

面向时序数据共享的后量子轻量化无证书混合签密方案

程冠杰¹, 苏韬龙², 黄俊钦³, 孔令和³

(1. 浙江大学软件学院, 浙江 宁波 315000; 2. 浙江大学计算机科学与技术学院, 浙江 杭州 310000;

3. 上海交通大学计算机学院, 上海 200240)

摘要: 面向物联网(IoT)场景中海量、连续的时序数据共享需求, 传统基于双线性映射或椭圆曲线的无证书签密(CLSC)方案在量子计算威胁下安全性受限, 且在多段数据的重复签密中引入了较高的计算与通信开销。为此, 提出一种面向时序数据共享的后量子轻量化无证书混合签密方案。首先, 设计双向哈希链驱动的对称密钥生成与管理机制, 实现多段连续数据的密钥快速生成与前向/后向安全; 然后, 在模块格框架下构建条件隐私保护的轻量化混合签密体制, 并结合离线/在线分离将部分计算前移, 在实现后量子安全的同时显著降低端侧与链路负担。在随机预言机模型下证明了该方案满足 IND-CCA 机密性与 EUF-CMA 不可伪造性。在资源受限的终端设备上开展原型实现与评测, 对比实验结果显示, 该方案相较代表性对比方案具有显著优势, 端侧整体时延与传输体量平均可降至同类后量子方案的 40% 左右。在长时段、连续时序数据场景下, 该方案优势将进一步扩大, 其能兼顾后量子安全、轻量化与可扩展性要求, 适用于新一代智能 IoT 中的时序数据共享应用。

关键词: 物联网; 时序数据共享; 无证书签密; 后量子密码; 密钥生成

中图分类号: TP309.7

文献标志码: A

DOI: 10.19678/j.issn.1000-3428.0260093

Post-Quantum Lightweight Certificateless Hybrid Signcryption Scheme for Time-Series Data Sharing

CHENG Guanjie¹, SU Taolong², HUANG Junqin³, KONG Linghe³

(1. School of Software Technology, Zhejiang University, Ningbo 315000, Zhejiang, China;

2. School of Computer Science and Technology, Zhejiang University, Hangzhou 310000, Zhejiang, China;

3. School of Computer Science, Shanghai Jiao Tong University, Shanghai 200240, China)

【Abstract】 The demand for continuous time-series data sharing in Internet of Things (IoT) scenarios has been increasing. Traditional Certificateless Signcryption (CLSC) schemes based on bilinear pairings or elliptic curves offer limited security under quantum computing threats and incur high computational and communication overheads when repeatedly signcrypting multiple data segments. To address these issues, this paper proposes a post-quantum lightweight certificateless hybrid signcryption scheme tailored for time-series data sharing. First, a bidirectional hash chain-driven symmetric key generation and management mechanism is designed to enable rapid key generation for multiple consecutive data segments while ensuring forward and backward security. Subsequently, a lightweight hybrid signcryption scheme with conditional privacy protection is constructed using a module lattice framework. Combined with an offline/online separation that shifts part of the computation ahead of time, the scheme achieves post-quantum security while significantly reducing the burden on end devices and communication links. The scheme is proven to satisfy IND-CCA confidentiality and EUF-CMA unforgeability in a random oracle model. Prototype implementation and evaluation on resource-constrained terminal devices show that, the proposed scheme achieves significant advantages: on average, the end-side overall latency and transmission volume can be reduced to approximately 40% of those of comparable post-quantum schemes. The advantages of this scheme become even more pronounced in long-duration continuous time-series data scenarios. It effectively balances post-quantum security, lightweight design, and scalability, making it well-suited for time-series data sharing applications in next-generation intelligent IoT.

【Key words】 Internet of Things (IoT); time-series data sharing; certificateless signcryption; post-quantum cryptography; key generation

基金项目: 国家自然科学基金青年科学基金项目(62502446); 宁波市甬江人才工程(2024A-402-G)。

作者简介: 程冠杰, 男, 研究员、博士, 主研方向为数据安全、多智能体协同; 苏韬龙, 硕士; 黄俊钦, 助理教授、博士; 孔令和, 教授、博士。

收稿日期: 2026-01-19

修回日期: 2026-03-18

E-mail: chengguanjie@zju.edu.cn

0 引言

近年来,大模型与物联网(IoT)技术的快速发展,使得全球连接的终端设备数量呈爆发式增长^[1],智能制造、车联网、医疗健康、智慧城市等应用场景中持续生成海量多源的时序数据,这类数据通常具有生成频率高、结构复杂、隐私敏感等特征,蕴含着丰富的历史与行为信息,是支撑 IoT 智能决策与预测分析的重要基础^[2]。例如,心率仪等穿戴式医疗设备可实时记录用户的健康状况,善加利用可为用户提供量体裁衣的医疗健康服务。但是,另一方面,健康状况也是用户的重要隐私,一旦泄露可能导致严重后果。然而,IoT 终端通常受限于存储与算力资源,需要通过数据共享来实现协同管理与分析决策^[3]。与此同时,数据在采集、传输与共享过程中,易遭受窃取、篡改、接收方身份伪造等攻击^[4-5],因此,如何在资源受限与复杂动态环境下实现高效、可信且可追溯的时序数据安全共享,成为亟待解决的核心问题。

为应对上述挑战,学术界与工业界提出了多种基于加密与认证的安全数据共享机制。其中,签密(Signcryption)技术通过将数字签名与加密操作进行融合,有效降低了通信与计算成本,是一种广受认可的隐私保护手段。在该技术的基础上发展而来的无证书签密(CLSC)方案,进一步消除了传统公钥基础设施(PKI)的证书管理负担,兼顾了安全性与可部署性,成为近年来的一大研究热点。现有的签密方案通常依赖于双线性映射^[6-8]或椭圆曲线密码学(ECC)^[9-11],此类方案的安全性通常建立在经典公钥密码的计算困难性假设之上,例如离散对数问题(DLP)与计算 Diffie-Hellman 问题等,并可在相应安全模型中给出可归约的安全性证明。具体而言,在现有经典计算模型与合理参数选取的前提下,已知算法难以在多项式时间内求解上述问题,因此,攻击者难以从密文中恢复明文或推断用户隐私信息,从而支撑方案的机密性与隐私性目标^[12]。然而,量子计算的发展改变了这一安全性基础:Shor 算法表明^[13],在具备足够规模的容错量子计算机时,DLP 等相关问题可在多项式时间内求解,从而使基于这些假设的方案面临潜在失效风险。基于此,研究与部署能够抵抗量子对手的后量子密码机制,已成为学术界与工业界的重要研究方向。

在上述背景下,研究逐步转向被普遍视为具有后量子安全潜力的格密码学(lattice-based cryptography)。格密码方案的安全性通常可归约

到格上的计算困难问题,例如最短向量问题(SVP)及其近似版本:给定一个格,寻找其中欧几里得范数最小(或近似最小)的非零向量。在现有公开研究中,尚未出现能够在量子计算模型下高效求解这类格困难问题的算法,因此,格密码学被认为是构建后量子安全通信机制的重要候选方案。

围绕上述方向,已有工作提出了多类 CLSC 方案,包括基于误差学习(LWE)^[13-15]、N 次截断多项式环(NTRU)^[16-17]以及模块格(Module Lattice)^[18]等。相关方案通常在所考虑的安全模型下,将机密性、不可伪造性或匿名性等安全目标归约到相应的格困难假设(LWE/Module-LWE 或与 SVP 相关的近似困难性)。然而,这类后量子 CLSC 方案在 IoT 场景中仍面临工程适配问题:一方面,密文尺寸、签名开销与交互/传输开销往往较大;另一方面,密钥更新与管理流程相对复杂。同时,对于大规模时序数据的持续加密与高频上链/传输等负载特征,现有方案在端侧计算与链路带宽方面成本偏高,从而限制了其在资源受限的 IoT 环境中的实用性。

针对上述不足,本文面向 IoT 时序数据安全共享场景,提出一种抗量子的时序数据轻量化安全共享方法,设计基于模块格的无证书混合签密(Certificateless Hybrid Signcryption)框架,实现抗量子、轻量化与高效性之间的平衡。首先,本文提出一种基于双向哈希链的对称密钥生成与管理机制,实现时序数据密钥的动态生成、可追溯更新与高效分发,显著降低在海量数据场景下的密钥维护与同步开销;其次,设计基于模块格的轻量化无证书混合签密算法,利用模块误差学习问题(MLWE)构建抗量子安全基础,并通过离线/在线分离计算机制实现签密效率优化;最后,将混合签密与时序数据结构深度融合,支持对多段数据的聚合签密与解签密操作,从而在保证端到端机密性与认证性的同时,极大提升时序数据传输与共享的可扩展性与能效表现。本文的主要贡献总结如下:

1)构建一种面向时序数据的后量子轻量化安全共享方案,实现从终端数据采集到云端共享的全流程安全保障。

2)提出双向哈希链密钥生成机制,在确保前向与后向安全的同时,简化密钥生命周期管理。

3)设计并形式化证明基于模块格的无证书混合签密算法满足 IND-CCA 机密性与 EUF-CMA 不可伪造性。

4)在多种典型方案下进行实验与对比分析,结果表明,本方案在计算和通信开销方面均优于对比

后量子签密方案,尤其在 IoT 设备侧,将整体时延与传输体量平均降至同类方案的 40% 左右。

1 相关工作

1.1 无证书签密

签密在单一流程中完成签名与加密,相比“先签后加/先加后签”,签密能够显著降低开销与时延^[19]。在此基础上,文献[20]提出了无证书密码学,消除了传统 PKI 的证书管理负担,催生了 CLSC 方向的研究^[6-7,9]。文献[6]面向车联网提出了一种基于双线性映射的 CLSC 方法,将签密过程分为离线和在线两部分,充分利用设备空闲时间的算力资源。类似地,文献[7]面向高速列车的异构通信场景,设计了一种基于双线性映射的 CLSC 方法。虽然基于双线性映射的签密方案具有较高的安全性,但是计算开销大,不适合资源受限的 IoT 设备。为此,文献[9]设计了一种基于 ECC 的 CLSC 方案,支持一对多的车联网高效数据共享。类似地,文献[10]面向医疗物联网设计了一种基于 ECC 的轻量化 CLSC 方案。然而,这些方案通常需要对明文长度进行预设,灵活性受限。针对该问题,研究人员引入混合签密策略,即“对称加密数据+非对称签密会话密钥”,以提升场景适配性与签密效率^[10]。

经典 CLSC 在证书管理与在线时延方面具有优势,但其安全性建立在双线性映射或 ECC 等数论难题之上,难以抵御量子计算威胁。同时,在海量连续的时序数据共享场景下,重复签密与密钥分发会放大通信与管理开销。

1.2 后量子无证书签密

近年来,量子计算的快速发展使得 Shor 算法^[21-22]的实现成为可能,该算法利用强大的并行能力和量子傅里叶变换算法,可将复杂的整数因式分解问题转化为函数周期求解问题,从而将时间复杂度降低到多项式级别。因此,上述依赖于离散对数/因式分解难题的 CLSC 方案将面临根本性威胁^[9-11],推动 CLSC 向格密码学迁移^[13-15]。文献[13]提出了一种面向医疗物联网的基于 LWE 的 CLSC 方案,成功解决了传统 CLSC 无法抵抗量子攻击的问题。然而,该方案计算开销较大。为此,文献[16]设计了一种基于 NTRU 格的签密方案,在效率与密钥存储方面更有优势,但是其安全保障相对有限。文献[18]提出了一种基于模块格的 CLSC 方案,借助数论变换技术在效率和安全性之间获得了更好的平衡。

然而,由于格基体制的参数体量与编码特性,不少后量子 CLSC 仍面临密文尺寸大、通信成本高的

问题,将其直接套用于大规模时序数据时,重复签密与多段密钥派发会叠加带宽与时延开销,成为落地应用的主要瓶颈。针对上述不足,本文在模块格框架下引入双向哈希链的密钥生成机制,并与混合签密进行耦合,在保证端到端机密性与认证性的同时,显著降低多段连续数据的密钥生成、分发与通信负担,从而兼顾后量子安全、轻量化与可扩展性。

2 预备知识

为便于后续描述,本节将介绍模块格以及本方案使用到的随机分布、陷门原像采样算法、困难问题假设以及 ChaCha20-Poly1305 对称加密算法。方案中使用的符号和语义描述如表 1 所示。

表 1 常用符号及其语义描述

Table 1 Common symbols and their semantic descriptions

符号	描述
mpk, msk	系统公钥和私钥
pk_m, sk_m	用户 ID_m 的完整公钥和私钥
δ, ϵ	DO 和 IoT 设备共享的长期密钥
RID/PID	用户的真实身份信息/假名
Tag_c	当前系统时间的标签
DT, T	特定日期和当前时间戳
M_i^{DT}	日期 DT 采集的第 i 段数据明文
K_i^{DT}	M_i^{DT} 对应的加密封钥
C_i^{DT}	M_i^{DT} 加密得到的密文
LK_i^{DT}/RK_i^{DT}	用于生成 K_i^{DT} 的左/右密钥
$CSet_{t_s, t_e}^{DT}$	DT 当天第 $t_s \sim t_e$ 段密文集合
$\mathbb{Z}/\mathbb{Z}_q$	整数环/ $\left[-\frac{q-1}{2}, \frac{q-1}{2}\right]$ 下的整数环
$\mathbb{Z}[x]/\mathbb{Z}_q[x]$	所有系数都来自 $\mathbb{Z}/\mathbb{Z}_q$ 的多项式集合
R/R_q	和的多项式环
I_d	d 维单位矩阵
q, N	模块格的模数和度数
d, k, n, m, l, l'	模块格维度参数
β_{psk}	部分私钥的边界值
n, r, e	中心二项分布参数
$\gamma_1, \gamma_2, \gamma_d$	系数范围
s	高斯参数
B_τ	系数中有 τ 个 1 和 -1 且其余系数为 0 的多项式集合
S_η	$[-\eta, \eta]$ 的整数集合
PT_{key}	对称密钥明文
PT_{data}/CT_{data}	时序数据明文/密文
σ	混合签密密文

2.1 模块格相关定义

对于一个度为 N 的多项式向量 $\mathbf{v} = (v_1, v_2, \dots, v_m)^T \in R_q^m$, 有 $\forall i \in [1, m], v_i = v_{i,0} + v_{i,1}x +$

