Olithium:基于格的无陷门在线/离线签名方案

王后珍[1)] 段小超[1)] 文嘉明[1),2)] 王亚辉[1),3)] 张焕国[1)]

[1)](武汉大学国家网络安全学院空天信息安全与可信计算教育部重点实验室 武汉 430072)

[2)](伍伦贡大学计算与信息技术学院 伍伦贡 2522 澳大利亚)

[3)](信阳师范大学计算机与信息技术学院 河南 信阳 464000)

摘 要 随着互联网和大数据时代的到来,数据在传输过程中的安全性和效率问题日益凸显。作为保障数据完整性并验证发送者身份的核心技术,数字签名显得尤为重要。与此同时,量子计算机的发展对传统的签名方案构成了巨大威胁。在这一背景下,美国国家标准与技术研究院于2023年公布了基于CRYSTALS-Dilithium算法的抗量子签名标准ML-DSA的草案。为了使其能更好地应用于海量的数据传输等场景,本文在此基础上设计了一个无陷门在线/离线签名方案,称之为Olithium。该方案允许签名者在未收到消息时(离线阶段)产生签名的一部分,并在收到消息后(在线阶段)继续完成签名。最终,能在存储空间小幅增加的前提下,将在线签名时间缩短约50%。本文还以数字证书颁发场景为例,说明该方案的现实有效性。

关键词 后量子密码;在线/离线签名;数字证书;基于格的密码学;优化实现

中图法分类号 TP309 **DOI号** 10.11897/SP.J.1016.2025.00877

Olithium: A Lattice-Based Online/Offline Signature Scheme Without Trapdoors

WANG Hou-Zhen[1)] DUAN Xiao-Chao[1)] WEN Jia-Ming[1),2)] WANG Ya-Hui[1),3)] ZHANG Huan-Guo[1)]

[1)](Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072)

[2)](School of Computing and Information Technology, University of Wollongong, Wollongong 2522, Australia)

[3)](School of Computer and Information Technology, Xinyang Normal University, Xinyang, Henan 464000)

Abstract With the rapid advancement of the internet and the era of big data, ensuring both the security and efficiency of data transmission has become a critical concern. Digital signatures, as a fundamental cryptographic primitive, play a pivotal role in maintaining data integrity and verifying the authenticity of the sender. However, the emergence of quantum computers threatens the security of traditional signature schemes that rely on hardness assumptions such as integer factorization and discrete logarithm problems. To address this challenge, the U. S. National Institute of Standards and Technology (NIST) has initiated a post-quantum cryptography (PQC) standardization process, leading to the proposal of ML-DSA, a lattice-based digital signature scheme derived from CRYSTALS-Dilithium, as a post-quantum signature standard. On the domestic front, the Chinese Association for Cryptologic Research (CACR) held the national

收稿日期:2024-11-07;在线发布日期:2025-02-13。本课题得到国家重点研发计划项目(2022YFB4500800)、湖北省重点研发计划项目(2022BAA041)、国家留学基金委联合培养博士生项目(202306270167)资助。**王后珍**,博士,副教授,硕士生导师,中国计算机学会(CCF)会员,主要研究领域为密码学、抗量子密码。E-mail:whz@whu.edu.cn。**段小超**,硕士研究生,主要研究领域为密码学、抗量子密码。**文嘉明**(通信作者),博士研究生,主要研究领域为密码学、抗量子密码。E-mail:wenjm@whu.edu.cn。**王亚辉**,博士,讲师,中国计算机学会(CCF)会员,主要研究领域为密码学、量子计算。**张焕国**,博士,教授,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为密码学、信息安全。

