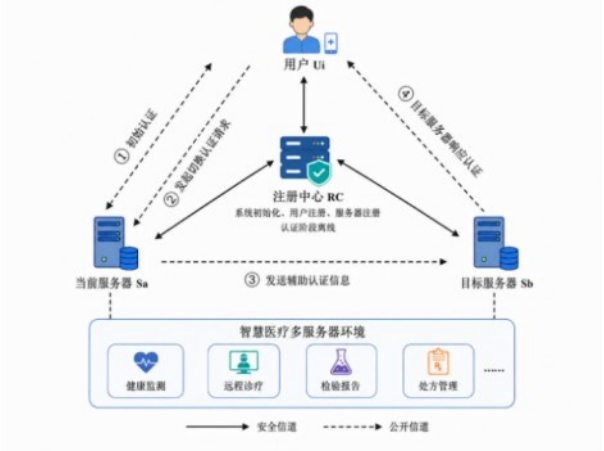


图 1　系统模型

3　协议设计

协议包括系统初始化、服务器注册、用户注册、初始认证与密钥协商、快速切换认证五个阶段。其中，系统初始化、服务器注册和用户注册阶段用于为用户和服务器分发长期认证参数；初始认证阶段用于完成用户与当前服务器 S_a 之间的完整认证，并建立主会话密钥 $MK_{i,a}$ 和业务会话密钥 $SK_{i,a}$；快速切换认证阶段用于在当前服务器 S_a 的辅助下，使用户 U_i 与目标服务器 S_b 完成认证并建立新的业务会话密钥 $SK_{i,b}$。协议中使用的主要符号如表 1 所示。

表 1　　符号和说明

符号	说明	符号	说明
RC	注册中心	PSR_j	服务器长期隐藏值
U_i	第 i 个用户	HPW_i	用户口令派生值
S_j	第 j 个服务器	PID_i	用户假名身份
S_a	当前已认证服务器	V_{ij}, P_{ij}	用户与 S_j 的绑定值
S_b	目标切换服务器	$MK_{i,a}$	主会话密钥
ID_i	用户真实身份	$SK_{i,a}$	业务会话密钥
PW_i	用户口令	K_{ab}	快速切换共享密钥
SID_j	服务器身份	E_{ab}	辅助恢复值
K_{RC}	注册中心系统根秘密	ΔT	最大允许传输时延
X_{ij}	用户与 S_j 的绑定认证值	Δ_{ab}	当前服务器辅助掩码

3.1　系统初始化

系统初始化阶段由注册中心 RC 执行。首先，

RC选取安全参数λ，确定哈希函数h、密钥派生函数KDF、对称加密/解密函数Enc_{sym}/Dec_{sym}、密钥生成函数KeyGen、密钥封装/解封装函数Encaps/Decaps。然后RC随机生成系统根秘密$K_{RC} \leftarrow \{0,1\}^{\lambda}$，用于生成合法服务器长期隐藏值。最后，$RC$发布$params = \{h, KDF, \mathrm{Enc}_{sym}, \mathrm{Dec}_{sym}, \mathrm{KeyGen}, \mathrm{Encaps}, \mathrm{Decaps}\}$，并安全保存$K_{RC}$。

3.2 服务器注册阶段

服务器注册阶段用于将合法医疗服务器加入系统。其注册过程如下。

（1）服务器S_j选择身份标识SID_j，生成公私钥对$(pk_j^{kem}, sk_j^{kem}) \leftarrow \mathrm{KeyGen}(1^{\lambda})$，其中，$pk_j^{kem}$为KEM公钥，$sk_j^{kem}$为私钥。服务器$S_j$向$RC$提交注册信息：$\{SID_j, pk_j^{kem}\}$。

（2）RC收到注册请求后，首先检查SID_j是否合法且未被占用。通过后，RC利用系统根秘密K_{RC}计算服务器长期隐藏值$PSR_j = h(SID_j \mathbin{//} K_{RC})$，随后，$RC$将$\mathcal{D} = \{(SID_j, pk_j^{kem})\}$写入服务器目录。$RC$向服务器$S_j$返回：$\{SID_j, PSR_j\}$。

（3）服务器S_j收到该消息后，在本地安全存储$\{SID_j, sk_j^{kem}, PSR_j\}$。

服务器注册完之后，RC将服务器目录$\mathcal{D}$下发至所有合法服务器。

3.3 用户注册阶段

用户注册阶段用于为用户U_i生成与各服务器的认证绑定信息。用户U_i选择身份标识ID_i和口令PW_i，并在本地终端执行如下操作。

（1）生成随机盐值$a_i \leftarrow \{0,1\}^{\lambda}$，计算口令派生值$HPW_i = h(ID_i \mathbin{//} PW_i \mathbin{//} a_i)$，生成用户假名$PID_i = h(ID_i \mathbin{//} a_i)$，用户$U_i$向$RC$发送注册请求：$\{PID_i, HPW_i\}$。

（2）RC收到用户注册请求后，检查PID_i是否唯一。若PID_i已经存在，则拒绝该注册请求；否则，RC遍历服务器目录$\mathcal{D}$，为用户U_i计算服务器绑定值。对于任意服务器S_j，RC计算$V_{ij} = h(PID_i \mathbin{//} PSR_j) \oplus HPW_i$，$P_{ij} = h(SID_j \mathbin{//} PSR_j) \oplus HPW_i$，其中，$V_{ij}$用于初始认证和快速切换认证阶段验证用户假名与服务器之间的注册绑定关系的相关认证材料，P_{ij}用于初始认证阶段帮助用户构造能够被服务器验证的认证值。随后，RC向用户U_i返回：$\{\mathcal{D}, (SID_j, V_{ij}, P_{ij})_{S_j \in \mathcal{D}}\}$。

（3）用户收到消息后，在本地安全存储$\{a_i, PID_i, \mathcal{D}, (SID_j, V_{ij}, P_{ij})_{S_j \in \mathcal{D}}\}$。

3.4 初始认证与密钥协商阶段

当用户U_i首次访问服务器S_a时，双方执行初始认证与密钥协商，如图2所示。

（1）用户发起认证

图2 初始认证与密钥协商阶段

User	ServerA

输入: (ID_i, PW_i)
计算: $HPW_i' = h(ID_i \mathbin{//} PW_i \mathbin{//} a_i)$
$X_{ia} = HPW_i' \oplus V_{ia}$
$M_0 = HPW_i' \oplus P_{ia}$
生成随机数x_i、时间戳T_1
$(pk_i^e, sk_i^e) \leftarrow \mathrm{KeyGen}(1^{\lambda})$
$(c_1, K_1) \leftarrow \mathrm{Encaps}(pk_a^{kem})$
$K_{1e} = \mathrm{KDF}(K_1 \mathbin{//} \text{"enc"})$
$Auth_1 = h(M_0 \mathbin{//} X_{ia} \mathbin{//} PID_i \mathbin{//} x_i \mathbin{//} pk_i^e \mathbin{//} K_1 \mathbin{//} T_1)$
$M_1 = \mathrm{Enc}_{sym}(K_{1e}, M_0 \mathbin{//} X_{ia} \mathbin{//} PID_i \mathbin{//} x_i \mathbin{//} pk_i^e \mathbin{//} Auth_1)$
$Msg_1 = (c_1, M_1, T_1)$
$\rightarrow$	