$v_{i,2}x^2 + \dots + v_{i,N-1}x^{N-1} \in R_q$, 将 $\mathbf{v}$ 的 L_2 范数定义为 $\|\mathbf{v}\|_2 = \sqrt{\sum_{i=1}^m \sum_{j=0}^{N-1} v_{i,j}^2}$, 将其 L_∞ 范数定义为 $\|\mathbf{v}\|_\infty = \max_{1 \leq i \leq m} (\max_{0 \leq j \leq N-1} |v_{i,j}|)$ 。

给定度 N 的 n 个线性独立的向量 $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n \in R_q^m$, 定义一个 $m \times n$ 的多项式矩阵 $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n)$, 由此得到一个模块格 $\Lambda = \mathcal{L}(\mathbf{B}) = \left\{ \mathbf{B}\mathbf{c} = \sum_{i=1}^n \mathbf{b}_i c_i \mid \mathbf{c} \in \mathbb{Z}^n \right\}$, 其中 $\mathbf{c}$ 是一个系数向量, $(\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n)$ 为模块格的一组基。

2.2 随机分布

2.2.1 均匀分布

给定一个集合 S , 本文用 $\mathcal{U}(S)$ 表示该集合上的均匀分布, 用 $u \xleftarrow{\$} S$ 表示从集合 S 中随机且均匀地采样得到 u , 另外, 本文用 $u \leftarrow \chi$ 表示直接从分布 χ 中采样得到 u 。

2.2.2 离散高斯分布

给定一个中心点 $\mathbf{c} \in \mathbb{R}^m$ 、一个标准差参数 $s > 0$, 其中 $\mathbb{R}$ 表示全体实数, 对 $\forall \mathbf{x} \in \mathbb{R}^m$, 定义高斯函数为 $\rho_{\mathbf{c},s}(\mathbf{x}) = \exp(-\pi \|\mathbf{x} - \mathbf{c}\|^2 / s^2)$ 。给定一个模块格 Λ , 对 $\forall \mathbf{x} \in \Lambda$, 将 Λ 上的离散高斯分布定义为 $D_{\Lambda,\mathbf{c},s}(\mathbf{x}) = \frac{\rho_{\mathbf{c},s}(\mathbf{x})}{\rho_{\mathbf{c},s}(\Lambda)}$, 其中 $\rho_{\mathbf{c},s}(\mathbf{x})$ 代表格点 $\mathbf{x}$ 的未归一化概率权重, $\rho_{\mathbf{c},s}(\Lambda) = \sum_{\mathbf{x} \in \Lambda} \rho_{\mathbf{c},s}(\mathbf{x})$ 用于实现该权重的归一化。另外, 当 $\mathbf{c} = \mathbf{0}$ 且 $s = 1$ 时, $D_{\Lambda,\mathbf{c},s}$ 被称为标准离散高斯分布, 可直接记为 D_Λ 。

2.2.3 中心二项分布

给定一个正整数 η , 从中心二项分布 ϕ_η 中采样得到 x 时, 首先采样 $(a_1, a_2, \dots, a_\eta, b_1, b_2, \dots, b_\eta) \xleftarrow{\$} \{0, 1\}^{2\eta}$, 然后计算 $x = \sum_{i=1}^{\eta} (a_i - b_i)$, 显然所得的 $x \in S_\eta$, 即 x 是 $[-\eta, \eta]$ 范围内的一个整数。

2.3 陷门原像采样

给定维度 d 、分解基数 $b \geq 2$ 、与模数 q 有关的分解位数 $k = \left\lceil \log_b \left(\left\lceil \frac{q-1}{2} \right\rceil \right) \right\rceil$, 定义工具矩阵 $\mathbf{G} = \mathbf{I}_d \otimes [1|b|\dots|b^{k-1}] \in \mathbb{Z}^{d \times dk} \subseteq R^{d \times dk}$, 其中 $\mathbf{I}_d$ 是 $d \times d$ 的单位矩阵, $\otimes$ 是克罗内克积运算, 再令 $\mathbf{G}^{-1}(\cdot)$ 为 $\mathbf{G}$ 的结构化逆运算, 它对向量 $\mathbf{v} \in R_q^d$ 的每个分量独立地进行基 b 的分解, 得到新向量 $\mathbf{v}' \in S_{b-1}^{dk}$, 此外对于 $\forall \mathbf{x} \in R_q^d$, 有 $\mathbf{G}(\mathbf{G}^{-1}(\mathbf{x})) = \mathbf{x}$ 。

另外, 本方案引入了近似原像采样算法 (ASPR)^[23] 用于生成部分私钥。首先, 根据文献[24]提出的方法, 将工具矩阵 $\mathbf{G}$ 按次数分割为低

阶工具矩阵 $\mathbf{G}_L = \mathbf{I}_d \otimes [1|b|\dots|b^{l-1}]$ 和高阶工具矩阵 $\mathbf{G}_H = \mathbf{I}_d \otimes [b^l|b^{l+1}|\dots|b^{k-1}]$ 。然后, 算法输入服从 $\mathcal{U}(S_1^{2d \times d(k-l)})$ 分布的矩阵 $\mathbf{R}$ 、矩阵 $\mathbf{A}' \in R_q^{d \times d}$ 、伴随式 $\mathbf{u} \in R_q^d$ 、一个正整数 M 和高斯参数 s , 先采样 $\mathbf{p}_1 \leftarrow D_S^{2d} \in R^{2d}$, 依次计算 $\mathbf{w} = \mathbf{u} - [\mathbf{I}_d | \mathbf{A}'] \mathbf{p}_1 \bmod q$ 和 $\mathbf{z} = \mathbf{G}^{-1}(\mathbf{w}) \in S_{b-1}^{dk}$ 。随后, 将 $\mathbf{z}$ 解析为满足 $\mathbf{G}\mathbf{z} = \mathbf{G}_L \mathbf{z}_L + \mathbf{G}_H \mathbf{z}_H$ 的 $\mathbf{z}_L \in S_{b-1}^{dl}$ 和 $\mathbf{z}_H \in S_{b-1}^{d(k-l)}$, 再计算 $\mathbf{v}'_1 = \mathbf{p}_1 + \mathbf{R}\mathbf{z}_H \in R_q^{2d}$ 。继续采样 $u \xleftarrow{\$} [0, 1]$, 若 $u > \min\left(1, \frac{D_S^{2d}}{MD_S^{2d}(\mathbf{p}_1)}\right)$, 则需要重新回到开头, 采样 $\mathbf{p}_1$ 重新计算, 否则继续。最后, 计算 $\mathbf{v}_1 = \mathbf{v}'_1 + \begin{bmatrix} \mathbf{G}_L \\ \mathbf{0} \end{bmatrix}$ 和 $\mathbf{v}_2 = \mathbf{z}_H$, 输出 $\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}$, 使得 $[\mathbf{A} | \mathbf{G}_H - \mathbf{A}\mathbf{R}] \cdot \mathbf{v} = \mathbf{u}$ 成立, 其中 $\mathbf{A} = [\mathbf{I}_d | \mathbf{A}']$ 。

基于以上算法, 有如下定理成立:

定理 1 给定一个矩阵 $\mathbf{R} \sim \mathcal{U}(S_1^{2d \times d(k-l)})$ 、一个正整数 M 和一个 $S_{b-1}^{d(k-l)}$ 上的分布 $\chi = \mathbf{G}_L \mathbf{G}_L^{-1}(\mathcal{U}(R_q^d))$, 先采样 $\mathbf{v}'_1 \leftarrow D_S^{2d}$ 、 $\mathbf{v}_2 \leftarrow \mathbf{G}_H^{-1}(\mathcal{U}(R_q^d))$ 和 $\mathbf{e} \leftarrow \chi$, 再计算 $\mathbf{v} = [\mathbf{v}'_1^T + [\mathbf{e}^T | \mathbf{0}] \mathbf{v}_2^T]^T$ 和 $\mathbf{v} = [\mathbf{v}'_1^T + [\mathbf{e}^T | \mathbf{0}] \mathbf{v}_2^T]^T$, 最后, 以 $\frac{1}{M}$ 的概率返回 $(\mathbf{u}, \mathbf{v})$, 所得 $(\mathbf{u}, \mathbf{v})$ 和 $(\mathbf{u}', \mathbf{v}')$ 是统计不可区分的, 其中 $\mathbf{u}' \xleftarrow{\$} R_q^d$ 、 $\mathbf{v}' = \text{ASPR}(\mathbf{R}, \mathbf{A}', \mathbf{u}', s)$ 。

2.4 困难问题假设

本方案的安全性基于 MLWE^[25]、判定性模误差学习 (d-MLWE)^[25]、模短整数解 (MSIS)^[26]、判定性模误差学习变体 (d-VMLWE)^[23] 这些困难问题。

2.4.1 MLWE 和 d-MLWE

给定一个安全参数 λ 、两个正整数 m 和 n 、模数 $q \geq 2$ 、一个 R_q 上的分布 χ , 对一个固定的秘密向量 $\mathbf{s} \xleftarrow{\$} R_q^n$, 定义 MLWE 分布为 $\text{MLWE}_s(m, n, q, \chi) = (\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}) \mid \mathbf{A} \leftarrow R_q^{m \times n}, \mathbf{e} \leftarrow \chi^m$ 。基于上述的 m, n, q, χ 参数, d-MLWE (m, n, q, χ) 问题旨在区分 MLWE 分布 $\text{MLWE}_s(m, n, q, \chi)$ 和均匀分布 $\mathcal{U}(R_q^{m \times n} \times R_q^m)$ 。

2.4.2 MSIS

给定两个正整数 m 和 n 、模数 $q \geq 2$ 、一个参数 $\beta \in (0, q)$ 、一个矩阵 $\mathbf{A} \leftarrow R_q^{m \times n}$, MSIS (m, n, q, β) 问题旨在找到一个向量 $\mathbf{s} \in R_q^m$, 使得 $\mathbf{A}^T \mathbf{s} = \mathbf{0} \bmod q$ 且 $0 < \|\mathbf{s}\|_\infty < \beta$ 。

2.4.3 d-VMLWE

给定 4 个正整数 d, m, k, q 和一个分布 χ , 对一个服从 $\chi^{(d+m) \times k}$ 的矩阵 $\mathbf{R}$, d-VMLWE_R (d, m, k, q, χ) 问题旨在区分 $(\mathbf{A}', [\mathbf{I}_m | \mathbf{A}'] \mathbf{R} \bmod q)$ 和 $(\mathbf{A}', \mathbf{B})$

两个分布,其中 $A' \xleftarrow{\$} R_q^{m \times d}$, $B \xleftarrow{\$} R_q^{m \times k}$, I_m 为单位矩阵。

2.5 ChaCha20-Poly1305 算法

本方案采用 ChaCha20-Poly1305 作为对称加密算法,该算法由文献[27]提出,并于 2015 年通过标准化,成为 TLS/SSL 协议的核心组件,也是 Google 所采用的一种新式认证加密算法。ChaCha20-Poly1305 不仅在 CPU 为精简指令集的 ARM 平台上具有出色的性能表现,还提供更优的抗侧信道攻击安全性^[28]。此外,由于该算法采用了 256 位的对称密钥,在对抗能快速搜索密钥的 Grover 量子算法时也具有较高的安全性^[29]。

3 方案构造

3.1 系统模型

本文设计了一种 IoT 时序数据安全共享模型,由数据所有者(DO)、数据用户(DU)、密钥生成中心(KGC)和 IoT 设备组成。各组件功能与交互流程

如图 1 所示(彩色效果见《计算机工程》官网 HTML 版,下同):

1)DO 利用分布式传感器网络实时采集 IoT 时序数据并完成数据存储,DO 需要保障自己采集的数据不被盗用,是受安全威胁的对象。

2)DU 基于自身业务需求,向 DO 提交数据请求,经 DO 验证授权后,获取时序数据用于挖掘分析和应用决策,DU 需要保障请求数据时不被窃听,是受安全威胁的对象。

3)KGC 作为系统的可信根,由政府或第三方安全机构承担,为系统生成公私钥对及全局公共参数。KGC 为每个 DO、DU 生成并定期更新安全通信所需的部分私钥和假名,另一方面,KGC 由于掌握了系统的私钥,可能成为“诚实但好奇”的内部攻击者。

4)IoT 设备由 DO 实际部署和拥有,负责时序数据的实时采集或生成,针对原始数据执行对称加密后传输给 DO,其为 DO 私有的设备,不构成安全威胁。

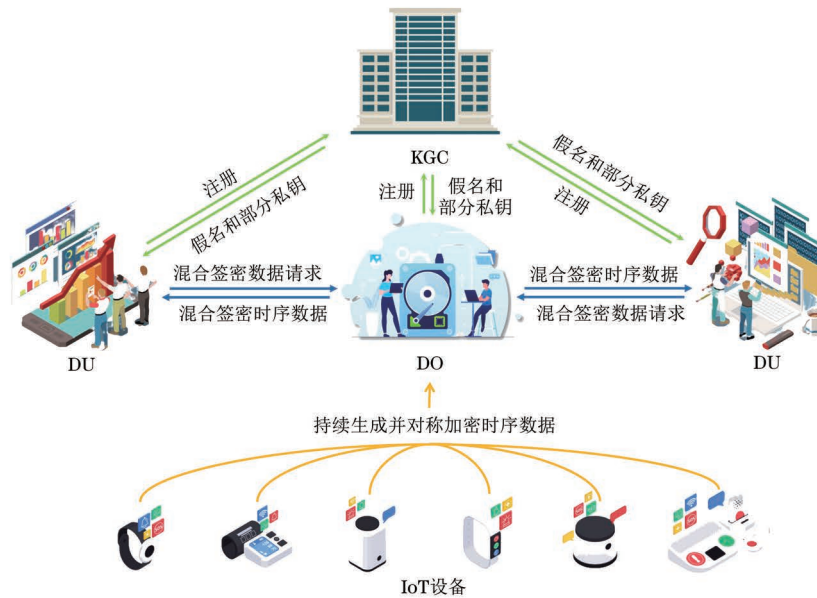


图 1 系统模型

Fig.1 System model

3.2 方案详细描述

本方案具体包括系统建立阶段、用户注册阶段、密文准备阶段、签密阶段、解签密阶段以及数据解密阶段。

3.2.1 系统建立阶段

给定一个安全参数 λ , KGC 首先选定模块格的模数 q 和度数 N ,再选定维度参数 d, k, n, m ,随后选定 $l, l' \in [0, k-1]$,再选定两个正整数 η, τ ,并计算 $\beta = \eta \cdot \tau$,设置两个工具矩阵 $G_H = I_d \otimes [2^l | \cdots | 2^{k-1}] \in R_q^{d \times d(k-l)}$ 和 $G_L = I_d \otimes [1 | \cdots | 2^{l-1}] \in R_q^{d \times dl}$,

再选定拒绝率 $M > 0$ 和高斯参数 $s > 0$,采样一个公钥矩阵 $A' \xleftarrow{\$} R_q^{d \times d}$ 。设定以下 6 个哈希函数:

- 1) $H_1: \{0, 1\}^* \rightarrow R_q^d$.
- 2) $H_2: \{0, 1\}^* \rightarrow B_\tau$.
- 3) $H_3: \{0, 1\}^* \rightarrow \{0, 1\}^{784}$.
- 4) $H_4: \{0, 1\}^* \rightarrow \{0, 1\}^{256}$.
- 5) $H_5: \{0, 1\}^* \rightarrow \{0, 1\}^{96}$.
- 6) $H_p: \{0, 1\}^* \rightarrow \{0, 1\}^{id}$.