cryptographic algorithm design competition in 2019, where the Aigis-sig signature scheme was the only first prize in the signature category. Aigis-sig adopts the similar structure as Dilithium and has received high recognition in China. With the advancement of digital signature standardization, extensive research has begun exploring more versatile signature schemes based on these standards, as well as more efficient and secure implementations. These efforts contribute to enhancing the practical applicability of post-quantum schemes in reality. Based on Dilithium, we propose Olithium, a lattice-based online/offline signature scheme without trapdoors, which optimizes the signing efficiency. Olithium follows the online/offline paradigm, where part of the signature is precomputed in an offline phase before receiving the message, significantly reducing the signing time in the online phase. Compared to Dilithium, Olithium achieves approximately a 50% reduction in online signing time while maintaining the same level of security. This improvement is particularly valuable for applications that require frequent and real-time signing, such as large-scale data transmission and certificate issuance. Technically, Olithium is designed based on a careful analysis of the computational bottlenecks of the signing stage in Dilithium. One of the key optimizations involves modifying the underlying zero-knowledge protocols to allow pre-computing the computationally expensive part in the offline phase. Additionally, Olithium refines the hint generation of Dilithium to further minimize the computational overhead during online signing. From the security perspective, Olithium retains the same security guarantees as Dilithium, ensuring strong unforgeability under chosen-message attacks in the quantum random oracle model (QROM). To validate the efficiency of Olithium, we implemented it and released the code as open source. The experimental results show that when compared with Dilithium, Olithium reduces online signing time by approximately 50% without significantly increasing storage overhead. Finally, we further illustrate the real-world applicability of Olithium, including its practical deployment and a case study on digital certificate issuance. In summary, Olithium represents an attempt in the field of post-quantum digital signatures by introducing an online/offline modification by leveraging zero-knowledge proof techniques and cryptographic engineering optimizations. Compared to Dilithium, Olithium achieves a significant reduction in signing time while maintaining the same level of security and without substantially increasing storage overhead. In the future, we aim to further optimize its implementation for constrained environments, explore its integration with other cryptographic primitives, and enhance its practical applicability in real-world scenarios.

Keywords post-quantum cryptography; online/offline signatures; digital certificates; lattice-based cryptography; optimized implementation

1 引 言

随着互联网技术的飞速发展和大数据时代的到来，海量数据在工业生产中扮演着越来越关键的角色，并为人们的日常生活带来诸多便利。在享受数据带来的好处的同时，确保数据安全变得尤为重要。例如，保护数据的完整性和验证数据发送者的真实身份，都已成为数据能被有效利用的基石。除此以外，研究如何提高数据处理效率、降低数据存储和传输的开销也具有重要的意义。为了应对相关挑战，各种密码技术应运而生。

数字签名技术，凭借其在检查消息完整性和实现身份认证方面的独特优势，已经在互联网中得到广泛应用。其中，前者保障数据在传输过程中未被篡改，而后者则确保数据来源可靠、有效遏制了未授权访问和欺诈行为的发生。目前广泛使用的数字签名标准（如RSA、ECDSA、EdDSA）安全性大多基于整数分解和离散对数等传统计算机难以解决的数学困难问题。然而，针对这些数学问题及密码方案，

已经出现了有效的量子算法。例如，Shor[1]提出的多项式时间量子算法，可以用于解决整数分解和离散对数问题，以及Wang等人[2]提出的基于方程求解与相位估计的量子算法，可以用于攻击RSA。这意味着一旦量子计算机在未来得以实用，这些困难问题以及基于它们的密码方案将面临前所未有的安全威胁。与此同时，量子计算机的研究也在不断取得进展[3]。

为了应对迫在眉睫的威胁，美国国家标准与技术研究院（National Institute of Standards and Technology，NIST）在2016年启动抗量子密码项目，旨在征集抗量子的密钥封装和数字签名标准，用于保障未来社会的信息安全。数字签名方面，基于格问题设计的无陷门签名方案CRYSTALS-Dilithium备受推荐，并在2023年颁布的FIPS 204草案中被标准化成为ML-DSA签名标准①。2024年11月，NIST再次发布相关政策文件②，在其中详细阐述现有密码标准的脆弱性以及采用抗量子密码标准的迫切性，其结论包括RSA，ECDSA，EdDSA等签名标准应在2030年后不再新使用，2035年后禁止使用，取而代之采用基于格的签名标准ML-DSA和基于哈希的签名标准SLH-DSA。国内方面，中国密码学会也于2019年举办全国密码算法设计竞赛，签名组唯一获得一等奖的作品Aigis-sig签名方案采用与Dilithium相同的无陷门结构，在国内的抗量子密码设计领域获得了高度评价，经过进一步的改进后，有望成为我国主导的重要抗量子密码方案。

随着数字签名标准化的推进，许多学者开始研究基于它们的功能更为丰富的签名方案，例如盲签名[4]和两方协同签名[5]。此外，还针对这些签名方案的高效实现和安全实现（分析其侧信道泄露等）展开了深入研究，例如素阶数域上的实现[6]和对能耗分析攻击的防护[7]，这些成果都有助于提升抗量子密码方案在现实世界的可用性。