	如果$\lvert T_1' - T_1 \rvert \le \Delta T$成立，则继续
	计算: $K_1' = \mathrm{Decaps}(sk_a^{kem}, c_1)$
	$K_{1e}' = \mathrm{KDF}(K_1' \mathbin{//} \text{"enc"})$
	$(M_0, X_{ia}, PID_i, x_i, pk_i^e, Auth_1) = \mathrm{Dec}_{sym}(K_{1e}', M_1)$
	验证：$Auth_1 = h(M_0 \mathbin{//} X_{ia} \mathbin{//} PID_i \mathbin{//} x_i \mathbin{//} pk_i^e \mathbin{//} K_1' \mathbin{//} T_1)$
	$X_{ia} = h(PID_i \mathbin{//} PSR_a)$
	$M_0 = h(SID_a \mathbin{//} PSR_a)$
	生成随机数y_a、时间戳T_2
	$(c_2, K_2) \leftarrow \mathrm{Encaps}(pk_i^e)$
	$MK_{i,a} = KDF(PID_i \mathbin{//} SID_a \mathbin{//} x_i \mathbin{//} y_a \mathbin{//} K_1' \mathbin{//} K_2)$
	$SK_{i,a} = KDF(MK_{i,a} \mathbin{//} \text{"session"})$
	$Auth_2 = h(MK_{i,a} \mathbin{//} c_2 \mathbin{//} y_a \mathbin{//} T_2)$

$Msg_2 = (c_2, y_a, Auth_2, T_2)$ $\leftarrow$	

如果$\lvert T_2' - T_2 \rvert \le \Delta T$成立，则继续
计算: $K_2' = Decaps(sk_i^e, c_2)$
$MK_{i,a}' = KDF(PID_i \mathbin{//} SID_a \mathbin{//} x_i \mathbin{//} y_a \mathbin{//} K_1 \mathbin{//} K_2')$
$Auth_2' = h(MK_{i,a}' \mathbin{//} c_2 \mathbin{//} y_a \mathbin{//} T_2)$
验证：$Auth_2 = Auth_2'$
$SK_{i,a}' = KDF(MK_{i,a}' \mathbin{//} \text{"session"})$	

用户 U_i 输入身份标识和口令 (ID_i,PW_i)，计算 $HPW_i' = h(ID_i \mathbin{//} PW_i \mathbin{//} a_i)$，选择当前访问服务器 S_a，并从本地服务器目录 $\mathcal{D}$ 中获得 S_a 的身份标识 SID_a 及公钥 pk_a^{kem}。同时，用户终端从本地存储中取出与服务器 S_a 对应的绑定值 (V_{ia},P_{ia})，计算 $X_{ia} = HPW_i' \oplus V_{ia}$，$M_0 = HPW_i' \oplus P_{ia}$，生成随机数 x_i 和时间戳 T_1，生成临时 KEM 密钥对 $(pk_i^e,sk_i^e) \leftarrow \mathrm{KeyGen}(1^\lambda)$，使用服务器 S_a 的 KEM 公钥执行封装 $(c_1,K_1) \leftarrow \mathrm{Encaps}(pk_a^{kem})$，派生对称加密密钥 $K_{1e} = KDF(K_1 \mathbin{//} \text{“enc”})$，计算认证标签 $Auth_1 = h(M_0 \mathbin{//} X_{ia} \mathbin{//} PID_i \mathbin{//} x_i \mathbin{//} pk_i^e \mathbin{//} K_1 \mathbin{//} T_1)$，构造 $M_1 = \mathrm{Enc}_{sym}(K_{1e},M_0 \mathbin{//} X_{ia} \mathbin{//} PID_i \mathbin{//} x_i \mathbin{//} pk_i^e \mathbin{//} Auth_1)$。用户 U_i 向服务器 S_a 发送 $Msg_1 = (c_1,M_1,T_1)$。

（2）服务器响应并完成密钥协商

服务器 S_a 收到用户发送的 Msg_1 后，首先记录当前时间戳 T_1'，并检查消息是否在允许的时间范围内到达，如果 $|T_1' - T_1| \leqslant \Delta T$ 不成立，则服务器终止本次会话；否则，服务器解封装 $K_1' = \mathrm{Decaps}(sk_a^{kem},c_1)$，计算 $K_{1e}' = KDF(K_1' \mathbin{//} \text{“enc”})$，解密 M_1：$(M_0,X_{ia},PID_i,x_i,pk_i^e,Auth_1) = \mathrm{Dec}_{sym}(K_{1e}',M_1)$，验证用户认证标签 $Auth_1 = h(M_0 \mathbin{//} X_{ia} \mathbin{//} PID_i \mathbin{//} x_i \mathbin{//} pk_i^e \mathbin{//} K_1' \mathbin{//} T_1)$，若验证通过，则验证 $X_{ia} = h(PID_i \mathbin{//} PSR_a)$，$M_0 = h(SID_a \mathbin{//} PSR_a)$，若验证通过，服务器 S_a 生成随机数 y_a 和时间戳 T_2，使用用户临时公钥 pk_i^e 执行 $(c_2,K_2) \leftarrow \mathrm{Encaps}(pk_i^e)$，计算主会话密钥 $MK_{i,a} = KDF(PID_i \mathbin{//} SID_a \mathbin{//} x_i \mathbin{//} y_a \mathbin{//} K_1' \mathbin{//} K_2)$，派生业务会话密钥 $SK_{i,a} = KDF(MK_{i,a} \mathbin{//} \text{“session”})$，计算认证标签 $Auth_2 = h(MK_{i,a} \mathbin{//} c_2 \mathbin{//} y_a \mathbin{//} T_2)$。服务器 S_a 向用户 U_i 返回响应消息 $Msg_2 = (c_2,y_a,Auth_2,T_2)$。

（3）用户处理并生成会话密钥

用户 U_i 收到服务器响应消息 Msg_2 后，首先记录当前时间戳 T_2'，并验证 $|T_2' - T_2| \leqslant \Delta T$，若不成立，则用户终止本次会话；否则，用户使用临时私钥 sk_i^e 对密文 c_2 执行解封装 $K_2' = \mathrm{Decaps}(sk_i^e,c_2)$，计算主会话密钥 $MK_{i,a}' = KDF(PID_i \mathbin{//} SID_a \mathbin{//} x_i \mathbin{//} y_a \mathbin{//} K_1 \mathbin{//} K_2')$，验证服务器认证标签 $Auth_2 = h(MK_{i,a}' \mathbin{//} c_2 \mathbin{//} y_a \mathbin{//} T_2)$，若验证通过，则派生业务会话密钥 $SK_{i,a}' = KDF(MK_{i,a}' \mathbin{//} \text{“session”})$。

3.5　快速切换认证阶段

当用户 U_i 已经与当前服务器 S_a 完成初始认证，并需要切换到目标服务器 S_b 时，执行快速切换认证阶段。具体过程如图3所示。在快速切换认证开始前，用户终端在当前会话期间临时保存 HPW_i' 和 $MK_{i,a}$。为了便于描述，定义如下中间变量：$\alpha = PID_i \mathbin{//} SID_a \mathbin{//} SID_b \mathbin{//} n_u$，$\alpha' = PID_i' \mathbin{//} SID_a \mathbin{//} SID_b \mathbin{//} n_u$，$\beta = \alpha \mathbin{//} n_a$，$\beta' = \alpha' \mathbin{//} n_a$，$\gamma = \beta \mathbin{//} n_b$，$\gamma' = \beta' \mathbin{//} n_b$，其中 n_u，n_a，n_b 分别为用户 U_i，服务器 S_a，服务器 S_b 生成的随机数。