式中: id 为用户假名的长度。选择两个中心二项分布参数 r, e 。选择时间密钥范数的边界值 β_{psk} 以及

3 个小边界值 γ_1, γ_2 和 γ_d 。确定一个假名和密钥更新的时间跨度 Δ , 选择一个随机值 $\text{salt} \in \{0, 1\}^{256}$, 公开当前时间区间的标签 $\text{Tag}_c = H_p(T_{\text{current}} \parallel \text{salt})$, 其中 T_{current} 表示当前时间区间, 每隔 Δ , KGC 都将重新计算并公布 Tag_c 。最后, 建立一个撤销列表 RL , 用于记录用户信息。

确定系统参数后, KGC 生成系统公私钥。首先, KGC 随机采样一个矩阵 $\mathbf{R} \xleftarrow{\$} S_1^{2d \times d(k-l)}$, 使得 $\|\mathbf{R}\| \leq \sqrt{2nd} + \sqrt{nd(k-l)}$, 令 $\mathbf{B} = [\mathbf{I}_d \mid \mathbf{A}']\mathbf{R} \bmod q \in R_q^{d \times d(k-l)}$, 再将 $\mathbf{B}$ 解析为 $\mathbf{B}_L + \mathbf{B}_H$, 其中 $\mathbf{B}_L \in S_{2^{l'}}^{d \times d(k-l)}$, $\mathbf{B}_H \in 2^{l'} S_{2^{k-l'-1}}^{d \times d(k-l)}$ 。之后, KGC 设置系统私钥为 $\text{msk} = \mathbf{R}$, 系统公钥为 $\text{mpk} = \mathbf{B}_H$ 。最后, KGC 公开系统参数 $PP = \{\mathbf{A}', \mathbf{G}_H, q, N, d, k, n, m, l, l', \tau, \eta, \beta, \beta_{psk}, M, r, e, s, \gamma_1, \gamma_2, \gamma_d, \text{mpk}, \text{Tag}_c\}$ 。

如表 2 所示, 本文给出一组可行的模块格参数, 对参数选择策略感兴趣的读者可参考文献[23]。

表 2 模块格相关参数选择

Table 2 Selection of parameters related to module lattice

模块格参数	参数值
q	131 071
N	1 024
d	1
k	16
l	15
l'	8
M	20
s	20 105
τ	60
η	2
γ_d	36 895
β_{psk}	16 638

3.2.2 用户注册阶段

在加入系统前, DO 和 DU 都必须向 KGC 完成注册。以用户 ID_m 为例, 首先 ID_m 随机采样一个部分私钥 $\mathbf{s}_m \xleftarrow{\$} \psi_\eta^n$, 再采样一个误差 $\mathbf{e}_m \xleftarrow{\$} \psi_\eta^m$, 采样一个公钥矩阵 $\mathbf{C}_m \xleftarrow{\$} R_q^{m \times n}$, 随后计算 $\mathbf{b}_m = \mathbf{C}_m \mathbf{s}_m + \mathbf{e}_m \in R_q^m$, 将部分公钥设置为 $\{\mathbf{C}_m, \mathbf{b}_m\}$, 随后 ID_m 将真实身份 RID_m 连同 $\{\mathbf{C}_m, \mathbf{b}_m\}$ 一起发送给 KGC 申请注册。

KGC 收到 ID_m 的申请后, 先根据 ID_m 的真实身份 RID_m 生成一个假名 $PID_m = H_p(RID_m \parallel \text{mpk}) \oplus \text{Tag}_c$, 然后将 $\{RID_m, H_p(RID_m \parallel \text{mpk}), \text{Exp}\}$ 信息记录在撤销列表 RL 上, 其中 Exp 为 ID_m 身份过期时间, 再公开 PID_m 与 $\{\mathbf{C}_m, \mathbf{b}_m,$

$H_p(RID_m \parallel \text{mpk})\}$ 的映射。接着, KGC 用文献[22]中的原像采样函数生成 $\mathbf{d}_m \leftarrow \text{ASPR}(\mathbf{R}, \mathbf{A}', H_1(PID_m \parallel \text{mpk} \parallel \text{Tag}_c), s)$, 其中 $\mathbf{d}_m \in R_q^{d(2+k-l)}$ 。将 $\mathbf{d}_m$ 解析为 $(\mathbf{d}_{m,1}^\top \mid \mathbf{d}_{m,2}^\top)^\top$, 其中 $\mathbf{d}_{m,1} \in R_q^{2d}$, $\mathbf{d}_{m,2} \in R_q^{d(k-l)}$, 再计算 $\mathbf{g}_m = ([\mathbf{I}_d \mid \mathbf{A}']\mathbf{R} - \mathbf{B}_H)\mathbf{d}_{m,2}$ 。然后将 $\mathbf{d}_{m,1}$ 解析为 $[\mathbf{d}_{m,1,1}^\top \mid \mathbf{d}_{m,1,2}^\top]^\top$, 其中 $\mathbf{d}_{m,1,1}, \mathbf{d}_{m,1,2} \in R_q^d$, 再令 $\mathbf{d}_{m,1,1} = \mathbf{d}_{m,1,1} - \mathbf{g}_m$ 。最后, 将部分私钥

psk_m 设置为 $\begin{bmatrix} \mathbf{d}_{m,1,1} \\ \mathbf{d}_{m,1,2} \\ \mathbf{d}_{m,2} \end{bmatrix}$: 若 $\|\text{psk}_m\| > \beta_{psk}$, 则需要重新

生成部分私钥; 否则, 就将 $\{\text{psk}_m, PID_m\}$ 通过安全信道发送给用户 ID_m 。

用户 ID_m 在收到 $\{\text{psk}_m, PID_m\}$ 后, 首先验证 $PID_m \oplus \text{Tag}_c = H_p(RID_m \parallel \text{mpk})$ 是否成立, 如果不成立, 则重新申请注册, 反之则接受此假名并继续。

接着, ID_m 将 psk_m 解析为 $\begin{bmatrix} \mathbf{d}_{m,1,1} \\ \mathbf{d}_{m,1,2} \\ \mathbf{d}_{m,2} \end{bmatrix}$ 并验证

$\mathbf{d}_{m,1,1} + \mathbf{A}' \mathbf{d}_{m,1,2} + (\mathbf{G}_H - \mathbf{B}_H) \mathbf{d}_{m,2} \bmod q = H_1(RID_m \parallel \text{mpk} \parallel \text{Tag}_c) \in R_q^d$ 是否成立, 如果成立则接受此部分私钥, 否则需要重新申请注册。最后, ID_m 将自身完整私钥 sk_m 设置为 $(\mathbf{s}_m, \text{psk}_m)$, 将自身完整公钥 pk_m 设置为 $(\mathbf{C}_m, \mathbf{b}_m, H_p(RID_m \parallel \text{mpk}))$ 。

3.2.3 密文准备阶段

本节仅介绍 DO 及其部署的 IoT 设备如何利用本文提出的基于双向哈希链的密钥生成机制准备时序数据密文。

由于 IoT 设备资源受限, 难以在本地安全保存持续采集的时序数据, 需将每段数据加密后传输给 DO。在 DU 向 DO 申请数据时, 此类密文将直接作为 DO 混合签密时的数据密文被发送。在 IoT 设备部署前, DO 先为其随机生成两个长期密钥 $\delta, \epsilon \in \{0, 1\}^{256}$, 随后将这两个长期密钥存入 IoT 设备中, 同时也在本地保存。

IoT 设备在采集 DT 当日的数据前, 会提前生成对称加密密钥。该算法的核心是一个基于双向哈希链的密钥生成机制, 流程如图 2 所示。具体来说, 给定日期 DT 、当天采集的时序数据的时间区间个数 num , 对每个时间区间, IoT 设备会通过双向哈希链生成其对应的对称加密密钥 $K_i^{DT}, i \in [1, num]$ 。首先, IoT 设备为第 1 个时间区间计算其右密钥 $RT_1^{DT} = H_4(\delta, DT, 1)$ 。接着, 对剩余的时间区间 $i \in [2, num]$, IoT 设备依次计算 $RK_i^{DT} =$

$H_4(RK_{i-1}^{DT}T)$, 最终得到右密钥集合 $RKSet^{DT} = \{RK_1^{DT}, RK_2^{DT}, \dots, RK_{num}^{DT}\}$ 。同理, IoT 设备为第 num 个时间区间计算其左密钥 LK_{num}^{DT} , 再为剩余的时间区间 $i \in [1, num-1]$, 按 i 从大到小依次计算 $LK_i^{DT} = H_4(LK_{i+1}^{DT})$, 最终得到左密钥集合 $LKSet^{DT} = \{LK_1^{DT}, LK_2^{DT}, \dots, LK_{num}^{DT}\}$ 。得到左、右密钥集合后, 设备通过计算 $K_i^{DT} = H_4(LK_i^{DT} \oplus RK_i^{DT})$, $i \in [1, num]$, 为每一个时间区间生成用于执行对称加密的密钥, 得到加密密钥集合 $KSet^{DT} = \{K_1^{DT}, K_2^{DT}, \dots, K_{num}^{DT}\}$, 然后, IoT 设备在本地保存 $KSet^{DT}$, 丢弃 $LKSet^{DT}$ 和 $RKSet^{DT}$ 。当 IoT 设备

完成 DT 当日的的数据收集后, 得到 num 个连续时间区间的数据, 其中每段数据记为 M_i^{DT} , $i \in [1, num]$ 。此时, IoT 设备将使用对称加密算法加密数据。对每一段数据, IoT 设备从 $KSet^{DT}$ 中取出对应的 K_i^{DT} , 计算临时随机数 $N_i = H_5(DT, i)$ 用于进行对称加密操作, 并计算 $\{C_i^{DT}, Tag_i^{DT}\} = \text{ChaCha20-Poly1305}(K_i^{DT}, N_i, M_i^{DT})$, 其中标签 Tag_i^{DT} 用于保证非验证用户无法通过发动比特篡改区分密文, C_i^{DT} 则为加密后的时序数据密文。为保障前向安全, IoT 设备在加密完成后删除 K_i^{DT} , 防止攻击者通过该密钥窃取数据隐私。

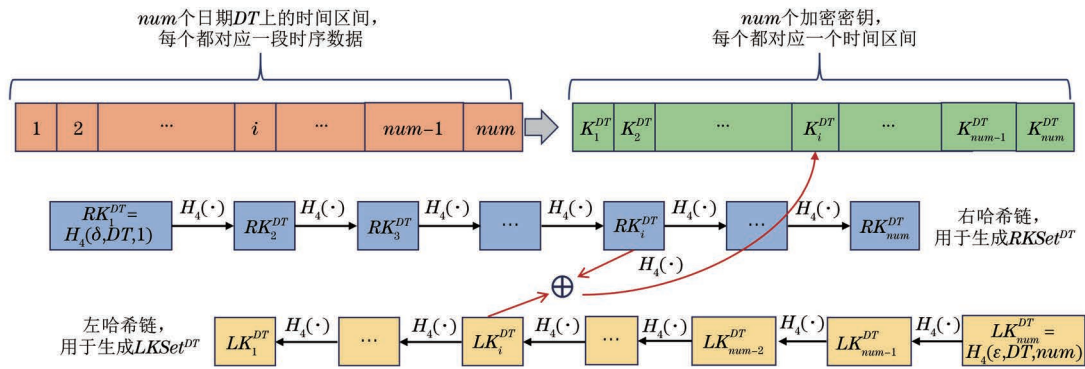


图 2 基于双向哈希链的密钥生成机制

Fig. 2 Key generation mechanism based on bidirectional hash chain

在得到全部密文后, IoT 设备将全部密文及其标签发送给 DO。由于 DO 同样持有长期密钥 $\{\delta, \epsilon\}$, 因此也可通过上文所述方法计算对称密钥以解密密文。

3.2.4 离线签密阶段

当数据发送方未准备好数据密文时, 如 DO 在调出所需时序数据密文或计算左右密钥时, 可并行执行离线签密。首先, 数据发送方 ID_S 随机选取一个整数 $u \xleftarrow{\$} R_1$, 计算 $(K, \text{coin}) = H_3(u \parallel PID_S \parallel PID_R \parallel pk_S \parallel pk_R)$, 其中 K 长度为 512 bit, coin 长度为 256 bit。再以 coin 为种子, 采样两个向量 $r_1 \leftarrow \phi_r \in R_q^d$ 和 $r_2 \leftarrow \phi_r \in R_q^m$, 其中 ϕ_r 是以 r 为参数的中心二项分布。继续用 coin 作为种子, 采样 $e_1 \leftarrow \phi_e^{d(2+k-l)}$, $e_2 \leftarrow \phi_e^n$, $e_{3,1} \leftarrow \phi_e$ 和 $e_{3,2} \leftarrow \phi_e$, 其中 ϕ_e 是以 e 为参数的中心二项分布。再令 $e_3 = e_{3,1} + e_{3,2} \in R_q$ 。接着, ID_S 计算密文的以下几个组件: $c_1 = [I_d | A' | (G_H - B_H)]^T r_1 + e_1 \in R_q^{d(2+k-l)}$, $c_2 = C_r^T r_2 + e_2 \in R_q^n$, $c_3 = r_1^T H_1(PID_R \parallel mpk \parallel Tag_c) + r_2^T b_r + \lfloor \frac{q-1}{2} u \rfloor + e_3 \in R_q$ 。继续采样 $y_1 \xleftarrow{\$} S_{\gamma_1}^{d(2+k-l)}$ 和 $y_2 \xleftarrow{\$} S_{\gamma_1}^n$, 并计算 $w_1 = [I_d | A' | (G_H - B_H)] y_1$ 和 $w_2 = \text{HighBits}(C_S y_2, \gamma_2)$, 最后输出离线签密结果 $CT_{\text{off}} = (c_1, c_2, c_3, K, y_1, y_2, w_1, w_2)$ 。