为了进一步提高后量子时代签名效率，本文设计了一个新颖的在线/离线签名方案。该方案能在未收到待签名消息时（离线阶段）先产生签名的一部分，从而显著缩短收到消息后（在线阶段）签名所需的时间。虽然小幅增加了存储空间，但能极大提高签名效率，使得其更加适合于大数据和云计算等需要快速处理海量数据的场景。具体而言，本文有如下的主要贡献：

（1）设计了一个基于格的无陷门在线/离线签名方案—Olithium，可以被视为对CRYSTALS-Dilithium方案的优化改进。改进过程的关键在于修改底层的零知识协议和重新设计签名过程中的提示位生成子算法，增加它们与在线/离线签名设定的适配性，从而显著提升（在线）签名效率。

（2）为Olithium方案提供详细的安全证明，结论表明其安全性与Dilithium方案相当，在量子随机谕言模型下满足强存在性不可伪造性。

（3）对Olithium方案进行编程实现和仿真，通过与Dilithium方案进行对比以说明其有效性。总体而言，不同安全等级的平均签名时间能缩短约50%。此外，将实验源代码公开，以促进后续的研究和技术应用。

（4）以数字证书颁发的应用场景作为例子，说明Olithium方案在现实场景中的实用性。即使在未来量子计算机实用化后，该方案仍能为证书颁发等现实应用场景提供可靠保障。

值得一提的是，本文的改进方法也适用于结构与CRYSTALS-Dilithium签名方案类似的国产抗量子签名方案Aigis-sig。因此，本文的研究工作对国产抗量子密码算法的发展也具有重要意义。

2　相关工作

在线/离线签名的概念最早由Even、Goldreich和Micali提出[8]，是一种变形的签名方案。其签名算法由两个阶段组成，第一阶段在收到待签名消息之前离线进行，而第二阶段在收到消息后在线进行。其主要思想在于通过离线阶段的预计算来减少在线阶段的签名时间。现有的在线/离线签名方案，可以粗略按照hash-sign-switch范式和Fiat-Shamir范式两条技术路线进行分类。

2.1　基于hash-sign-switch范式的在线/离线签名

作者在提出在线/离线签名的概念后[8]，基于一次性签名方案设计了符合定义的方案。然而，其签名尺寸增长为原来的平方级别，通信开销的显著增加使其难以满足实际应用需求。

在2001年的美密会上，Shamir和Tauman在此基础上提出了一种通用转换[9]，称为hash-sign-switch范式。通过结合一次性签名和变色龙Hash

① NIST，FIPS 204. Module-Lattice-Based Digital Signature Standard，https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.ipd.pdf

② NIST，IR 8547. Transition to Post-Quantum Cryptography Standards，https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8547.ipd.pdf

函数，使每个签名的尺寸仅增加为原来的两倍，并通过增强离线阶段的随机性进一步提高了整个方案的安全性。不久后，该范式被实例化于经典的强RSA假设上，并在此基础上进一步改进[10]。然而，后续研究表明这种范式无法避免变色龙Hash函数的密钥暴露问题[11]。具体而言，如果签名者多次使用相同的Hash值来获得不同消息对应的签名，则验证者可以由此找到该变色龙Hash函数的碰撞，并利用它恢复签名者的陷门信息，这使得签名者必须在离线阶段预先计算并存储大量不同的变色龙Hash值以及这些Hash值对应的签名，并在在线阶段全部发送，从而导致之前未考虑到的通信开销显著增加。在引入特殊的基于离散对数的双陷门Hash函数[12]，虽然可以克服这个问题，但对计算效率有着严重影响。

2018年，Zheng等人[13]将经过安全性修复的hash-sign-switch范式在格上进行了实例化。然而，除了和基于经典问题的实例一样存在计算效率较低的问题之外，签名尺寸也增加数倍，对于设计实用化抗量子签名是难以接受的。

2.2 基于Fiat-Shamir范式在线/离线签名

基于Σ-协议以及Fiat-Shamir变换[14]得到的无陷门签名方案的第一步并不依赖于待签名的消息，因此可以将它们直接拆分为离线阶段(第一步)和在线阶段(后续步骤)。然而，这样将所有后续步骤都放在在线阶段运行，花费的在线时间相比起不拆分提升并不明显。此外，通过该范式设计基于格的签名方案时还需要额外地拒绝抽样(中止)以防止密钥泄露[15]，即要求签名值不落在特定范围时就重启整个签名过程。若不加修改地直接将基于格的Σ-协议拆分并转换为在线/离线形式，只会导致更差的签名效率。