（1）用户向当前服务器发起切换请求

用户 U_i 准备从当前服务器 S_a 切换到目标服务器 S_b。用户终端首先生成随机数 n_u 和时间戳 T_4，用户从本地存储中取出与目标服务器 S_b 对应的绑定值 V_{ib}，并计算 $X_{ib} = HPW_i' \oplus V_{ib}$，$\alpha = PID_i \mathbin{//} SID_a \mathbin{//} SID_b \mathbin{//} n_u$，$Auth_a = h(MK_{i,a} \mathbin{//} \alpha \mathbin{//} X_{ib} \mathbin{//} T_4)$。用户 U_i 向当前服务器 S_a 发送消息 $Msg_1 = (SID_b,n_u,X_{ib},Auth_a,T_4)$。

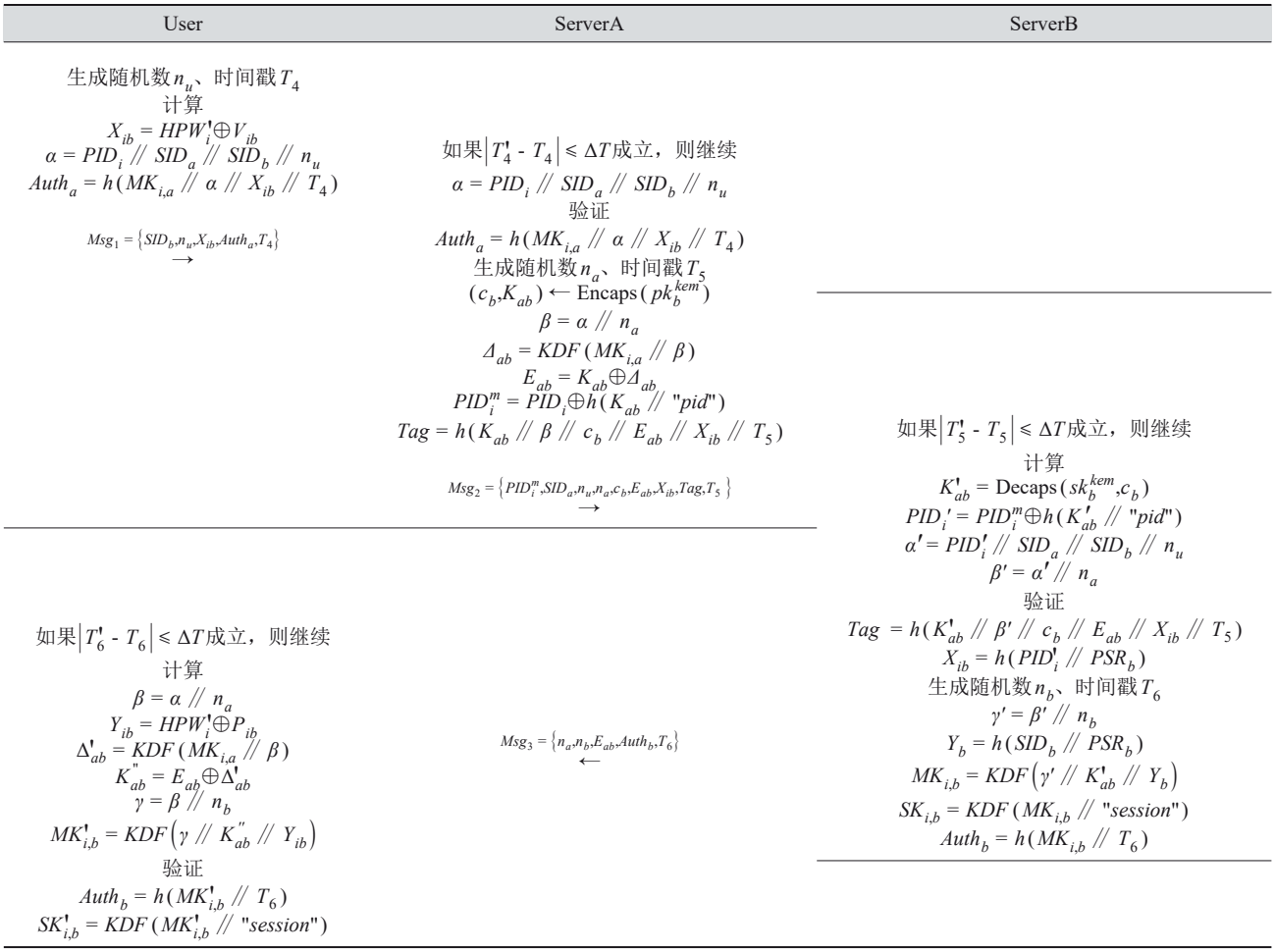
（2）当前服务器向目标服务器转发切换材料

服务器 S_a 收到用户发送的切换请求后，首先记录当前时间戳 T_4'，若 $|T_4' - T_4| \leqslant \Delta T$ 不成立，则终止本次切换请求；否则服务器 S_a 计算 $\alpha = PID_i \mathbin{//} SID_a \mathbin{//} SID_b \mathbin{//} n_u$，验证 $Auth_a = h(MK_{i,a} \mathbin{//} \alpha \mathbin{//} X_{ib} \mathbin{//} T_4)$，若成立则服务器 S_a 生成随机数 n_a 和时间戳 T_5，执行封装算法 $(c_b,K_{ab}) \leftarrow \mathrm{Encaps}(pk_b^{kem})$，计算 $\beta = \alpha \mathbin{//} n_a$，生成辅助掩码 $\Delta_{ab} = KDF(MK_{i,a} \mathbin{//} \beta)$，生成辅助值 $E_{ab} = K_{ab} \oplus \Delta_{ab}$，计算掩码假名 $PID_i^m = PID_i \oplus h(K_{ab} \mathbin{//} \text{“pid”})$，生成完整性标签 $Tag = h(K_{ab} \mathbin{//} \beta \mathbin{//} c_b \mathbin{//} E_{ab} \mathbin{//} X_{ib} \mathbin{//} T_5)$。当前服务器 S_a 向目标服务器 S_b 发送消息 $Msg_2 = (PID_i^m,SID_a,n_u,n_a,c_b,E_{ab},X_{ib},Tag,T_5)$。

（3）目标服务器验证并响应用户

目标服务器 S_b 收到服务器 S_a 发送的消息后，首先记录当前时间戳 T_5'，若 $|T_5' - T_5| \leqslant \Delta T$ 不成立，则终止本次切换认证；否则 S_b 使用KEM私钥 sk_b^{kem} 对密文 c_b 执行解封装 $K_{ab}' = \mathrm{Decaps}(sk_b^{kem},c_b)$，恢复

图3 快速切换认证与密钥协商阶段



假名 $PID_i' = PID_i^m \oplus h(K_{ab}' \,//\, \text{“}pid\text{”})$，计算 $\alpha' = PID_i' \,//\, SID_a \,//\, SID_b \,//\, n_u$，$\beta' = \alpha' \,//\, n_a$，验证 $Tag = h(K_{ab}' \,//\, \beta' \,//\, c_b \,//\, E_{ab} \,//\, X_{ib} \,//\, T_5)$，若成立则 S_b 验证 $X_{ib} = h(PID_i' \,//\, PSR_b)$，若成立，目标服务器 S_b 生成随机数 n_b 和时间戳 T_6，计算 $\gamma' = \beta' \,//\, n_b$，$Y_b = h(SID_b \,//\, PSR_b)$，生成主会话密钥 $MK_{i,b} = KDF(\gamma' \,//\, K_{ab}' \,//\, Y_b)$，派生业务会话密钥 $SK_{i,b} = KDF(MK_{i,b} \,//\, \text{“}session\text{”})$，计算 $Auth_b = h(MK_{i,b} \,//\, T_6)$。目标服务器 S_b 向用户 U_i 发送响应消息 $Msg_3 = (n_a, n_b, E_{ab}, Auth_b, T_6)$。