3.2.5 在线混合签密阶段

首先, 数据发送方 ID_S 准备需要签密的密钥明文。对于 DO, 其首先根据 DU 提出的日期 DT 上的特定时间区间 $[t_s, t_e]$, 通过利用长期密钥 $\{\delta, \epsilon\}$ 计算获取 $LK_{t_e}^{DT}$ 和 $RK_{t_s}^{DT}$, 令需要签密的密钥明文为 $PT_{\text{key}} = \{LK_{t_e}^{DT}, RK_{t_s}^{DT}\}$, 再令需要发送的时序数据密文为 $CT_{\text{data}} = \{C_{t_s}^{DT}, Tag_{t_s}^{DT}, C_{t_s+1}^{DT}, Tag_{t_s+1}^{DT}, \dots, C_{t_e}^{DT}, Tag_{t_e}^{DT}\}$, DU 只需发送一条经对称加密的消息来请求 $[t_s, t_e]$ 时间段内的时序数据, 因此密钥明文 PT_{key} 只包含一条密钥, 但仍需要填充至 512 bit 以符合算法设计。

得到 PT_{key} 之后, ID_S 先计算 $h \leftarrow H_2(PT_{\text{key}} \parallel w_1 \parallel w_2) \in B_\tau$, 再计算 $z_1 = y_1 + h d_S \in R_q^{d(2+k-l)}$ 和 $z_2 = y_2 + h s_S \in R_q^n$, 之后令 $y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$, $t = h \begin{bmatrix} d_S \\ s_S \end{bmatrix}$, $z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$ 。如果 $\|z_2\|_\infty \leq \gamma_1 - \beta$ 且 $\|\text{LowBits}(C_S y_2 - h s_S, 2\gamma_2)\|_\infty \leq \gamma_2 - \beta$, 就以 $\min(1, D_s^{d(2+k-l)}(y_1) / MD_{t,s}^{d(2+k-l)}(z_1))$ 的概率接受 $\{z, h\}$ 并继续, 否则需要重新计算 h 。然后, 计算 $CT_{\text{key}} = K \oplus PT_{\text{key}}$, 得到混合签密密文 $\sigma = \{CT_{\text{key}}, CT_{\text{data}}, c_1, c_2, c_3, z, h\}$ 。

$PID_S, pk_S, T\}$, 其中 T 为时间戳, 最后 ID_S 将 σ 发送给接收方 ID_R 。

3.2.6 解混合签密阶段

当 ID_R 收到 ID_S 发来的密文 σ 时, 首先检查时间戳 T , 如果超时则抛弃该密文; 再检查 $PID_S \oplus Tag_c = H_p(ID_S \parallel mpk)$ 是否成立, 若不成立, 说明 ID_S 身份已过期, 若两者皆通过则继续。 ID_R 计算 $u' = \left\lceil \frac{2}{q-1} \right\rceil (c_3 - c_1^T d_R - c_2^T s_R)$, 计算 $(K', coin) = H_3(u' \parallel PID_S \parallel PID_R \parallel pk_S \parallel pk_R)$, 再以 $coin$ 为种子, 采样两个向量 $r'_1 \leftarrow \psi_r \in R_q^d$ 和 $r'_2 \leftarrow \psi_r \in R_q^m$, 继续用 $coin$ 作为种子, 采样 $e'_1 \leftarrow \psi_e^{d(2+k-l)}$, $e'_2 \leftarrow \psi_e^n$, $e'_{3,1} \leftarrow \psi_e$ 和 $e'_{3,2} \leftarrow \psi_e$, 令 $e'_3 = e'_{3,1} + e'_{3,2} \in R_q$ 。接着计算 $c'_1 = [I_d | A'] (G_H - B_H)^T r'_1 + e'_1 \in R_q^{d(2+k-l)}$, $c'_2 = C_r^T r'_2 + e'_2 \in R_q^n$, $c'_3 = r'_1^T H_1(PID_R \parallel mpk \parallel Tag_c) + r'_2^T b_r + \left\lceil \frac{q-1}{2} \right\rceil u' + e'_3 \in R_q$ 。如果 $c_1 = c'_1, c_2 = c'_2, c_3 = c'_3$ 成立则继续, 否则终止解混合签密。接着, ID_R 计算密文明文 $PT'_{key} = K' \oplus CT_{key}$, 计算完毕后, ID_R 继续计算 $w'_1 = [I_d | A'] (G_H - B_H)^T z_1 - h H_1(PID_S \parallel mpk \parallel Tag_c)$ 和 $w'_2 = \text{HighBits}(C_S z_2 - h b_S)$ 。随后, ID_R 验证 $\|z_2\|_\infty \leq$

$\gamma_1 - \beta$ 和 $h = H_2(PT'_{key} \parallel w'_1 \parallel w'_2)$ 是否成立, 如果成立, 则接受 PT_{key} 并继续解对称加密。

当 DO 获得了 DU 发送的密钥 PT_{key} 后, 可以简单执行解对称加密操作, 故不作赘述, 本节仅讨论 DU 获得 $PT_{key} = \{LK_{t_e}^{DT}, RK_{t_s}^{DT}\}$ 后如何解密时序数据密文。 DU 得到端点的左右密钥 $LK_{t_e}^{DT}, RK_{t_s}^{DT}$ 和 $CT_{data} = \{C_{t_s}^{DT}, Tag_{t_s}^{DT}, C_{t_s+1}^{DT}, Tag_{t_s+1}^{DT}, \dots, C_{t_e}^{DT}, Tag_{t_e}^{DT}\}$ 后, 先利用双向哈希链机制生成解密密文的密钥。具体地, DU 按下标增序依次计算右密钥 $RK_i^{DT} = H_4(RK_{i-1}^{DT})$, 其中 $i \in [t_s+1, t_e]$, 按下标减序依次计算左密钥 $LK_i^{DT} = H_4(LK_{i+1}^{DT})$, 其中 $i \in [t_s, t_e-1]$ 。接着, 对每一个 $i \in [t_s, t_e]$, DU 先根据左右密钥计算解密密钥 $K_i^{DT} = H_4(LK_i^{DT} \oplus RK_i^{DT})$, 再计算 $N_i = H_5(DT, i)$, 使用 K_i^{DT} 计算并验证标签 Tag_i^{DT} , 验证失败则丢弃。最后, DU 解对称加密, 使用 $KSet_{t_s, t_e}^{DT} = \{K_{t_s}^{DT}, K_{t_s+1}^{DT}, \dots, K_{t_e}^{DT}\}$ 密钥集合和 $NonceSet_{t_s, t_e}^{DT} = \{N_{t_s}^{DT}, N_{t_s+1}^{DT}, \dots, N_{t_e}^{DT}\}$ 按下标一一对应解密密文集合 $CSet_{t_s, t_e}^{DT} = \{C_{t_s}^{DT}, C_{t_s+1}^{DT}, \dots, C_{t_e}^{DT}\}$, 得到所需的明文时序数据集 $PT_{data} = MSet_{t_s, t_e}^{DT} = \{M_{t_s}^{DT}, M_{t_s+1}^{DT}, \dots, M_{t_e}^{DT}\}$ 。 DO 与 DU 的交互流程如图 3 所示。

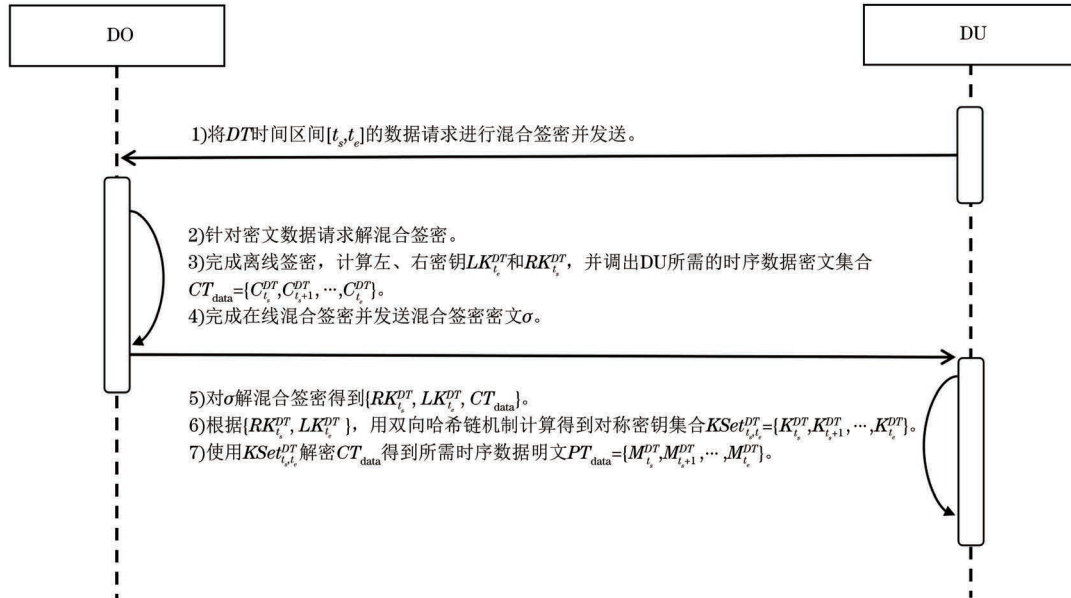


图 3 DO 和 DU 的交互流程

Fig.3 The interaction process between DO and DU

3.2.7 假名更新阶段

每隔一个固定时间段 Δ , 系统会进入一个新的时间区间 $T'_{current}$, KGC 将遍历撤销名单 RL , 根据新的时间区间 $T'_{current}$, 使用第 3.2.2 节中的方法为身份尚未过期的用户重新生成新的假名和私钥, 并通过安全信道发送给用户。

3.3 正确性验证

本节主要讨论解混合签密过程中数据接收方计算的 u', w'_1 和 w'_2 为何与发送方计算得出的 u, w_1 和 w_2 相等。

当 $\|r'_2 e_R - e_1^T d_R - e_2^T s_R + e_3\|_\infty < \frac{q}{4}$, 即原始

信号未被噪声淹没时,接收方 ID_R 计算得到的 u' 与发送方采样得到的 u 相等,证明如下:

$$\begin{aligned} & c_3 - c_1^T d_R - c_2^T s_R = \\ & r_1^T H_1 (PID_R \parallel mpk \parallel Tag_c) + r_2^T b_R + \\ & \left\lfloor \frac{q-1}{2} \right\rfloor u + e_3 - e_1^T d_R - \\ & r_1^T [I_d | A'] (G_H - B_H) \begin{bmatrix} d_{R,1,1} \\ d_{R,1,2} \\ d_{R,2} \end{bmatrix} - r_2^T C_R s_R - \\ & e_2^T s_R = r_2^T (C_R s_R + e_R - C_R s_R) + \left\lfloor \frac{q-1}{2} \right\rfloor u + \\ & e_3 - e_1^T d_R - e_2^T s_R = \\ & \left\lfloor \frac{q-1}{2} \right\rfloor u + (r_2^T e_R - e_1^T d_R - e_2^T s_R + e_3) \quad (1) \end{aligned}$$

由于 $u' = \left\lfloor \frac{2}{q-1} \right\rfloor (c_3 - c_1^T d_R - c_2^T s_R)$, 为便于呈现, 将 $(r_2^T e_R - e_1^T d_R - e_2^T s_R + e_3)$ 记为 ρ , 且 $\|\rho\|_\infty < \frac{q}{4}$ 。对于 u 中的元素 0, 有 $\left\lfloor \frac{2}{q-1} \right\rfloor \cdot \rho = 0$, 对于 u 中的元素 1, 则有 $\left\lfloor \frac{2}{q-1} \right\rfloor \cdot \left(\left\lfloor \frac{q-1}{2} \right\rfloor + u \right) = 1$ 。综上, 当 $\|r_2^T e_R - e_1^T d_R - e_2^T s_R + e_3\|_\infty < \frac{q}{4}$ 时, $u' = u$ 以极大概率成立。

在解混合签密过程中计算得出的 w'_1 与在线签密过程中计算得出的 w_1 相等, 证明如下:

$$\begin{aligned} & \text{令 } A = [I_d | A'], \text{ 有:} \\ & w'_1 = [I_d | A'] (G_H - B_H) z_1 - \\ & h H_1 (PID_S \parallel mpk \parallel Tag_c) = \\ & [A | (G_H - B_H)] (y_1 + h d_S) - \\ & h H_1 (PID_S \parallel mpk \parallel Tag_c) = \\ & [A | (G_H - B_H)] y_1 = w_1 \quad (2) \end{aligned}$$

由文献[29]中关于 HighBits 函数的证明可知 $w'_2 = w_2$, 证明如下:

$$\begin{aligned} & w'_2 = \text{HighBits}(C_S z_2 - h b_S) = \\ & \text{HighBits}(C_S (y_2 + h s_S) - h (C_S s_S + e_S)) = \\ & \text{HighBits}(C_S y_2 - h e_S) = w_2 \quad (3) \end{aligned}$$

4 安全性分析

本节将采用随机预言机模型证明本文所设计的抗量子 CLSC 方案满足 IND-CCA 机密性和 EUF-CMA 不可伪造性, 并展开非形式化安全性分析。为方便讨论, 本节不考虑前文提出的假名机制。

4.1 安全模型

根据文献[11, 13], 在典型的无证书签密工作中, 有两类敌手 $\mathcal{A}_I$ 和 $\mathcal{A}_{II}$, 其中 $\mathcal{A}_I$ 代表外部敌手, 其

可以调换用户的公钥, 但不知道系统主私钥, $\mathcal{A}_{II}$ 则代表内部敌手, 其知道系统主私钥但无法调换用户的公钥, 两者的攻击目标既包括破解密文获取隐私数据, 又包括伪造签名以欺骗接收者。给定以上两类敌手, 一款无证书签密方案需满足两类安全要求, 分别为 IND-CCA 机密性和 EUF-CMA 不可伪造性。为方便讨论, 在下文中, 令 $\mathcal{A}_I^1$ 和 $\mathcal{A}_{II}^1$ 为对方案机密性开展攻击的敌手, 令 $\mathcal{A}_I^2$ 和 $\mathcal{A}_{II}^2$ 为对方案不可伪造性开展攻击的敌手。