为了增强由Fiat-Shamir范式得到的签名方案与在线/离线签名的适配性，Yao和Zhao将Σ-协议进行修改得到Γ-协议[16]，并给出了与Fiat-Shamir变换相对应的Γ-变换，能直接用来将Γ-协议转换为在线/离线签名。相比起直接基于Σ-协议，使用Γ-协议能使更多的计算在离线阶段进行，提升了在线阶段的效率。

2021年，Zhang等人[17]依照Γ-协议的设定，将其从离散对数困难问题迁移到了基于格问题，并以NIST第二轮的候选算法qTESLA[18]为基础设计了一个基于格的在线/离线签名方案，提升其在线性能。2023年，基于Γ-协议提出新的改进[19]，用于设计依赖于环上的格问题的在线/离线签名方案。然而，这些方案的性能分析仅停留在理论层面，缺乏具体的编程实现、测试和模拟仿真。

根据上述调研可知，目前仅有的几个基于格的在线/离线签名方案[13, 17, 19-20]尚有不足之处。随着量子计算机的到来，以NIST候选签名算法为基础，特别是获胜并被标准化的ML-DSA算法(原型为Dilithium方案)，设计在线/离线签名方案并进行优化改进，同时通过充分的实验评估其效果，是一项具有重要理论和现实意义的研究。

3 预备知识

本节介绍与本文方案相关的预备知识，主要包括所需要用到的符号说明，格上的困难问题，在线/离线签名方案的算法定义和安全模型，以及CRYSTALS-Dilithium数字签名方案。

3.1 符号说明

本文用$\boldsymbol{A}^T$表示矩阵$\boldsymbol{A}$的转置，$\boldsymbol{I}_m$表示m阶单位矩阵，$\lceil x \rceil$表示不小于实数x的最小整数，lb表示以2为底的对数。

R和R_q分别表示多项式环$\mathbb{Z}[x]/(x^n+1)$和$\mathbb{Z}_q[x]/(x^n+1)$，其中n为正整数，$n-1$为环中的多项式的最高次数，实际方案中n一般选取为2的幂次(例如$n=256$)；q表示环R_q的模数，一般选取较大的素数，在选取时需要综合考虑其他参数和工程实现的效率。

对于多项式$w=w_0+w_1x+\cdots+w_{n-1}x^{n-1}\in R_q$，其中$w_i$为整数，用$\|w\|_\infty=max_i|w_i|$表示其$\ell_\infty$范数，用$\|w\|_2=\sqrt{\sum_i|w_i|^2}$表示其$\ell_2$范数。对于由多项式组成的向量$\boldsymbol{p}=(p_1,p_2,\cdots,p_k)^T\in R^k$，其中$p_i$为环$R$中的多项式，用$\|\boldsymbol{p}\|_\infty=max_i\|p_i\|_\infty$表示其$\ell_\infty$范数，用$\|\boldsymbol{p}\|_2=\sqrt{\sum_i\|p_i\|^2}$表示其$\ell_2$范数。环$R_q$中的范数定义类似。

对于整数w_i，用$w_i'=w_i\,mod^+\,\alpha$表示满足$w_i'=w_i\,mod\,\alpha$且落在区间$0\leqslant w_i'<\alpha$的整数w_i'。类似地，用$w_i'=w_i\,mod^\pm\,\alpha$表示落在区间$-\frac{\alpha}{2}<w_i'\leqslant\frac{\alpha}{2}$(若$\alpha$为偶数)和$-\frac{\alpha-1}{2}\leqslant w_i'\leqslant\frac{\alpha-1}{2}$(若$\alpha$为奇数)的整数$w_i'$。对于多项式$w$，$mod$运算对其每一个系数独立作用。

对于$\eta>0$，用S_η表示由所有的满足$\|w\|_\infty\leqslant$

η的多项式$w\in R_q$组成的集合，并用$\tilde{S}_\eta$表示集合$\{w\ mod^{\pm}2\eta: w\in R_q\}$。

对于集合R_q和分布D，用$x\leftarrow_{\$}R_q$表示从$R_q$中均匀随机选取$x$，用$s\leftarrow R_q$表示按照分布$D$选取$s$。对于矩阵或向量，运算对其每一个系数独立作用。