（4）用户验证目标服务器并生成新会话密钥

用户 U_i 收到目标服务器 S_b 发送的消息后，首先记录当前时间戳 T_6'，若 $|T_6' - T_6| \le \Delta T$ 不成立，则终止本次切换认证；否则用户根据先前保存的 α 和收到的 n_a 计算 $\beta = \alpha \,//\, n_a$，$Y_{ib} = HPW_i' \oplus P_{ib}$，计算辅助掩码 $\Delta_{ab}' = KDF(MK_{i,a} \,//\, \beta)$，恢复共享密钥 $K_{ab}'' = E_{ab} \oplus \Delta_{ab}'$，计算 $\gamma = \beta \,//\, n_b$，生成主会话密钥 $MK_{i,b}' = KDF(\gamma \,//\, K_{ab}'' \,//\, Y_{ib})$，验证 $Auth_b = h(MK_{i,b}' \,//\, T_6)$，若成立则继续派生业务会话密钥 $SK_{i,b}' = KDF(MK_{i,b}' \,//\, \text{“}session\text{”})$。

4 安全性分析

4.1 威胁模型与安全目标

本文采用 Dolev-Yao 威胁模型[25]描述敌手能力。在该模型下，敌手 $\mathcal{A}$ 可以完全控制公开信道，能够窃听、拦截、篡改、删除、重放和伪造公开信道上传输的消息。敌手可以尝试获取用户终端中存储的部分认证参数，也可以尝试猜测用户口令，但不能同时获得用户口令和足以恢复合法认证秘密的全部本地参数。敌手还可以尝试泄露服务器长期秘密、业务会话密钥以及快速切换认证所依赖的主会话密钥。本文考虑量子计算威胁，并假设协议中采用的 ML-KEM 满足 IND-CCA 安全性，即敌手无法以不可忽略优势区分合法密文对应的共享密钥与等长随机串。

在上述威胁模型下，本文协议应满足以下安全目标：双向认证、会话密钥安全、后量子安全、用户身份匿名性、抗离线口令猜测攻击、抗重放攻击、抗中间人攻击、抗冒充攻击、抗非法快速切换攻击等安全目标。

4.2 ROR 安全模型

本文采用 ROR 模型证明所提协议的会话密钥语义安全性。在 ROR 模型中，敌手 $\mathcal{A}$ 通过执行查询与协议实例交互，并通过 *Test* 查询区分真实会话密钥与等长随机串。若任意概率多项式时间敌手均不能以不可忽略优势完成区分，则称协议在 ROR 模型下满足会话密钥语义安全。

4.2.1 系统参与方与会话实例

系统包含注册中心 RC、用户集合 $\mathcal{U}$ 和服务器集合 $\mathcal{S}$。其中，$U_i \in \mathcal{U}$ 表示第 i 个用户，$S_j \in \mathcal{S}$ 表示第 j 个医疗服务器。设 S_a 表示用户当前已认证服务器，S_b 表示目标切换服务器。

本文采用会话实例描述协议执行过程。记 U_i^p 表示用户 U_i 的第 p 个会话实例，S_j^q 表示服务器 S_j 的第 q 个会话实例。当某个会话实例完成协议执行并通过所有认证检查后，该实例进入接受状态，并输出相应业务会话密钥。初始认证与密钥协商阶段输出用户 U_i 与当前服务器 S_a 之间的业务会话密钥 $SK_{i,a}$；快速切换认证阶段输出用户 U_i 与目标服务器 S_b 之间的业务会话密钥 $SK_{i,b}$。

4.2.2 敌手能力与查询接口

为了刻画被动攻击、主动攻击、口令泄露、本地存储泄露以及会话密钥语义安全性，本文为敌手提供如下查询接口。

Execute(U_i^p, S_a^q)：用于模拟用户 U_i 与当前服务器 S_a 之间一次诚实执行的初始认证过程。该查询返回初始认证阶段的公开通信消息。

Execute(U_i^p, S_a^q, S_b^r)：用于模拟用户 U_i、当前服务器 S_a 和目标服务器 S_b 之间一次诚实执行的快速切换认证过程。该查询返回快速切换阶段的公开通信消息。

Send(Π, m)：用于模拟主动攻击，其中 Π 表示某个用户实例或服务器实例，m 表示敌手发送给该实例的任意消息。敌手可以通过 Send 查询模拟消息伪造、中间人攻击、重放攻击和篡改攻击等。

Corrupt$(U_i, 0)$：表示敌手获得用户 U_i 的身份和口令。

Corrupt$(U_i, 1)$：表示敌手获得用户终端中存储的长期信息。

Corrupt(S_j)：表示敌手获得服务器 S_j 的长期秘密信息。

Test(Π)：用于定义会话密钥的语义安全性。敌手只能对已经接受且满足新鲜性要求的会话实例发起 Test 查询。实验开始时随机选择比特 $b \in \{0,1\}$。若 $b = 1$，则 Test 查询返回该实例的真实会话密钥；若 $b = 0$，则返回一个与真实会话密钥等长的均匀随机串。敌手最终输出猜测比特 b'。若 $b' = b$，则称敌手成功。

Reveal(Π)：用于模拟会话密钥泄露攻击。若会话实例 Π 已经接受并生成业务会话密钥，则该查询返回对应业务会话密钥；否则返回 $\perp$。敌手不能对被执行 Test 查询的会话及其匹配会话执行 Reveal 查询。

4.2.3 新鲜会话

新鲜会话用于限定 Test 查询的合法对象。若敌手已经获得足以恢复某会话密钥的全部秘密信息，则该会话不再作为 Test 查询对象。本文将完成认证并输出业务会话密钥的会话称为新鲜会话，当且仅当其满足以下条件：

（1）敌手未对该会话或其匹配会话执行 Reveal 查询，即未获得对应会话密钥；

（2）敌手未同时获得用户 U_i 的口令因子和本地存储因子；

（3）敌手未在会话接受前腐化参与该会话的目标服务器，即未获得其 ML-KEM 私钥和服务器长期隐藏值；

（4）敌手未获得足以恢复该会话密钥的临时 KEM 私钥、临时随机数或相关中间密钥材料。

4.2.4 ROR 安全定义

设 $\mathcal{P}$ 表示本文提出的后量子口令认证与快速切换认证协议，$\mathcal{A}$ 表示概率多项式时间敌手。令 $Exp_{\mathcal{P},\mathcal{A}}^{ROR}(\lambda)$ 表示如下 ROR 安全实验。

首先，挑战者根据安全参数 λ 执行系统初始化，并生成系统公共参数。随后，敌手 $\mathcal{A}$ 可以自适应发起 Execute、Send、Corrupt、Test、Reveal 和 Hash 查询。对于 Test 查询，敌手只能选择满足第 4.2.3 节新鲜性要求的会话实例。挑战者随机选择隐藏比特 $b \in \{0,1\}$：若 $b = 1$，则向敌手返回该会话的真实业务会话密钥；若 $b = 0$，则返回等长随

机串。最终，敌手输出猜测比特b'。

记事件$Succ$表示敌手成功猜中隐藏比特，即$b'=b$。敌手$\mathcal{A}$攻破协议$\mathcal{P}$的ROR优势定义为：

$$Adv_{\mathcal{A}}^{\mathcal{P}}(\lambda)=|2Pr[Succ]-1| \tag{1}$$

如果对任意概率多项式时间敌手$\mathcal{A}$，其优势$Adv_{\mathcal{A}}^{\mathcal{P}}(\lambda)$均为可忽略函数，则称协议$\mathcal{P}$在ROR模型下满足会话密钥语义安全性。