4.2 IND-CCA 机密性定理和证明

定理 2 如果 MLWE 和 VMLWE 问题是困难的, 则本方案对 $\mathcal{A}_I^1$ 敌手满足 IND-CCA 机密性。

证明 假设存在一名挑战者 $\mathcal{C}$ 试图解决 MLWE 和 VMLWE 问题, 并能够构造本方案环境, 在其中与 $\mathcal{A}_I^1$ 进行交互, 则本方案的 IND-CCA 安全性将通过以下 4 个交互式游戏进行证明, 其中, 记每个交互式游戏中 $\mathcal{A}_I^1$ 取胜的概率为 $Pr_{\text{IND-CCA}}^{G_i}[b=b']$ 。

1) 交互式游戏 0。

(1) 初始化阶段。挑战者 $\mathcal{C}$ 初始化本方案, 生成系统参数, 并将公共参数返回给 $\mathcal{A}_I^1$, 再创建 4 个列表 $\mathcal{L}_{\text{user}}$ 、 $\mathcal{L}_{H_1}$ 、 $\mathcal{L}_{H_2}$ 和 $\mathcal{L}_{H_3}$, 以记录随机预言机问询结果。

(2) 问询阶段 1。 $\mathcal{A}_I^1$ 可以执行如下问询:

① $\mathcal{O}_{H_1}$ 。输入 $\langle ID_i, mpk \rangle$, 如果 h_i 已存在于 $\mathcal{L}_{H_1}$ 中, 则 $\mathcal{C}$ 直接返回 h_i ; 否则, $\mathcal{C}$ 采样 $h_i \xleftarrow{\$} R_q^d$ 并将 $\langle ID_i, mpk, h_i \rangle$ 写入 $\mathcal{L}_{H_1}$ 。

② $\mathcal{O}_{H_2}$ 。输入 $\langle m, w_1, w_2 \rangle$, 如果 h 已存在于 $\mathcal{L}_{H_2}$ 中, 则 $\mathcal{C}$ 直接返回 h ; 否则, $\mathcal{C}$ 采样 $h \xleftarrow{\$} R_q$ 并将 $\langle m, w_1, w_2, h \rangle$ 写入 $\mathcal{L}_{H_2}$ 。

③ $\mathcal{O}_{H_3}$ 。输入 $\langle \bar{m}, ID_S, ID_R, pk_S, pk_R \rangle$, 如果 h 已存在于 $\mathcal{L}_{H_3}$ 中, 则 $\mathcal{C}$ 直接返回 h ; 否则, $\mathcal{C}$ 采样 $h \xleftarrow{\$} \{0, 1\}^{768}$ 并将 $\langle \bar{m}, ID_S, ID_R, pk_S, pk_R, h \rangle$ 写入 $\mathcal{L}_{H_3}$ 中。

④ $\mathcal{O}_{rpk}$ 。 $\mathcal{A}_I^1$ 给定一个 ID_i 和一个用于替换的公钥 pk'_i , $\mathcal{C}$ 将 $\mathcal{L}_{\text{user}}$ 中的 $\langle ID_i, psk_i, sk_i, pk_i \rangle$ 更新为 $\langle ID_i, psk'_i, sk'_i, pk'_i \rangle$, 其中 $\langle psk'_i, sk'_i \rangle$ 也需要 $\mathcal{A}_I^1$ 提供。

⑤ $\mathcal{O}_{psk}$ 。 $\mathcal{A}_I^1$ 输入一个 ID_i , $\mathcal{C}$ 先检查 ID_i 是否为之后在挑战阶段会用到的 ID_S^* 或者 ID_R^* , 如果都不是, 则 $\mathcal{C}$ 检查 ID_i 是否存在于 $\mathcal{L}_{\text{user}}$ 中, 若存在, 则直接返回与之对应的 psk_i , 否则, $\mathcal{C}$ 调用 $\mathcal{O}_{H_1}$ 并执行第 3.2.2 节中 KGC 为用户生成部分私钥的算法, 来为 ID_i 生成部分私钥 psk_i , 再将结果发给 $\mathcal{A}_I^1$, 最后将 $\langle ID_i, psk_i, *, * \rangle$ 写入 $\mathcal{L}_{\text{user}}$ 中。

⑥ $\mathcal{O}_{usk}$ 。 $\mathcal{A}_1^1$ 输入一个 ID_i , $\mathcal{C}$ 先检查 ID_i 是否为 ID_S^* 或者 ID_R^* , 如果都不是, 则 $\mathcal{C}$ 检查 ID_i 是否存在于 $\mathcal{L}_{user}$ 中, 若存在, 则直接返回与之对应的 sk_i , 否则, $\mathcal{C}$ 执行第 3.2.2 节中用户为自己生成部分密钥的算法来为 ID_i 生成完整公私钥 $\langle pk_i, sk_i \rangle$, 再将 sk_i 发给 $\mathcal{A}_1^1$, 最后将 $\langle ID_i, psk_i, sk_i, pk_i \rangle$ 写入 $\mathcal{L}_{user}$ 中。

⑦ $\mathcal{O}_{upk}$ 。 $\mathcal{A}_1^1$ 输入一个 ID_i , $\mathcal{C}$ 检查 ID_i 是否存在于 $\mathcal{L}_{user}$ 中, 若存在, 则直接返回与之对应的 pk_i , 否则, $\mathcal{C}$ 执行第 3.2.2 节中用户为自己生成部分密钥的算法来为 ID_i 生成完整公私钥 $\langle pk_i, sk_i \rangle$, 再将 pk_i 发给 $\mathcal{A}_1^1$, 最后将 $\langle ID_i, psk_i, sk_i, pk_i \rangle$ 写入 $\mathcal{L}_{user}$ 中。

⑧ $\mathcal{O}_{sc}$ 。 $\mathcal{A}_1^1$ 输入一条消息 m 、发送方和接收方的身份 $\langle ID_S, ID_R \rangle$, $\mathcal{C}$ 调用 $\mathcal{O}_{H_2}, \mathcal{O}_{H_3}$ 来执行完整签密流程, 得到密文 σ , 并将其返回给 $\mathcal{A}_1^1$ 。

⑨ $\mathcal{O}_{usc}$ 。 $\mathcal{A}_1^1$ 输入一条密文 σ 、发送方和接收方的身份 $\langle ID_S, ID_R \rangle$, $\mathcal{C}$ 调用 $\mathcal{O}_{H_1}, \mathcal{O}_{H_2}$ 和 $\mathcal{O}_{H_3}$ 来执行完整解密流程, 得到明文 m , 并将其返回给 $\mathcal{A}_1^1$ 。

(3)挑战阶段。 $\mathcal{A}_1^1$ 会选择两条消息 (m_0, m_1) 、发送方和接收方的身份 (ID_S^*, ID_R^*) , 将它们全部发给 $\mathcal{C}$ 。 $\mathcal{C}$ 随机选择 $b \in \{0, 1\}$, 然后对消息 m_b 执行完整的签密过程, 生成密文 σ^* , 将挑战密文发给 $\mathcal{A}_1^1$ 。

(4)问询阶段 2。除了不能对挑战密文 σ^* 进行解签密问询 $\mathcal{O}_{usc}(\sigma^*, ID_S^*, ID_R^*)$ 外, $\mathcal{A}_1^1$ 可以进行上文所述的各种问询。

(5)猜测阶段。 $\mathcal{A}_1^1$ 输出其猜测 $b' \in \{0, 1\}$, 如果 $b' = b$, 且该敌手在问询阶段 1 中没有对 ID_S^*, ID_R^* 进行过 $\mathcal{O}_{rpk}, \mathcal{O}_{psk}$ 和 $\mathcal{O}_{usk}$ 问询, 也没有在问询阶段 2 中对挑战密文 σ^* 进行解签密问询 $\mathcal{O}_{usc}(\sigma^*, ID_S^*, ID_R^*)$, 那么称 $\mathcal{A}_1^1$ 获得了游戏胜利。

定义 $\mathcal{A}_1^1$ 破解本方案 IND-CCA 安全性的优势为 $Adv_{IND-CCA} = \left| Pr_{IND-CCA}[b=b'] - \frac{1}{2} \right|$, 而在该游戏中, 敌手等同于直接对本方案的原始设计进行攻击, 故 $Pr_{IND-CCA}^{G_0}[b=b'] = Pr_{IND-CCA}[b=b']$ 。

2) 交互式游戏 1。

在该游戏中, $\mathcal{C}$ 不会如实执行 $\mathcal{O}_{H_1}$ 和 $\mathcal{O}_{psk}$, 会在不使用系统主私钥的条件下按以下方式执行, 其他部分则与交互式游戏 0 一致。

(1) $\mathcal{O}_{H_1}$ 。给定 $S_{b'-1}^d$ 上的误差分布 $\chi = \mathbf{G}_L \mathbf{G}_L^{-1}(U(R_q^d))$, 当 $\mathcal{A}_1^1$ 输入 $\langle ID_i, mpk \rangle$ 时, 如果 h_i 已存在于 $\mathcal{L}_{H_1}$ 中, 则 $\mathcal{C}$ 直接返回 h_i , 否则, $\mathcal{C}$ 先采样 $\mathbf{v}'_1 \leftarrow D_s^{2d}, \mathbf{v}_2 \leftarrow \mathbf{G}_H^{-1}(U(R_q^d))$ 以及 $\mathbf{e} \leftarrow \chi$, 然后 $\mathcal{C}$ 计

算 $\mathbf{v} = [\mathbf{v}'_1^T + [\mathbf{e} | \mathbf{0}] | \mathbf{v}_2^T]^T \in R_q^{d(2+k-l)}$, 以 $\frac{1}{M}$ 的概率

返回 $\mathbf{v}$ 。接着, $\mathcal{C}$ 计算 $\mathbf{h}_i = [\mathbf{A} | \mathbf{G}_H - \mathbf{A}\mathbf{R}] \mathbf{v} = \mathbf{u}$, 其中 $\mathbf{A} = [\mathbf{I}_d | \mathbf{A}']$, 然后将 $\langle ID_i, mpk, \mathbf{h}_i \rangle$ 写入 $\mathcal{L}_{H_1}$ 中。最后, $\mathcal{C}$ 按第 3.2.2 节中 KGC 为用户生成部分私钥的方法继续计算得到 psk_i , 并将 $\langle ID_i, psk_i, *, * \rangle$ 写入 $\mathcal{L}_{user}$ 中。

(2) $\mathcal{O}_{psk}$ 。 $\mathcal{A}_1^1$ 输入一个 ID_i , $\mathcal{C}$ 检查 ID_i 是否存在于 $\mathcal{L}_{user}$ 中, 若存在, 则直接返回与之对应的 psk_i , 否则, $\mathcal{C}$ 调用 $\mathcal{O}_{H_1}$ 来为 ID_i 生成 psk_i , 并将 $\langle ID_i, psk_i, *, * \rangle$ 写入 $\mathcal{L}_{user}$ 中, 把 psk_i 发送给 $\mathcal{A}_1^1$ 。

交互式游戏 1 和交互式游戏 0 的唯一区别在于, 前者生成 psk_i 时不会用到 msk , 而根据定理 1, 敌手在这两个游戏中取胜的概率在统计层面上是不可区分的, 即 $|Pr_{IND-CCA}^{G_0}[b=b'] - Pr_{IND-CCA}^{G_1}[b=b']| < negl(\lambda)$, 其中 $negl(\lambda)$ 代表一个可忽略的值。

3) 交互式游戏 2。

在该游戏中, $\mathcal{C}$ 不再如实执行 $\mathcal{O}_{usc}$, 而按如下方法不使用 sk_R^* 执行。

$\mathcal{O}_{usc}$: 当 $\mathcal{A}_1^1$ 输入一条密文 $\sigma = (ct, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, z, h)$ 以及发送方、接收方的身份 (ID_S, ID_R) 时, $\mathcal{C}$ 先检查该密文与挑战密文 σ^* 是否一致, 若一致, 则 $\mathcal{C}$ 拒绝解密该密文, 反之, $\mathcal{C}$ 会从 $\mathcal{L}_{H_3}$ 中查找一组记录 $\langle \bar{m}', ID_S, ID_R, pk_S, pk_R, K' \rangle$, 使得以这组数据为输入执行离线签密算法得到的 $(\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$ 与 σ 中的 $(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$ 完全一致, 如果能查找到这样一组数据, 则 $\mathcal{C}$ 计算 $m' = K' \oplus ct$ 并将结果返回给 $\mathcal{A}_1^1$, 反之, 说明该密文不是通过合法问询生成的, $\mathcal{C}$ 直接拒绝此次问询。

与交互式游戏 1 不同, 在交互式游戏 2 中, $\mathcal{C}$ 解密与 ID_R^* 有关的密文时, 不再使用 ID_R^* 的私钥 sk_R^* , 但该区不影响解密所得明文的正确性, 也不影响返回明文的统计分布, 使得敌手无法区分是在交互式游戏 1 还是在交互式游戏 2 中进行的操作, 故敌手 $\mathcal{A}_1^1$ 在交互式游戏 2 中的优势与交互式游戏 1 相同, 即 $Pr_{IND-CCA}^{G_2}[b=b'] = Pr_{IND-CCA}^{G_1}[b=b']$ 。

4) 交互式游戏 3。

在该游戏中, $\mathcal{C}$ 不会如实执行 $\mathcal{O}_{sc}$, 而按如下方式不使用 pk_R^* 执行。

$\mathcal{O}_{sc}$: $\mathcal{A}_1^1$ 输入一条消息 m 、发送方和接收方的身份 (ID_S, ID_R) , $\mathcal{C}$ 先检查 ID_R 是否等于 ID_R^* , 如果不相等, $\mathcal{C}$ 按交互式游戏 2 中的方法继续执行, 如果相等, 则 $\mathcal{C}$ 先部分执行本方案中的离线签密算法得到 coin, 再以 coin 为种子采样 $\mathbf{c}'_1 \leftarrow R_q^{d(2+k-l)}, \mathbf{c}'_2 \leftarrow R_q^n$ 和

$c'_3 \leftarrow R_q$, 再令 $c'_3 = c'_3 + \left\lfloor \frac{q-1}{2} \right\rfloor \bar{m}$, 得到签密的密文。

首先, C 采样 $b'_R \xleftarrow{\$} R_q^m$ 代替用户公钥, 由 $d\text{-MLWE}_{s_R}(m, n, q, \psi_e)$, b'_R 和 b_R 是计算不可区分的, 所以游戏 2 中的 $[c_1^T | c_2^T | c_3^T]^T$ 和式(4)也是计算不可区分的。

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} [I_d | A' | (G_H - B_H)]^T \\ 0 \\ H_1(ID_R \| mpk)^T \end{bmatrix} r_1 + \begin{bmatrix} 0 \\ C_R^T \\ b_R'^T \end{bmatrix} r_2 + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \left\lfloor \frac{q-1}{2} \right\rfloor \bar{m} \quad (4)$$

接着, C 采样 $B'_H \xleftarrow{\$} 2^{l'} S_{s-k-l-1}^{d \times d(k-l)}$, 使得 B_H 和 B'_H 由于 $d\text{-VMLWE}_R(d, d, d(k-l), q, \psi_e)$ 难题而在计算上不可区分, 再采样 $u = [u_1^T | 0^T | u_2^T]^T$, 其中 $u_1 \xleftarrow{\$} R_q^{d(2+k-l)}$, $u_2 \xleftarrow{\$} R_q$ 。由于 $d\text{-MLWE}_{r_1}(d(2+k-l), d, q, \psi_e)$ 难题, 式(5)与式(4)也是计算不可区分的。

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = u + \begin{bmatrix} 0 \\ C_R^T \\ b_R'^T \end{bmatrix} r_2 + \begin{bmatrix} 0 \\ e_2 \\ e_3^{(2)} \end{bmatrix} + \left\lfloor \frac{q-1}{2} \right\rfloor \bar{m} \quad (5)$$

最后, C 采样 $v = [0 | v_1^T | v_2^T]^T$, 其中 $v_1 \xleftarrow{\$} R_q^n$, $v_2 \xleftarrow{\$} R_q$ 。由于 $d\text{-MLWE}_{r_2}(n+1, m, q, \psi_e)$ 难题, 式(6)与式(5)也是计算不可区分的。

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = u + v + \left\lfloor \frac{q-1}{2} \right\rfloor \bar{m} \quad (6)$$

由以上信息可知, 游戏 3 中 $\mathcal{O}_{sc}$ 的输出和游戏 2 中的输出是计算不可区分的, 所以有:

$$\begin{aligned} & |Pr_{\text{IND-CCA}}^{G_3}[b=b'] - Pr_{\text{IND-CCA}}^{G_2}[b=b']| < \\ & Adv_{d\text{-MLWE}_{s_R}(m, n, q, \psi_e)} + Adv_{d\text{-VMLWE}_R(d, d, d(k-l), q, \psi_e)} + \\ & Adv_{d\text{-MLWE}_{r_1}(d(2+k-l), d, q, \psi_e)} + Adv_{d\text{-MLWE}_{r_2}(n+1, m, q, \psi_e)} \end{aligned}$$

在游戏 3 中, 关于 ID_R^* 的密文都是均匀随机生成的, 所以敌手 $\mathcal{A}_1^1$ 在该游戏中获胜的概率为 $\frac{1}{2}$, 且由于 $d\text{-MLWE}$ 和 $d\text{-VMLWE}$ 难题, 可得:

$$\begin{aligned} Adv_{\text{IND-CCA}} &= \left| Pr_{\text{IND-CCA}}[b=b'] - \frac{1}{2} \right| = \\ & |Pr_{\text{IND-CCA}}^{G_0}[b=b'] - Pr_{\text{IND-CCA}}^{G_3}[b=b']| < \\ & negl(\lambda) + Adv_{d\text{-MLWE}_{s_R}(m, n, q, \psi_e)} + \\ & Adv_{d\text{-VMLWE}_R(d, d, d(k-l), q, \psi_e)} + \\ & Adv_{d\text{-MLWE}_{r_1}(d(2+k-l), d, q, \psi_e)} + \\ & Adv_{d\text{-MLWE}_{r_2}(n+1, m, q, \psi_e)} < 5 \cdot negl(\lambda) \end{aligned}$$

综上可知, 敌手 $\mathcal{A}_1^1$ 破解本方案 IND-CCA 机密性的优势是可以忽略的, 因此本方案面对敌手 $\mathcal{A}_1^1$ 具有 IND-CCA 安全性。

定理 3 如果 MLWE 和 VMLWE 问题是困难的, 则本方案对 $\mathcal{A}_H^1$ 敌手满足 IND-CCA 机密性。

证明 与定理 2 类似, 该定理也是基于 MLWE 难题和 VMLWE 难题, 通过交互式游戏证明, 其中, 在每个游戏的第一步, 挑战者 C 除了生成系统参数外, 还需要将系统主私钥发给 $\mathcal{A}_H^1$ 以对应敌手的能力, 此外, C 不再执行 $\mathcal{O}_{pk}$ 问询。该定理的证明流程同样可以分为 4 个交互式游戏, 其中, 交互式游戏 0 同样模拟真实攻击场景, 交互式游戏 1 中 C 不会如实生成部分私钥 psk_R^* , 交互式游戏 2 中 C 不会如实解签密与 ID_R^* 相关的密文, 交互式游戏 3 中 C 不会如实签密与 ID_R^* 相关的消息, 最后同样有签密所得密文的分布和均匀分布不可区分, 使得 $\mathcal{A}_H^1$ 的优势可以忽略, 最终证明本方案在面对 $\mathcal{A}_H^1$ 时具有 IND-CCA 安全性。

4.3 EUF-CMA 不可伪造性定理和证明

定理 4 如果存在一名概率多项式时间敌手 $\mathcal{A}_1^2$ 能够破解本方案的 EUF-CMA 安全性, 则存在一名概率多项式时间挑战者 C 能够以不可忽略的优势解决 MSIS 难题和 MLWE 难题实例。

证明 假设存在一名挑战者 C 试图解决 MSIS 和 MLWE 难题, 并能够构造本方案环境, 在其中与 $\mathcal{A}_1^2$ 进行交互, 则本方案的 EUF-CMA 安全性将通过以下 4 个交互式游戏进行证明, 其中, 记每个交互式游戏中 $\mathcal{A}_1^2$ 取胜的概率为 $Pr_{\text{EUF-CMA}}^{G_i}[\mathcal{A}_1^2 \text{ wins}]$ 。

1) 交互式游戏 0。

(1) 初始化阶段。与定理 2 中游戏 0 的初始化阶段相同。

(2) 问询阶段。与定理 2 中游戏 0 的问询阶段 1 相同。

(3) 伪造阶段。给定挑战密文 m^* 和一组挑战身份 (ID_S^*, ID_R^*) , $\mathcal{A}_1^2$ 输出一条密文 σ^* , 若该密文能够通过 C 的验证, 则认为 $\mathcal{A}_1^2$ 获得了游戏胜利。

定义敌手 $\mathcal{A}_1^2$ 破解本方案 EUF-CMA 安全性的优势为 $Adv_{\text{EUF-CMA}} = Pr_{\text{EUF-CMA}}[\mathcal{A} \text{ wins}]$, 由于该游戏模拟了真实攻击场景, 故有 $Pr_{\text{EUF-CMA}}^{G_0}[\mathcal{A} \text{ wins}] = Pr_{\text{EUF-CMA}}[\mathcal{A} \text{ wins}]$ 。

2) 交互式游戏 1。

在该游戏中, 和定理 2 中的交互式游戏 1 一样, C 在执行 $\mathcal{O}_{H_1}$ 和 $\mathcal{O}_{psk}$ 时不会使用 msk , 同样由定理 1, 该游戏和游戏 0 中 $\mathcal{A}_1^2$ 取胜的概率在统计上是不

可区分的, 即 $|Pr_{\text{EUF-CMA}}^{G_1}[\mathcal{A} \text{ wins}] - Pr_{\text{EUF-CMA}}^{G_0}[\mathcal{A} \text{ wins}]| < \text{negl}(\lambda)$ 。

3) 交互式游戏 2。

在该游戏中, $\mathcal{C}$ 不再如实地执行 $\mathcal{O}_{\text{sc}}$, 而按如下方法, 在不使用 sk_S^* 的情况下执行。

$\mathcal{O}_{\text{sc}}: \mathcal{A}_1^2$ 输入一条消息 m 、发送方和接收方的身份 (ID_S, ID_R) , $\mathcal{C}$ 先检查 ID_S 是否为 ID_S^* , 如果不是, 就按交互式游戏 1 中的方式执行, 反之, $\mathcal{C}$ 以 $\frac{1}{M}$ 的概率采样 $\mathbf{z} = [\mathbf{z}_1^T | \mathbf{z}_2^T]^T \leftarrow D_s^{2+k-l+n}$, 如果 $\|\mathbf{z}_2\|_{\infty} \leq \gamma_1 - \beta$, 则 $\mathcal{C}$ 采样 $h \in B_{\epsilon}$ 并计算 $\mathbf{w}_1 = [\mathbf{I}_d | \mathbf{A}'](\mathbf{G}_H - \mathbf{B}_H) \mathbf{z}_1 - h\mathbf{H}_1(ID_S \parallel \text{mpk}) \in R_q^d$ 和 $\mathbf{w}_2 = \text{HighBits}(\mathbf{C}_S \mathbf{z}_2 - h\mathbf{b}_S) \in R_q^m$, 最后, $\mathcal{C}$ 将 $\langle m, \mathbf{w}_1, \mathbf{w}_2, h \rangle$ 写入 $\mathcal{L}_{H_2}$ 中。

该游戏和游戏 1 的唯一区别在于, 在该游戏中, 给定一条消息 m , $\mathcal{C}$ 在为其生成签名时不会使用 ID_S^* 的私钥 sk_S^* , 但该操作并不改变所得密文的正确性和分布, 故 $\mathcal{A}_1^2$ 在游戏 2 中获胜的概率和在游戏 1 中相等, 即 $Pr_{\text{EUF-CMA}}^{G_2}[\mathcal{A} \text{ wins}] = Pr_{\text{EUF-CMA}}^{G_1}[\mathcal{A} \text{ wins}]$ 。

4) 交互式游戏 3。

在该游戏中, $\mathcal{C}$ 不再如实地执行 $\mathcal{O}_{\text{upk}}$, 而按如下方法, 在不使用 sk_S^* 的情况下执行。

$\mathcal{O}_{\text{upk}}: \mathcal{A}_1^2$ 输入一个身份 ID_i , $\mathcal{C}$ 首先检查 ID_i 是否等于 ID_S^* , 若不等, 则按游戏 2 中的方法执行, 反之, $\mathcal{C}$ 采样 $\mathbf{b}_S^* \leftarrow R_q^m$ 。

该游戏和游戏 2 的唯一区别在于, 在该游戏中, ID_S^* 的公钥生成方式不同, 而由 d-MLWE $_{r_1}(m, n, q, \psi_e)$ 难题可知 $|Pr_{\text{EUF-CMA}}^{G_3}[\mathcal{A} \text{ wins}] - Pr_{\text{EUF-CMA}}^{G_2}[\mathcal{A} \text{ wins}]| < \text{Adv}_{\text{d-MLWE}_{r_1}(n, m, q, \psi_e)}$ 。

由分叉引理^[30]可知, $\mathcal{A}_1^2$ 可以通过重放来对同一组 $(m, \mathbf{w}_1, \mathbf{w}_2)$ 生成 2 个合法签名 $\sigma^* = (ct^*, \mathbf{c}_1^*, \mathbf{c}_2^*, \mathbf{c}_3^*, \mathbf{z}^*, h^*)$ 和 $\sigma'^* = (ct'^*, \mathbf{c}_1'^*, \mathbf{c}_2'^*, \mathbf{c}_3'^*, \mathbf{z}'^*, h'^*)$, 故 $\mathcal{C}$ 对于一个 MSIS($d(3+k-l), d, q, \gamma_d$) 实例 $[-H_1(ID_S \parallel \text{mpk}) | [\mathbf{I}_d | \mathbf{A}'](\mathbf{G}_H - \mathbf{B}_H)] \mathbf{s} = 0$, 令 $\mathbf{s} = \begin{bmatrix} h'^* - h^* \\ \mathbf{z}_1'^* - \mathbf{z}_1^* \end{bmatrix}$, 由于 $h'^* \neq h^*$ 且 $h^*, h'^*, \mathbf{z}_1^*$ 和 $\mathbf{z}_1'^*$ 都很小, 所以可使此实例成立。同理, 由文献[24]可知, $\mathcal{C}$ 还可以解决 MSIS($m+n, n, q, \gamma_1$)。综上, 有 $Pr_{\text{EUF-CMA}}^{G_3} < \text{Adv}_{\text{MSIS}(d(3+k-l), d, q, \gamma_d)} + \text{Adv}_{\text{MSIS}(m+n, n, q, \gamma_1)}$ 。

综上所述, 可知:

$$\text{Adv}_{\text{EUF-CMA}} = Pr_{\text{EUF-CMA}}^{G_0}[\mathcal{A} \text{ wins}] < \text{negl}(\lambda) +$$

$$\text{Adv}_{\text{d-MLWE}_{r_1}(n, m, q, \psi_e)} + \text{Adv}_{\text{MSIS}(d(3+k-l), d, q, \gamma_d)} + \text{Adv}_{\text{MSIS}(m+n, n, q, \gamma_1)} < 4 \cdot \text{negl}(\lambda)。$$

而由 d-MLWE 和 MSIS 难题可知, 本方案在面对 $\mathcal{A}_1^2$ 敌手时具有 EUF-CMA 安全性。

定理 5 如果存在一名概率多项式时间敌手 $\mathcal{A}_1^2$ 能够破解本方案的 EUF-CMA 安全性, 那么存在一名概率多项式时间挑战者 $\mathcal{C}$ 能够以不可忽略的优势解决 MSIS 难题和 MLWE 难题实例。

证明 该定理的证明可参考定理 4 的证明, 将本方案对 $\mathcal{A}_1^2$ 的 EUF-CMA 安全性规约到 MLWE、VMLWE 和 MSIS 难题上, 最终证明本方案在面对 $\mathcal{A}_1^2$ 敌手时具有 EUF-CMA 安全性。

4.4 非形式化安全性分析

4.4.1 条件隐私安全性

在本方案中, 用户的假名 PID 由 KGC 通过计算 $\text{Tag}_c \oplus H_p(RID \parallel \text{mpk})$ 得到, 且其参与用户部分私钥的生成和更新过程, 与用户私钥绑定。因此, 即使一名用户非法获取新的时间标签 Tag'_c 并计算 $PID' = \text{Tag}'_c \oplus H_p(RID \parallel \text{mpk})$ 得到新的合法假名, 也无法得到与该假名对应的合法的部分私钥, 导致无法正常通信。另一方面, 由于哈希函数的单向性, 虽然接收方可以根据假名计算出 $H_p(RID \parallel \text{mpk})$, 但其无法从中获取发送方的真实身份 RID , 所以当接收方接收到问题数据(如虚假错误信息或无意义信息)时, 只能通过提交此值向 KGC 提出申诉, 由 KGC 查阅撤销名单获取发送方的真实身份。综上, 由于接收方无法根据假名直接获取发送方的真实身份, 且接收方可在接收到有问题的数据时向 KGC 发起申诉并由后者查出发送方的真实身份, 所以本方案实现了安全的条件隐私机制。