对于本文协议而言，会话密钥语义安全性具体包括初始认证阶段生成的$SK_{i,a}$与等长随机串不可区分；快速切换认证阶段生成的$SK_{i,b}$与等长随机串不可区分。

因此，若敌手无法在ROR实验中有效区分真实会话密钥与随机串，则说明本文协议能够在初始认证和快速切换认证两个阶段保证会话密钥安全。

4.3 安全性定理

定理1 设$\mathcal{A}$是一个运行时间为t的概率多项式时间敌手。令q_h表示Hash查询次数，q_s表示Send查询次数，q_e^{init}表示初始认证阶段Execute查询次数，q_e^{sw}表示快速切换认证阶段Execute查询次数。令l_h表示哈希函数输出长度，l_K表示ML-KEM共享密钥长度，l_R表示注册中心系统根秘密K_{RC}的长度，$l_S=min(l_R,l_h)$表示服务器长期隐藏值PSR_j的有效安全强度。$|\mathcal{D}_{pw}|$表示用户口令字典大小。设$q_K=2q_e^{init}+q_e^{sw}+2q_s$。其中，$q_K$表示最多ML-KEM共享密钥数量。若ML-KEM满足IND-CCA安全性，且哈希函数与密钥派生函数均建模为随机预言机，则敌手$\mathcal{A}$攻破本文协议$\mathcal{P}$的优势满足：

$$Adv_{\mathcal{A}}^{\mathcal{P}}(t)\leqslant 2q_K\cdot Adv_{ML-KEM}^{cca}(\mathcal{A})+\frac{q_h^2}{2^{l_h}}+\frac{q_K^2}{2^{l_K}}+\frac{2q_s}{|\mathcal{D}_{pw}|}+\frac{4q_s}{2^{l_S}}+\frac{10q_s}{2^{l_h}} \tag{2}$$

其中$Adv_{ML-KEM}^{cca}(\mathcal{A})$表示敌手区分ML-KEM封装共享密钥与随机值的优势。由于ML-KEM满足后量子IND-CCA安全性，且l_h，l_K，l_S足够大，所以$Adv_{\mathcal{A}}^{\mathcal{P}}(t)$为可忽略量。

4.4 游戏序列证明

下面通过一系列游戏证明定理成立。记$Game_i$表示第i个游戏，$Succ_i$表示敌手$\mathcal{A}$在$Game_i$中成功猜中Test查询隐藏比特b的事件。

Game 0：真实攻击游戏。在该游戏中，所有用户、服务器按照协议诚实执行。初始认证阶段，用户U_i和服务器S_a通过两次ML-KEM操作得到K_1和K_2，并生成$MK_{i,a}$和$SK_{i,a}$。快速切换认证阶段，当前服务器S_a与目标服务器S_b通过ML-KEM得到快速切换共享密钥K_{ab}，用户通过$K_{ab}^{''}$恢复该共享密钥，并进一步生成$MK_{i,b}$和$SK_{i,b}$。根据ROR安全定义，有：

$$Adv_{\mathcal{A}}^{\mathcal{P}}(t)=|2Pr[Succ_0]-1| \tag{3}$$

Game 1：替换ML-KEM共享密钥。$Game_1$与$Game_0$唯一差别是模拟器将协议中由ML-KEM封装得到的共享密钥替换为等长均匀随机串。在初始认证阶段，将K_1，K_2分别替换为等长随机串K_1^R，K_2^R。在快速切换认证阶段，将K_{ab}替换为等长随机串K_{ab}^R。若敌手能够区分$Game_0$和$Game_1$，则可以构造一个算法利用敌手区分ML-KEM封装共享密钥与随机串，从而破坏ML-KEM的IND-CCA安全性。因此：

$$|Pr[Succ_1]-Pr[Succ_0]|\leqslant q_K\cdot Adv_{ML-KEM}^{cca}(\mathcal{A}) \tag{4}$$

Game 2：处理随机预言机碰撞和密钥碰撞。$Game_2$与$Game_1$的差别在于，若出现以下任一事件，模拟器立即中止游戏：

（1）敌手在随机预言机查询中得到哈希输出碰撞；

（2）实验中随机替换得到的ML-KEM共享密钥出现碰撞。

由于随机预言机输出长度为l_h，敌手最多发起q_h次Hash查询，根据生日悖论，发生哈希碰撞的概率至多为$\frac{q_h^2}{2^{l_h+1}}$；并且ML-KEM共享密钥长度为l_K，实验中最多涉及q_K个共享密钥，因此随机共享密钥发生碰撞的概率至多为$\frac{q_K^2}{2^{l_K+1}}$。所以：

$$|Pr[Succ_2]-Pr[Succ_1]|\leqslant\frac{q_h^2}{2^{l_h+1}}+\frac{q_K^2}{2^{l_K+1}} \tag{5}$$

Game 3：拒绝伪造的Send查询。$Game_3$与$Game_2$不同之处在于，模拟器拒绝所有不能通过认证验证的Send查询。下面证明，在$Game_2$中，敌手伪造Send查询并通过验证的概率本身就是可忽略的。

（1）伪造初始认证请求Msg_1。敌手若伪造合法Msg_1，必须同时满足$Auth_1$、X_{ia}和M_0的验证。若敌手试图通过猜测PSR_a或注册中心根秘密K_{RC}来生成正确的X_{ia}和M_0，其成功概率至多为$\frac{1}{2^{l_S}}$。若

敌手通过猜测口令计算HPW_i，进而从V_{ia}和P_{ia}中恢复X_{ia}和M_0，其成功概率至多为$\frac{1}{|\mathcal{D}_{pw}|}$。若敌手不掌握秘密参数的情况下直接猜测$Auth_1$，其成功概率至多为$\frac{1}{2^{l_h}}$。因此，敌手伪造初始认证请求并通过验证的概率至多为：$\frac{1}{|\mathcal{D}_{pw}|}+\frac{1}{2^{l_s}}+\frac{1}{2^{l_h}}$。

（2）伪造初始认证响应Msg_2。敌手必须生成正确的$Auth_2$才能通过验证。然而，$Auth_2$中的$MK'_{i,a}$依赖于K_1和K'_2。在ML-KEM的IND-CCA安全性下，敌手不能从c_1中恢复K_1，也无法从临时KEM私钥获得正确K'_2。因此，敌手只能猜测$Auth_2$，其成功概率至多为$\frac{1}{2^{l_h}}$。

（3）伪造快速切换请求Msg_1。敌手伪造消息时需要验证$Auth_a$，由于$MK_{i,a}$是用户U_i与服务器S_a的主会话密钥，且新鲜性要求敌手不能获得该会话主密钥。若敌手试图伪造，则只能猜测哈希值，其成功概率至多为$\frac{1}{2^{l_h}}$。

（4）伪造当前服务器到目标服务器的切换认证材料Msg_2。若敌手不知道目标服务器S_b的长期隐藏值PSR_b，则无法计算正确的X_{ib}。若敌手不知道K'_{ab}，也无法计算正确的Tag。虽然敌手可以自己选择c_b并尝试构造某些字段，但若无法同时使PID'_i、X_{ib}和Tag与S_b的PSR_b保持一致，则认证无法通过。因此，敌手伪造该消息并通过验证的概率至多为$\frac{1}{2^{l_s}}+\frac{1}{2^{l_h}}$。