4.4.2 双向哈希链安全性

本方案通过双向哈希链为时序数据生成密钥, 其中, 由于任一条哈希链都是单向的, 攻击者无法通过当前左密钥 LK_i^{DT} 得到 LK_{i+1}^{DT} , 所以尽管攻击者能计算出 RK_{i+1}^{DT} , 也无法得到 K_{i+1}^{DT} , 从而保障了密钥的安全性。另一方面, 本方案在得到 LK_i^{DT} 和 RK_i^{DT} 后, 不是简单地进行异或操作得到 K_i^{DT} , 而需再次进行哈希运算, 因此可抵御彩虹表攻击, 进一步提高了密钥的安全性。

5 性能分析

如表 3 所示, 将本方案与 4 种无证书签密方案进行对比实验, 其中 IBSC^[6] 和 LHSC^[11] 为传统的非抗量子无证书签密方案, IBSC^[6] 提出了一种基于

双线性映射的在线/离线 CLSC 方案, LHSC^[11]提出了一种基于 ECC 的轻量高效的无证书混合签密方案。PQSC^[13]和 MLSC^[18]则为后量子 CLSC 方案, PQSC^[13]提出了一种基于传统格密码学的方案, MLSC^[18]提出了一种基于模块格的 CLSC 方案。另外, 本节针对方案可扩展性额外进行了理论分析。

表 3 对比方案信息

Table 3 Information of comparative schemes

方案	期刊	年份
IBSC ^[6]	IEEE IOTJ	2024
LHSC ^[11]	IEEE IOTJ	2024
PQSC ^[13]	IEEE TIFS	2025
MLSC ^[18]	IEEE IOTJ	2025

5.1 实验环境

为了模拟 DO 和 DU, 本文在一台配备 Intel i5-13450HX CPU、32 GB 内存的笔记本上使用 Ubuntu 22.04 Desktop 开展实验。为了模拟 IoT 设备, 本文在一台配备 Cortex-A72 1.5 GHz CPU、4 GB 内存的树莓派 4B 上展开实验。

5.2 IoT 设备侧计算开销

本节分析 ChaCha20-Poly1305 算法与 AES256-GCM 算法用于加解密操作的效率对比, 然后分析在时序数据不断增加的情况下, 本方案中 IoT 设备上的对称密钥生成、数据加密和数据解密 3 种操作的计算开销变化情况。实验使用 Python 的 Pycryptodome 库实现, 结果如表 4 所示, 其中各密码学操作耗时均为 1 000 次重复实验的平均结果。

表 4 各密码学操作在 IoT 设备上的执行时间

Table 4 The execution time of various cryptographic operations on IoT devices

操作	时间
TD_{hash}	0.002 61
$TD_{ChaCha20}^{enc}$	0.238 00
$TD_{ChaCha20}^{dec}$	0.661 00
TD_{AES256}^{enc}	0.460 00
TD_{AES256}^{dec}	2.069 00

表 4 中展示了针对 1 KB 数据进行对应操作所花时间。其中, TD_{hash} 表示进行一次哈希操作所花时间, $TD_{ChaCha20}^{enc}$ 和 $TD_{ChaCha20}^{dec}$ 分别表示用 ChaCha20-Poly1305 算法进行一次加密和解密所花时间, TD_{AES256}^{enc} 和 TD_{AES256}^{dec} 分别表示用 AES256-GCM 算法进行一次加密和解密所花时间。可以看出, 在同样支持加密和验证两种功能的前提下, 本方案选取的 ChaCha20-Poly1305 算法在效率上

明显优于 AES256-GCM 算法, ChaCha20 算法加密过程的耗时仅为 AES 算法的 51.74%, 而解密过程也仅为 AES 算法的 31.95%, 这主要是因为树莓派等 IoT 设备的 CPU 普遍不支持对 AES 算法的硬件加速, 而 ChaCha20-Poly1305 算法对通用 CPU 进行了优化, 针对 IoT 设备展现出较高的适配性与高效性。

如图 4 所示, 在本文提出的算法中, IoT 设备需要为每一段时序数据生成对称密钥并使用对称加密算法加密该段数据, 随着时序数据数量的增加, 计算开销将会迅速增大。如果采用传统的 AES256-GCM 算法来加密、解密数据, 其耗时都将远高于 ChaCha20-Poly1305 算法, 例如当数据量达到 300 段时, 使用 AES 算法进行加密需要 138 ms, 而 ChaCha20 算法仅需 71.4 ms, 约为 AES 算法的一半。另外, 传统的对称加密方案中密钥往往是临时生成的, 虽然生成每个密钥的计算开销可忽略不计, 但会额外引入密钥管理开销, 而本方案提出的基于双向哈希链的对称密钥生成方法, IoT 设备在生成每个对称密钥时仅需 3 次哈希操作, 当需要生成 300 个密钥时, 时间开销仅为 2.349 ms, 加上加密所需的计算开销, 总开销仅为 73.6 ms, 约为 AES256-GCM 算法的 53.33%, 此外, 由于各对称密钥可由长期密钥计算派生得出, 所以该方法只会引入极小的密钥管理开销。综上所述, 本方案提出的对称密钥生成与加密算法在 IoT 时序数据安全共享场景中表现出良好的可扩展性。

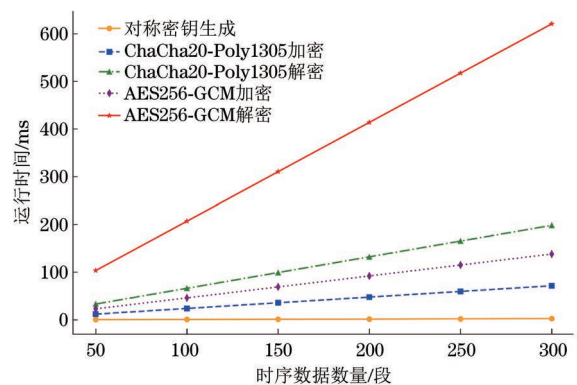


图 4 可扩展性分析结果

Fig.4 Scalability analysis results

5.3 用户侧计算开销

本节主要分析在混合签密中发生得最多的混合签密和解混合签密过程在用户侧的计算开销, 且两者都仅讨论混合(解)签密大批时序数据的计算开销, 其中前者时序数据密文生成的开销由 IoT 设备承担, 后者解密时序数据密文的开销由 DU 承担。

为模拟 IBSC^[6] 和 LHSC^[11] 中的双线性映射和椭圆曲线密码学操作,本节使用 RELIC-toolkit 密码学库。为模拟 PQSC^[13] 中的格密码学操作,本节使用微软在 Github 上开源的 Frodo640 算法代码。为模拟本方案和 MLSC^[18] 中的模块格密码学操作,本节使用 MLSC^[18] 开源的测试代码。

实验结果均为 1 000 次实验得到的均值,如表 5 所示,其中, TM_{bp} 代表执行一次双线性映射所花时间, TM_{sm}^G 代表在双线性映射群 G 上进行一次标量乘法操作所花时间, TM_{ep}^{GT} 代表在双线性映射群 G_T 上进行一次幂操作所花时间,以上 3 种操作均在曲线 BN254 上进行, TM_{ecc}^{pa} 和 TM_{ecc}^{pm} 分别代表在椭圆曲线 secp256k1 上进行一次点加和点乘操作所花时间, $TM_{ChaCha20}^{dec}$ 和 TM_{AES256}^{dec} 分别代表使用 ChaCha20-Poly1305 和 AES256-GCM 算法验证并解密 1 KB 数据所花时间, TM_{ma} 和 TM_{mm} 分别代表一次格密码学中矩阵加和矩阵乘操作所花时间, TM_{pma} 和 TM_{pmm} 分别代表一次模块格密码学中多项式矩阵加和多项式矩阵乘所花时间, TM_{hash} 代表一次哈希操作所花时间。

表 6 各方案中混合(解)签密 n 段时序数据的执行时间

Table 6 The execution time of mixing (decrypting) n segments of sequential data in various schemes

方案	离线混合签密	在线混合签密	解混合签密	期刊	年份
IBSC ^[6]	$n(TD_{AES256}^{enc} + 2 \cdot TM_{sm}^G + TM_{bp})$	—	$n(TM_{AES256}^{dec} + 3 \cdot TM_{sm}^G + TM_{bp})$	IEEE IOTJ	2024
LHSC ^[11]	$n \cdot TD_{AES256}^{enc}$	$n(4 \cdot TM_{ecc}^{pm} + 3 \cdot TM_{ecc}^{pa})$	$n(TM_{AES256}^{dec} + 8 \cdot TM_{ecc}^{pm} + 6 \cdot TM_{ecc}^{pa})$	IEEE IOTJ	2024
PQSC ^[13]	$n \cdot TD_{AES256}^{enc}$	$n(5 \cdot TM_{ma} + 4 \cdot TM_{mm})$	$n(TM_{AES256}^{dec} + 3 \cdot TM_{ma} + 2 \cdot TM_{mm})$	IEEE TIFS	2025
MLSC ^[18]	$n(TD_{AES256}^{enc} + 2 \cdot TM_{pma} + 4 \cdot TM_{pmm})$	$2n \cdot TM_{pma}$	$n(TM_{AES256}^{dec} + 4 \cdot TM_{pma} + 4 \cdot TM_{pmm})$	IEEE IOTJ	2025
本方案	$n \cdot TD_{ChaCha20}^{enc} + 2 \cdot TM_{pma} + 4 \cdot TM_{pmm}$	$2 \cdot TM_{pma}$	$n \cdot TM_{ChaCha20}^{dec} + 4 \cdot TM_{pma} + 4 \cdot TM_{pmm}$	—	—

如图5所示,相较于传统的基于双线性映射和 ECC 的混合签密方案,格基方案在混合签密阶段普遍具有更低的计算开销,当需要签密的时序数据达到 300 段时,基于双线性映射的 IBSC^[6] 需要 324.3 ms 完成混合签密,而基于椭圆曲线的 LHSC^[11] 需要 203.98 ms,基于传统格密码学的 PQSC^[13] 需要 198.09 ms,基于模块格的 MLSC^[18] 需要 178.93 ms,本方案则只需 71.54 ms, PQSC^[13]、MLSC^[18] 和本方案分别只需 IBSC^[6] 的 61.08%、55.17% 和 22.06% 的耗时即可完成,核心原因在于,格基签密方案中主要是相对普通的矩阵加法和乘法运算,不仅计算简单,而且支持并行运算,所以效率出众,传统的 ECC 和双线性映射方案中大部分计算基于数论难题,通常需要串行运行,效率相对较低。此外,基于模块格的 MLSC^[18] 和本方案还支持 PTT 计算加速技术,可以进一步提高计算效率。

表 5 各密码学操作的执行时间

Table 5 The execution time of each

cryptographic operation 单位:ms

操作	在 DO 和 DU 上的运行时间
TM_{bp}	0.327 000
TM_{sm}^G	0.147 000
TM_{ep}^{GT}	0.093 600
TM_{ecc}^{pa}	0.000 512
TM_{ecc}^{pm}	0.054 600
$TM_{ChaCha20}^{dec}$	0.004 090
TM_{AES256}^{dec}	0.002 090
TM_{ma}	0.002 700
TM_{mm}	0.046 700
TM_{pma}	0.000 208
TM_{pmm}	0.033 900
TM_{hash}	0.000 998

混合签密与解混合签密 n 段数据的计算过程时间开销组成如表 6 所示,其中混合签密被分为离线/在线两个阶段,离线签密除了为密钥明文生成离线签密密文外,还包括对称加密时序数据明文,生成对应密文。此处将哈希等开销甚微的操作忽略不计。

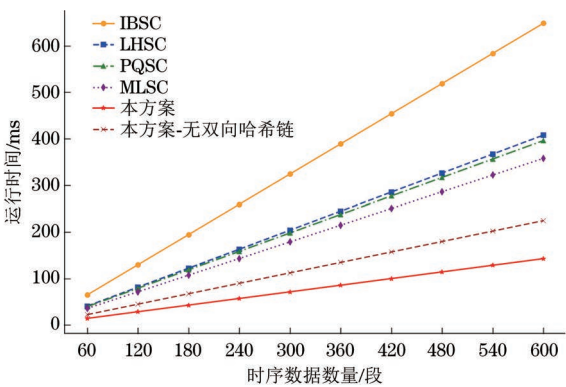


图 5 混合签密阶段的计算开销对比

Fig.5 Comparison of computational overhead in the hybrid signcryption phase

另一方面,由于传统的混合签密方案并不对混合签密中对称加密部分进行优化,所以当需要发送大量数据时,就必须重复进行签密计算,导致时间开销大大增加。而本方案在理想条件下仅需一次签密

就可以发送大量的对称加密密文,因此在混合签密过程中节省了大量的时间。在混合签密 300 段时序数据时,本方案的用时约为 71.54 ms,与同样基于格的 PQSC^[13] (198.09 ms) 和 MLSC^[18] (178.93 ms) 对比,较 PQSC^[13] 节省了 63.89% 的时间,较 MLSC^[18] 节省了 60.02% 的时间。当本方案不使用双向哈希链机制时,用时约为 112.33 ms,约为使用时的 1.57 倍,可见本文提出的基于双向哈希链生成对称密钥的设计,能够显著提升时序数据共享的效率。

如图 6 所示,在解混合签密阶段,当时序数据数量快速增长时,3 种格基方案展现出了明显优势,其中本方案由于只需一次解签密,使得性能尤其突出。当时序数据的数量达到 600 段时,本方案解混合签密的总耗时约为 2.59 ms,分别为 PQSC^[13] 的 4.16% 和 MLSC^[18] 的 3.12%,尽管在用户侧解密 ChaCha20-Poly1305 算法加密的密文所需时间约为解密 AES256-GCM 算法加密的密文的 1.96 倍,在解混合签密大量时序数据时,本方案相较其他两种方案依旧有较大优势,凸显基于双向哈希链的密钥生成方法的高效性,而当本方案不使用双向哈希链机制时,运行时间将会超过 PQSC^[13],总耗时将达到 84.31 ms,为使用双向哈希链时的 32.55 倍,进一步凸显了双向哈希链机制的必要性。