（5）伪造目标服务器响应Msg_3。此时，敌手需要利用当前主会话密钥$MK_{i,a}$恢复$K''_{ab}=E_{ab}\oplus KDF(MK_{i,a}\,//\,\beta)$，并计算$MK'_{i,b}=KDF(\gamma\,//\,K''_{ab}\,//\,Y_{ib})$，随后验证$Auth_b=h(MK'_{i,b}\,//\,T_6)$。由于敌手无法获得新鲜会话中的$MK_{i,a}$，也无法获得目标服务器相关值$Y_{ib}=HPW'_i\oplus P_{ib}=h(SID_b\,//\,PSR_b)$，因此敌手无法计算正确的$MK'_{i,b}$，从而无法生成正确的$Auth_b$。其成功概率至多为$\frac{1}{2^{l_h}}$。

综合以上，在q_s次Send查询后，敌手成功伪造并通过验证的概率至多为

$$q_s\left(\frac{1}{|\mathcal{D}_{pw}|}+\frac{2}{2^{l_s}}+\frac{5}{2^{l_h}}\right) \tag{6}$$

因此：

$$|Pr[Succ_3]-Pr[Succ_2]|\leqslant q_s\left(\frac{1}{|\mathcal{D}_{pw}|}+\frac{2}{2^{l_s}}+\frac{5}{2^{l_h}}\right) \tag{7}$$

Game 4：理想随机密钥游戏。$Game_4$与$Game_3$不同在于，模拟器对所有新鲜会话的会话密钥直接返回均匀随机串。

由$Game_1$至$Game_3$可知，在未发生ML-KEM区分事件、随机预言机碰撞事件以及Send伪造成功事件的条件下，敌手无法获得新鲜会话密钥的派生输入。对于任意新鲜的会话密钥$SK_{i,a}$和$SK_{i,b}$，在敌手视图中与均匀随机串不可区分。

由于KDF被建模为随机预言机，对敌手未知输入的输出本身就是等长均匀随机串，因此$Game_4$与$Game_3$在敌手视图中不可区分，且不引入新的优势差异，即：

$$|Pr[Succ_4]-Pr[Succ_3]|=0 \tag{8}$$

且在$Game_4$中，Test查询返回真实会话密钥还是随机串对敌手而言没有区别，因此敌手最优策略只能随机猜测，则有$Pr[Succ_4]=\frac{1}{2}$。

综上所述：

$$Adv^{\mathcal{P}}_{\mathcal{A}}(t)\&\leqslant 2q_K\cdot Adv^{cca}_{ML\text{-}KEM}(\mathcal{A})+\frac{q_h^2}{2^{l_h}}+\frac{q_K^2}{2^{l_K}}+\frac{2q_s}{|\mathcal{D}_{pw}|}+\frac{4q_s}{2^{l_s}}+\frac{10q_s}{2^{l_h}} \tag{9}$$

由于ML-KEM满足IND-CCA安全性，因此上述优势为可忽略量。故本文协议在ROR模型下满足会话密钥语义安全性。

4.5　已知攻击分析

本节在形式化安全证明的基础上，对本文协议能够抵抗的常见攻击进行分析。

（1）双向认证性。在初始认证阶段，服务器通过验证$Auth_1$、X_{ia}和M_0确认用户身份，用户通过验证$Auth_2$确认服务器身份。在快速切换认证阶段，当前服务器通过$Auth_a$验证切换请求来源，目标服务器通过Tag和X_{ib}验证切换认证材料，用户通过$Auth_b$确认目标服务器身份。因此，本文协议能够实现双向认证。

（2）用户匿名性。协议公开信道中不直接传输用户真实身份ID_i。初始认证阶段，用户假名PID_i

被封装在加密负载中；快速切换认证阶段，用户假名以掩码形式PID_i^m传输。敌手无法由公开消息恢复用户真实身份，因此协议能够实现用户匿名性。

（3）抗离线口令猜测与本地存储泄露攻击。用户口令不直接存储或传输，而是与用户身份、随机盐值等信息共同参与派生。敌手即使获得公开通信消息或用户终端中的部分存储信息，也无法在不知道服务器长期隐藏值和相关会话秘密的情况下离线验证候选口令是否正确。因此，协议能够抵抗离线口令猜测攻击和本地存储泄露攻击。

（4）抗重放、中间人与篡改攻击。协议引入时间戳、随机数和认证标签，并将用户假名、服务器身份、随机数和切换上下文绑定。敌手若重放旧消息，将无法通过时间戳和新鲜性验证；若篡改通信消息，接收方重新计算的认证值将与接收值不一致。因此，协议能够抵抗重放攻击、中间人攻击和消息篡改攻击。

（5）抗实体冒充攻击。敌手若冒充合法用户，需要生成能够通过服务器验证的认证请求，而该请求依赖口令派生值、用户与服务器绑定值和会话共享密钥等秘密信息。敌手若冒充服务器，则需要生成合法的$Auth_2$或$Auth_b$，而这些认证值依赖服务器侧解封装得到的共享密钥、服务器长期隐藏值或快速切换共享密钥。由于敌手无法从公开消息中获得上述秘密信息，因此无法成功冒充合法用户或服务器。

（6）抗非法快速切换攻击。快速切换认证要求用户、当前服务器和目标服务器共同参与。敌手若绕过当前服务器直接向目标服务器发起切换认证，需要同时构造合法的用户认证材料、目标服务器绑定值和完整性标签；若伪造当前服务器转发的切换材料，则需要获得用户与当前服务器之间的主会话密钥$MK_{i,a}$或相关切换状态。由于这些信息在新鲜会话中对敌手保持秘密，敌手无法构造能够通过目标服务器验证的切换认证消息。

综上，本文协议能够实现双向认证和会话密钥协商，并抵抗上述常见攻击。

5 性能分析

5.1 功能特征对比

为全面评价本文方案的性能，本文选取4个具有代表性的认证方案作为对比对象，包括多服务器电子医疗认证的Barman方案、多服务器自适应切换认证的汪文明方案、后量子多服务器认证的Wen方案、后量子IoMT多因素认证的Chen方案。

表2给出了本文方案与对比方案的功能特征。其中，“√”表示支持该功能，“×”表示不支持该功能，“—”表示该方案不涉及相应场景或无法直接比较。

表2 功能特征对比

功能特征	Barman	汪文明	Wen	Chen	本文
后量子安全	×	×	√	√	√
智慧医疗场景	√	√	×	√	√
多服务器支持	√	√	√	×	√
离线注册中心	√	√	√	—	√
口令认证	√	√	√	√	√
快速切换认证	×	√	×	×	√
会话密钥协商	√	√	√	√	√
用户匿名性	√	√	√	√	√

由表2可知，现有方案尚未同时兼顾后量子安全与快速切换认证。汪文明方案虽支持多服务器快速切换认证，但仍基于传统密码机制，缺乏后量子安全性；Wen和Chen方案虽具备后量子安全特征，但未面向多服务器连续访问设计快速切换认证机制。本文方案在保证后量子安全的同时通过快速切换认证机制减少跨服务器访问中的重复完整认证开销，更适用于智慧医疗多服务器连续访问场景。

5.2 效率分析

本节通过对比方案的计算开销和通信开销，分析各个方案的效率。考虑到方案在单服务器、多服务器之间的差异，分为两个场景进行讨论。对于支持快速切换认证的方案，访问n个服务器的总计算开销为$Comp(n) = Comp_{init} + (n - 1)Comp_{sw}$，总通信开销为$Comm(n) = Comm_{init} + (n - 1)Comm_{sw}$。对于不支持快速切换认证的方案，每次访问重新执行一次完整认证。对于传统方案采用其原文报告的计算与通信开销作为参考基线，并统一换算单位。

5.2.1 计算开销

本文选择核心密码原语作为计算开销的对象，比如哈希运算、密钥派生运算、对称加解密运算、ML-KEM密钥生成、ML-KEM封装、ML-KEM解封装等，而忽略异或、拼接等开销非常小的运算。