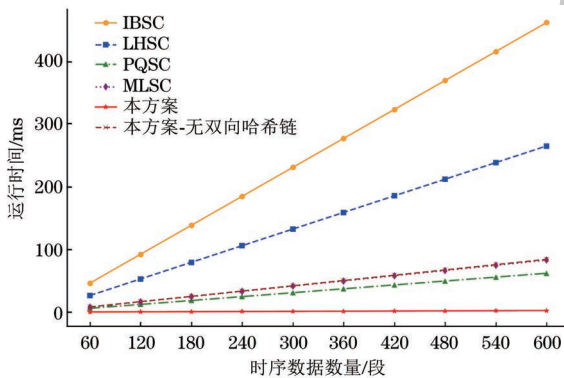


图 6 解混合签密阶段的计算开销对比

Fig.6 Comparison of computational overhead in the decryption and hybrid signcryption stages

5.4 用户侧通信开销

将本方案与其他 4 种对比方案在混合签密阶段随着时序数据数量增长而变化的通信开销进行对比,结果如表 7 所示,其中 Num 表示时序数据段数, G 表示双线性映射群上的元素长度,为 520 bit, P 表示椭圆曲线上点的长度,为 320 bit, Z 表示一个大整数的长度,为 160 bit, n 和 q 分别表示传统格密码学方案中格的维度和模数,由于 PQSC^[13] 并未给出具体数值,因此本节采用 Frodo640 算法的参数,令 $n=640$, $q=2^{16}=65\,536$ 。 $d=1$, $k=16$, $l=$

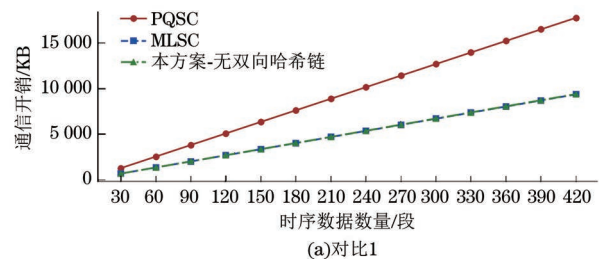
15 , $n'=1$ 为前文所述的模块格的维度参数; $N=1\,024$ 和 $q'=2^{17}-1=131\,071$ 为前文所述模块格的度和模数; M_{key} 和 M_{data} 分别表示密钥密文的长度和一段时序数据密文的长度,分别为 256 bit 和 1 KB。

表 7 各方案的通信开销

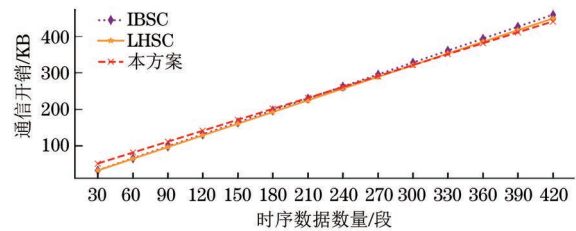
Table 7 Communication overhead of each scheme

方案	混合签密通信开销/bit
IBSC ^[6]	$Num(G+M_{key}+M_{data})$
LHSC ^[11]	$Num(P+Z+M_{data})$
PQSC ^[13]	$Num(2n\lg^2 q+n\lg q+l\lg q+M_{data})$
MLSC ^[18]	$Num(2[d(2+k-l)+n'+1]N\lg q'+M_{key}+M_{data})$
本方案	$2[d(2+k-l)+n'+1]N\lg q'+3\cdot M_{key}+Num\cdot M_{data}$

如图 7 所示,为了抵御量子计算攻击,基于格密码学的签密方案通常需在签密后得到大量高维矩阵,产生非常大的通信开销。如 PQSC^[13] 在签密任一段数据时都会添加额外的约 41.25 KB 的附加信息,当需要发送的时序数据量达到 150 段时, PQSC^[13] 的通信开销高达 5 070.23 KB, MLSC^[18] 的通信开销达到了 3 342.19 KB,而其中有价值的时序数据明文长度也仅有 150 KB,仅占 PQSC^[13] 总通信开销的 2.96%,占 MLSC^[18] 的 4.49%。本方案由于提出了基于双向哈希链的对称密钥生成方法,数据共享仅需一次混合签密,故通信开销仅为 171.34 KB,其中有效的时序数据占比 87.55%,分别是 PQSC^[13] 的 3.38% 与 MLSC^[18] 的 5.13%。此外,当本方案不使用双向哈希链机制时,通信开销略多于 MLSC^[18],约为使用双向哈希链机制时的 19.53 倍。由此可见,本方案在发送大量时序数据时,相较一般的基于格的混合签密方案,具有非常大的优势。



(a)对比1



(b)对比2

图 7 签密阶段的通信开销对比

Fig.7 Comparison of computational overhead of the signcryption stage

另一方面,由于传统的基于双线性映射和 ECC 的混合签密方案的密文中包含的椭圆曲线上的元素远小于格密码学方案的密文中包含的高维矩阵,所以传统方案在通信开销上普遍有较大优势,但是它们不支持后量子安全,主要是因为 Shor 算法为快速解决离散对数难题提供了可行的方案,使得传统方案中仅凭几组椭圆曲线上的元素就可隐藏信息的方法变得不再安全,但 Shor 算法乃至其他现存的量子算法都无法破解格密码学方案引入的噪声,所以密文的安全性能得到保障。然而,如果噪声余量不足,合法用户同样无法解密密文,所以,为了确保密文能够正确得到解密,格密码学方案必须引入高维矩阵,保障充足的噪声余量,但这也使得密文的尺寸相较传统方案变得巨大。为降低通信开销,本方案设计了基于双向哈希链的混合签密,在发送的数据段数增加时,签密次数不变,减少了需要的高维矩阵,取得了优良的效果。例如,当需要发送的时序数据为 30 段时,IBSC^[6]中总通信开销约为 32.84 KB, LHSC^[11]总通信开销约为 31.76 KB,本方案的总通信开销约为 51.34 KB,分别是 IBSC^[6]的 156.33% 以及 LHSC^[11]的 161.65%。但是,当需要混合签密的时序数据达到 420 段时,IBSC^[6]中的通信开销就会达到 459.79 KB, LHSC^[11]则为 444.61 KB,而本方案的总通信开销约为 441.34 KB,优于前两者。当数据量继续增加时,此优势会进一步扩大,这同样得益于本方案在混合签密中引入的基于双向哈希链生成对称密钥的设计,发送大量数据时只需一次签密,避免引入体积庞大的高维矩阵,大幅降低了重复签密不同密钥带来的开销。

6 结束语

本文针对海量、连续时序数据的安全共享需求,提出一种面向时序数据共享的后量子轻量化无证书混合签密方案。该方案在模块格框架下构建抗量子安全基础,结合双向哈希链密钥生成机制与混合签密结构,实现了密钥的高效生成与安全分发,有效降低了端侧计算与通信开销。本方案在随机预言机模型下满足 IND-CCA 机密性和 EUF-CMA 不可伪造性,并具备良好的前向与后向安全特性。此外,本方案在计算效率与通信开销方面均显著优于现有后量子无证书签密方法,在用户端可将总体开销降低约 60%,充分体现了其轻量化与可扩展优势,为构建面向智能物联网的安全通信与数据协同机制提供了可行方案。

然而,当前工作主要面向一对一数据共享场景,

在通信模式灵活性方面仍存在不足。未来将探索面向一对多通信的多接收者 CLSC 以及多对一通信的支持批量验证的后量子 CLSC 等领域,以进一步提高方案的实用性。

参考文献

- [1] CHENG G J, HUANG J Q, WANG Y W, et al. Conditional privacy-preserving multi-domain authentication and pseudonym management for 6G-enabled IoV [J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 10206-10220.
- [2] WEI Y Y, JANG J J, XU W, et al. LSTM-autoencoder-based anomaly detection for indoor air quality time-series data[J]. IEEE Sensors Journal, 2023, 23(4): 3787-3800.
- [3] CHENG G J, WANG Y W, DENG S G, et al. A lightweight authentication-driven trusted management framework for IoT collaboration[J]. IEEE Transactions on Services Computing, 2024, 17(3): 747-760.
- [4] 李哲成, 张波. 基于区块链的轻量级工业物联网跨域认证与数据共享方案[J]. 计算机研究与发展, 2025, 62(10): 2416-2427.
LI Z C, ZHANG B. A lightweight blockchain-based cross-domain authentication and data sharing scheme for the industrial Internet of Things [J]. Journal of Computer Research and Development, 2025, 62(10): 2416-2427. (in Chinese)
- [5] 陈骁, 黄牧鸿, 田一凡, 等. 基于分片区块链的车联网数据共享方案[J]. 计算机研究与发展, 2024, 61(9): 2246-2260.
CHEN X, HUANG M H, TIAN Y F, et al. Internet of vehicles data sharing scheme via blockchain sharding [J]. Journal of Computer Research and Development, 2024, 61(9): 2246-2260. (in Chinese)
- [6] WANG Y H, JIA X Y, BAO Y Y, et al. Efficient and provably secure offline/online heterogeneous signcryption scheme for VANETs[J]. IEEE Internet of Things Journal, 2024, 11(24): 41248-41260.
- [7] DU J W, AI B, LIANG J Y, et al. An adaptive secure identity-based signcryption with equality test for high-speed railway train-ground communication[J]. IEEE Transactions on Network Science and Engineering, 2026, 13: 1872-1886.
- [8] RAMADAN M, RAZA S. Secure equality test technique using identity-based signcryption for telemedicine systems [J]. IEEE Internet of Things Journal, 2023, 10(18): 16594-16604.
- [9] ZHOU Y W, XU R, QIAO Z R, et al. An anonymous and efficient multimessage and multireceiver certificateless signcryption scheme for VANET [J]. IEEE Internet of Things Journal, 2023, 10(24): 22823-22835.
- [10] ZHANG J H, DONG C H, LIU Y N. Efficient pairing-free certificateless signcryption scheme for secure data transmission in IoMT[J]. IEEE Internet of Things Journal, 2024, 11(3): 4348-4361.
- [11] HAN Y, ZHOU Y W, YANG B, et al. An efficient and secure lightweight certificateless hybrid signcryption scheme[J]. IEEE Internet of Things Journal, 2024, 11(7): 12252-12265.
- [12] 杨小东, 周航, 任宁宁, 等. 支持多密文等值测试的无线体域网聚合签密方案[J]. 计算机研究与发展, 2023, 60(2): 341-350.
YANG X D, ZHOU H, REN N N, et al. Aggregate signcryption scheme supporting multi-ciphertext equality test for wireless body area network [J]. Journal of Computer Research and Development, 2023, 60(2): 341-350. (in Chinese)
- [13] XU S Y, CHEN X, GUO Y, et al. Efficient and secure post-quantum certificateless signcryption with linkability for IoMT[J]. IEEE Transactions on Information Forensics and Security, 2025,

- 20: 1119-1134.
- [14] JIAO J T, GUO L, YU W S, et al. An efficient lattice-based heterogeneous signcryption scheme for VANETs [J]. *Concurrency and Computation: Practice and Experience*, 2025, 37(3): e8384.
- [15] XIAO K F, CHEN X J, LI H B, et al. A fully secure lattice-based signcryption with designated equality test in standard model[J]. *Information Sciences*, 2024, 658: 120015.
- [16] YU H F, DANG N Z, YANG B. Blockchain-based insurance claim scheme using proxy signcryption from NTRU lattice[J]. *IEEE Sensors Journal*, 2025, 25(17): 33801-33809.
- [17] YAN J H, LU X H, LI M Z, et al. Practical NTRU signcryption in the standard model [J]. *Entropy*, 2023, 25(12): 1651.
- [18] BAI Y, HE D B, YANG Z C, et al. Efficient module-lattice-based certificateless online/offline signcryption scheme for Internet of medical things [J]. *IEEE Internet of Things Journal*, 2025, 12(14): 27350-27363.
- [19] ZHENG Y L. Digital signcryption or how to achieve cost (signature & encryption) $\ll$ cost (signature) + cost (encryption)[EB/OL]. [2025-12-05]. https://link.springer.com/chapter/10.1007/978-3-540-40061-5_29.
- [20] AL-RIYAMI S S, PATERSON K G. Certificateless public key cryptography [EB/OL]. [2025-12-05]. https://link.springer.com/chapter/10.1007/978-3-540-40061-5_29.
- [21] SHOR P W. Algorithms for quantum computation: discrete logarithms and factoring[C]//*Proceedings of the 35th Annual Symposium on Foundations of Computer Science*. Washington D. C., USA: IEEE Press, 2002: 124-134.
- [22] BAVDEKAR R, JAYANT CHOPDE E, AGRAWAL A, et al. Post quantum cryptography: a review of techniques, challenges and standardizations [C] // *Proceedings of the International Conference on Information Networking (ICIN)*. Washington D. C., USA: IEEE Press, 2023: 146-151.
- [23] JEUDY C, ROUX-LANGLOIS A, SANDERS O. Phoenix: hash-and-sign with aborts from lattice gadgets [EB/OL]. [2025-12-05]. <https://eprint.iacr.org/2023/446>.
- [24] CHEN Y L, GENISE N, MUKHERJEE P. Approximate trapdoors for lattices and smaller hash-and-sign signatures [EB/OL]. [2025-12-05]. <https://eprint.iacr.org/2019/1029.pdf>.
- [25] BRAKERSKI Z, GENTRY C, VAIKUNTANATHAN V. (leveled) fully homomorphic encryption without bootstrapping [J]. *ACM Transactions on Computation Theory*, 2014, 6(3): 1-36.
- [26] LANGLOIS A, STEHLÉ D. Worst-case to average-case reductions for module lattices [J]. *Designs, Codes and Cryptography*, 2015, 75(3): 565-599.
- [27] NIR Y, LANGLEY A. ChaCha20 and Poly1305 for IETF protocols[EB/OL]. [2025-12-05]. <https://www.rfc-editor.org/info/rfc7539>.
- [28] NAJM Z, JAP D, JUNGK B, et al. On comparing side-channel properties of AES and ChaCha20 on microcontrollers [C] // *Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*. Washington D. C., USA: IEEE Press, 2019: 552-555.
- [29] BARKER E, ROGINSKY A. NIST special publication 800-131A: transitioning the use of cryptographic algorithms and key lengths [EB/OL]. [2025-12-05]. <https://doi.org/10.6028/NIST.SP.800-131Ar2>.
- [30] POINTCHEVAL D, STERN J. Security proofs for signature schemes[EB/OL]. [2025-12-05]. https://www.di.ens.fr/david.pointcheval/Documents/Papers/1996_eurocrypt.pdf.

文字编辑 吴云芳
栏目编辑 宋 圆