同时，明确方案评估的软硬件环境为：CPU 为 2.60 GHz Intel(R) Core(TM) i7-9750H 处理器，内存 16GB，操作系统为 Windows 11 Version 25H2 64 位。明确基本操作及其执行时间如表 3 所示。

表 3　　运算符号、含义及时间

符号	含义	时间/s	符号	含义	时间/s
T_H	哈希或$KDF(\cdot)$	0.0004	$T_{SE/D}$	对称加密或解密	0.0030
T_{Ke}	ML-KEM 密钥生成	0.0038	T_{En}	ML-KEM 封装	0.0051
T_{De}	ML-KEM 解封装	0.0063	T_{PUF}	PUF	0.0008
T_{Fe}	模糊提取器	0.0421	T_s	采样	0.0168
T_{PM}	多项式乘法	0.0035	T_{PAS}	多项式加减	0.0013
T_{rec}	Rec	0.0016	T_{hel}	HelpRec	0.0019

（1）单服务器初始认证场景

在初始认证场景下，本文方案执行 KDF 运算、ML-KEM 密钥生成、ML-KEM 封装与解封装、对称加解密以及哈希运算，计算开销为 $Comp_{init} = 13T_H + T_{Ke} + 2T_{En} + 2T_{De} + 2T_{SE/D} = 0.0378s$。

对比方案中，Barman 方案计算开销为 0.00225s；汪文明方案计算开销为 0.00213s；Wen 方案计算开销为 $12T_H + T_{Ke} + 2T_{En} + 2T_{De} + T_{Fe} = 0.0735s$；Chen 方案计算开销为 $10T_H + 4T_s + 2T_{PAS} + 12T_{PM} + 4T_{rec} + 2T_{hel} = 0.1260s$。

所有方案单次计算开销柱状对比如图 4 所示。

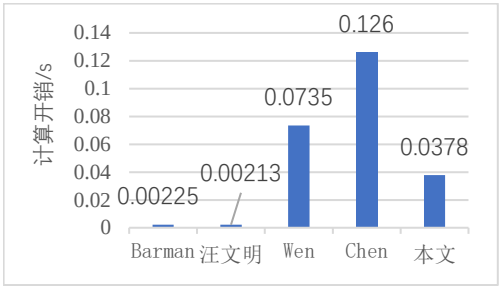


图 4　单次计算开销对比

（2）多服务器切换认证场景

快速切换认证场景下，本文方案执行 ML-KEM 封装、ML-KEM 解封装以及哈希运算，计算开销为 $Comp_{sw} = 16T_H + T_{En} + T_{De} = 0.0178s$。连续访问 5 个服务器时 Barman 方案总计算开销为 0.01125s；汪文明方案总计算开销为 0.00710s；Wen 方案总计算开销为 0.3675s；Chen 方案总计算开销为 0.6300s。从连续访问 2 个服务器即 $n = 2$ 开始，本文方案的总计算开销与对比方案中的后量子方案的差距持续拉大，当用户连续访问 5 个服务器时，本文方案的总计算开销为 0.1090s。

5.2.2　通信开销

通信开销方面，本文仅统计认证与密钥协商阶段在公开信道上传输的消息长度。为了公平对比，本文统一规定如下参数长度：哈希值或认证标签长度为 160bit，身份标识长度为 40bit，随机数长度为 40bit，时间戳长度为 32bit，ML-KEM 密文长度为 8704bit，ML-KEM 公钥长度为 9472bit，ML-KEM 共享密钥长度为 256bit。本文快速切换认证阶段中的辅助值 E_{ab} 和共享密钥 K_{ab} 为 256bit。

（1）单服务器初始认证场景

本文初始认证消息为 (c_1, M_1, T_1) 与 $(c_2, y_a, Auth_2, T_2)$，其中 c_1、c_2 为 8704bit，M_1 为 10152bit，$Auth_2$ 为 160bit，T_1、T_2 为 32bit。因此本文初始阶段通信开销 $Comm_{init}$ 为 27824bit。

同理，Barman 方案的通信开销为 1116bit；汪文明方案的通信开销为 1632bit；Wen 方案的通信开销为 27616bit；Chen 方案通信开销为 19328bit。各方案单次通信开销对比如图 5 所示。

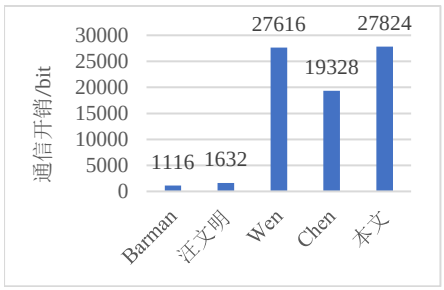


图 5　单次通信开销对比

（2）多服务器切换认证场景

在快速切换认证场景下，本文快速切换认证的通信开销 $Comm_{sw}$ 为 10552bit。连续访问 5 个服务器时 Barman 方案总通信开销为 5580bit；汪文明方案总通信开销为 6208bit；Wen 方案总通信开销为 138080bit；Chen 方案总通信开销为 96640bit。从连续访问 2 个服务器即 $n = 2$ 开始，本文方案通信开销全面低于对比方案中的后量子方案。当用户连续访问 5 个服务器时，本文方案的总通信开销为 70032bit。

5.3　综合性能分析

表 4 给出了本文方案与对比方案在单次认证、

切换访问5个服务器两种场景下在计算开销与通信开销方面的对比情况。

表4　　开销对比

方案	计算开销/s	通信开销/bit	连续访问5个服务器	
			总计算开销/s	总通信开销/bit
Barman	0.00225	1116	0.01125	5580
汪文明	0.00213	1632	0.00710	6208
Wen	0.0735	27616	0.3675	138080
Chen	0.1260	19328	0.6300	96640
本文	0.0378	27824	0.1090	70032

由表4可知， Barman和汪文明这两个传统方案主要采用轻量级密码操作，未引入后量子密码机制，因此在单次认证和连续访问5个服务器时均具有较低开销。但二者不具备后量子安全性，其低开销结果不能直接代表后量子安全场景下的认证效率。相比之下，本文方案采用ML-KEM构造后量子口令认证与密钥协商机制，单次初始认证计算开销为0.0378 s，低于Wen方案的0.0735s和Chen方案的0.1260 s；单次通信开销为27824bit，与Wen方案接近，但高于Chen方案。其原因在于，本文方案在初始认证阶段不仅完成身份认证和密钥协商，还建立了支持后续快速切换认证的主会话密钥和认证状态。

在多服务器连续访问场景下，上述初始认证状态能够转化为明显的累计开销优势。Wen方案和Chen方案不支持快速切换认证，用户连续访问5个服务器时需要重复执行5次完整认证；本文方案仅需执行1次初始认证和4次快速切换认证，总计算开销为0.1090s，总通信开销为70032bit。相比Wen方案，本文方案的总计算开销和总通信开销分别降低约70.34%和49.28%；相比Chen方案，分别降低约82.70%和27.53%。

因此，本文方案的效率优势主要体现在后量子多服务器连续访问场景中。与传统轻量级方案相比，本文方案因引入后量子安全机制产生一定额外开销；但与现有后量子认证方案相比，本文通过初始认证加快速切换认证的机制减少了跨服务器访问中的重复完整认证过程，在保证后量子安全的同时降低了累计计算开销和通信开销，体现了较好的多服务器适用性和可扩展性。

5.4　存储开销分析

本文方案中用户端长期存储数据为$\{a_i, PID_i, \mathcal{D}, (SID_j, V_{ij}, P_{ij})_{S_j \in \mathcal{D}}\}$，n个服务器下用户端长期存储开销$S_U = 160 + 160 + n(40 + 9472 + 320) = (320 + 9832n)$bit，当有5个服务器即n=5时，用户端长期存储开销为6.04KB，而每增加一个服务器，用户端长期存储开销仅增加1.2KB，因此本文方案在实际场景中的部署成本较低。

6　结束语

本文面向智慧医疗多服务器连续访问场景，提出一种基于ML-KEM的后量子口令认证与快速切换密钥协商协议。协议通过初始认证加快速切换认证机制，避免重复执行完整认证。安全性分析表明，协议在ROR模型下满足会话密钥语义安全，并能够抵抗离线口令猜测、重放、中间人、冒充和非法快速切换等攻击。性能分析表明，协议在连续访问多个服务器时能够有效降低累计计算开销和通信开销。后续将进一步研究用户不可追踪性、动态撤销机制和原型实现。

参考文献：

[1] SHOR P W. Algorithms for quantum computation: discrete logarithms and factoring[C]//Proceedings of the 35th Annual Symposium on Foundations of Computer Science. Santa Fe, USA, 20-22 November 1994. Los Alamitos: IEEE Computer Society, 1994: 124-134. DOI: 10.1109/SFCS.1994.365700.

[2] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. Module-Lattice-Based Key-Encapsulation Mechanism Standard: FIPS 203[S]. Gaithersburg: NIST, 2024. DOI: 10.6028/NIST.FIPS.203.

[3] CANTOR S, KEMP J, PHILPOTT R, et al. Assertions and protocols for the OASIS Security Assertion Markup Language (SAML) V2.0[S]. OASIS Standard, 2005.

[4] HARDT D. The OAuth 2.0 Authorization Framework: RFC 6749[S]. Fremont: IETF, 2012.

[5] SAKIMURA N, BRADLEY J, JONES M, et al. OpenID Connect Core 1.0[S]. OpenID Foundation, 2014.

[6] LAMPORT L. Password authentication with insecure communication [J]. Communications of the ACM, 1981, 24(11): 770-772.

[7] BELLOVIN S M, MERRITT M. Encrypted key exchange: password-based protocols secure against dictionary attacks[C]//Proceedings of the IEEE Symposium on Research in Security and Privacy. Oakland, USA, 4-6 May 1992. Los Alamitos: IEEE Computer Society Press, 1992: 72-84. DOI: 10.1109/RISP.1992.213269.

[8] BELLARE M, POINTCHEVAL D, ROGAWAY P. Authenticated key exchange secure against dictionary attacks[C]//Advances in Cryptology—EUROCRYPT 2000. Bruges, Belgium, 14-18 May 2000. Berlin:

Springer, 2000: 139-155.

[9] CHUANG M C, CHEN M C. An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics[J]. Expert Systems with Applications, 2014, 41(4): 1411-1418.

[10] MISHRA D, DAS A K, MUKHOPADHYAY S. A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards[J]. Expert Systems with Applications, 2014, 41(18): 8129-8143.

[11] KUMARI S, LI X, WU F, et al. Design of a provably secure biometrics-based multi-cloud-server authentication scheme[J]. Future Generation Computer Systems, 2017, 68: 320-330.

[12] CHATURVEDI A, DAS A K, MISHRA D, et al. Design of a secure smart card-based multi-server authentication scheme[J]. Journal of Information Security and Applications, 2016, 30: 64-80.

[13] FENG Q, HE D, ZEADALLY S, et al. Anonymous biometrics-based authentication scheme with key distribution for mobile multi-server environment[J]. Future Generation Computer Systems, 2018, 84: 239-251.

[14] BARMAN S, SHUM H P H, CHATTOPADHYAY S, et al. A secure authentication protocol for multi-server-based e-healthcare using a fuzzy commitment scheme[J]. IEEE Access, 2019, 7: 12557-12574.

[15] ALI Z, HUSSAIN S, REHMAN R H U, et al. ITSSAKA-MS: an improved three-factor symmetric-key based secure AKA scheme for multi-server environments[J]. IEEE Access, 2020, 8: 107993-108003.

[16] LUO H, WANG F, XU G. Provably secure ECC-based three-factor authentication scheme for mobile cloud computing with offline registration centre[J]. Wireless Communications and Mobile Computing, 2021, 2021: 8848032. DOI: 10.1155/2021/8848032.

[17] 王雄,李伟麟,王志强,等.无线医疗传感器网络中基于PUF的轻量级匿名认证协议[J].信息安全研究,2025,11(12):1134-1145.
WANG X, LI W L, WANG Z Q, et al. Lightweight PUF-based anonymous authentication protocol for wireless medical sensor networks[J]. Journal of Information Security Research, 2025, 11(12): 1134-1145.

[18] 王雄,李伟麟,张泽昊,等.多网关无线医疗传感器网络中基于PUF的轻量级匿名认证协议[J].计算机应用研究,2025,42(10):3152-3158. DOI:10.19734/j.issn.1001-3695.2024.12.0536.
WANG X, LI W L, ZHANG Z H, et al. Lightweight PUF-based anonymous authentication protocol for multi-gateway wireless medical sensor networks[J]. Application Research of Computers, 2025, 42(10): 3152-3158. DOI: 10.19734/j.issn.1001-3695.2024.12.0536.

[19] 汪文明.多服务器环境下智慧医疗的认证与密钥协商协议研究[D].南京邮电大学,2022.DOI:10.27251/d.cnki.gnjdc.2022.001785.
WANG W M. Research on authentication and key agreement protocols for smart healthcare in multi-server environments[D]. Nanjing: Nanjing University of Posts and Telecommunications, 2022. DOI: 10.27251/d.cnki.gnjdc.2022.001785.

[20] BAHACHE A N, CHIKOUCHE N, AKLEYLEK S. Securing cloud-based healthcare applications with a quantum-resistant authentication and key agreement framework[J]. Internet of Things, 2024, 26: 101200. DOI: 10.1016/j.iot.2024.101200.

[21] AHMAD A, JAGATHESWARI S. Quantum safe multi-factor user authentication protocol for cloud-assisted medical IoT[J]. IEEE Access, 2025, 13: 3532-3545. DOI: 10.1109/ACCESS.2024.3523530.

[22] CHEN X, WANG B C, LI H. A privacy-preserving multi-factor authentication scheme for cloud-assisted IoMT with post-quantum security [J]. Journal of Information Security and Applications, 2024, 81: 103708.

[23] PALANISWAMY B, KARATI A. QPTA: quantum-safe privacy-preserving multi-factor authentication scheme for lightweight devices [C]//Proceedings of the 21st International Conference on Security and Cryptography. Dijon, France, 8-10 July 2024. Setúbal: SCITEPRESS, 2024: 804-811. DOI: 10.5220/0012835100003767.

[24] WEN Y, SU Y, LI W. Post-quantum secure multi-factor authentication protocol for multi-server architecture[J]. Entropy, 2025, 27(7): 765.

[25] DOLEV D, YAO A C. On the security of public key protocols[J]. IEEE Transactions on Information Theory, 1983, 29(2): 198-208.

王雄（1977− ），男，北京电子科技学院副教授，主要研究方向为密码协议、密码系统。

张震（2001− ），男，北京电子科技学院硕士生，主要研究方向为身份认证协议。

刘长天（2003− ），男，北京电子科技学院硕士生，主要研究方向为身份认证协议。

郭豪（2003− ），男，北京电子科技学院硕士生，主要研究方向为身份认证协议。

张迪（2004− ），女，北京电子科技学院硕士生，主要研究方向为身份认证协议。