3.2　格相关的困难问题

设计基于格的密码方案时，常用的困难问题包括小整数解问题(Short Integer Solution，SIS)以及错误学习问题(Learning with Errors，LWE)等。为了压缩尺寸，也有工作对它们在特定代数结构(例如环或者模)上的扩展进行了深入研究。本文设计的方案和Dilithium一样，基于模格问题。

定义 1.　模上的错误学习问题(Module-$\mathrm{LWE}_{q,m,k,D}$)[21]. 给定模数q，正整数m,k，矩阵$\boldsymbol{A}\leftarrow_{\$}R_q^{m\times k}$和概率分布$D: R_q\rightarrow[0,1]$，定义敌手算法$\mathcal{A}$的优势为

$$Adv^{\mathrm{MLWE}}_{q,m,k,D}(\mathcal{A}):=\left|\Pr\left[b=1\middle|b\leftarrow\mathcal{A}(\boldsymbol{A},\boldsymbol{t})\right]-\Pr\left[b=1\middle|b\leftarrow\mathcal{A}(\boldsymbol{A},\boldsymbol{As}_1+\boldsymbol{s}_2)\right]\right|$$

其中，$\boldsymbol{t}\leftarrow_{\$}R_q^m$；$\boldsymbol{s}_1\leftarrow D^k$；$\boldsymbol{s}_2\leftarrow D^m$。

对于特定参数，该优势对敌手算法$\mathcal{A}$是可忽略的。

定义 2.　模上的小整数解问题(Module-$\mathrm{SIS}_{q,m,k,\gamma}$)[21]. 给定模数$q$，正整数$m,k$，矩阵$\boldsymbol{A}\leftarrow_{\$}R_q^{m\times k}$和$\gamma>0$，定义敌手算法$\mathcal{A}$的优势为

$$Adv^{\mathrm{MSIS}}_{q,m,k,\gamma}(\mathcal{A}):=$$

$$\Pr\left[\boldsymbol{y}\leftarrow\mathcal{A}(\boldsymbol{A}): 0<\|\boldsymbol{y}\|_\infty\leqslant\gamma\wedge[\boldsymbol{I}_m|\boldsymbol{A}]\cdot\boldsymbol{y}=0\right]$$

对于特定参数，该优势对敌手算法$\mathcal{A}$是可忽略的。

3.3　在线/离线签名的定义与模型

在线/离线签名在数字签名的基础上，将签名阶段分为了两个阶段：离线阶段和在线阶段。第一个阶段是离线阶段，签名者可以在收到待签名的消息之前离线预计算。第二个阶段是在线阶段，一旦收到待签名的消息，签名者就会快速生成签名，定义如下。

定义 3.　在线/离线签名[8]. 在线/离线签名方案包括三个概率多项式时间算法：密钥生成(*KeyGen*)、签名(*Sign*)和验证(*Verify*)，定义如下：

(1) $KeyGen(1^\lambda)$：该算法输入安全参数λ，输出一个随机的公钥/私钥对(pk,sk)。

(2) $Sign(sk,pk,\mu)$：该算法分为两个阶段

① 离线阶段(预计算)：输入密钥对(pk,sk)，输出中间值φ。

② 在线阶段(签名产生)：输入φ和消息μ，输出签名σ。

(3) $Verify(pk,\mu,\sigma)$：输入公钥pk和一个消息/签名对(μ,σ)，如果σ是μ的合法签名，则输出1(接受)；否则输出0(拒绝)。

在安全模型上，本文沿用常见的关于签名的安全模型，即选择消息攻击下的强存在性不可伪造性(Strong Existential Unforgeability under Chosen Message Attack，SUF-CMA)，定义如下：

定义 4.　选择消息攻击下的强存在性不可伪造性(SUF-CMA)[20]. 给定在线/离线签名方案($KeyGen, Sign, Verify$)，定义敌手$\mathcal{A}$的优势

$$Adv^{\mathrm{SUF-CMA}}_{\mathrm{SIG}}=\Pr\left(\begin{array}{r}(pk,sk)\leftarrow\mathrm{KeyGen}(\lambda),\\ Verify(pk,M^*,\sigma^*)=1\ \middle|\ (M^*,\sigma^*)\leftarrow\mathcal{A}^{\mathcal{O}_s}(pk),\\ (M^*,\sigma^*)\notin L_{\mathrm{query}}\end{array}\right)$$

对于安全的签名方案，该优势对敌手算法$\mathcal{A}$是可忽略的。其中$\mathcal{O}_s$为签名谕言机，L_{query}为对应的Hash列表。

3.4　CRYSTALS-Dilithium数字签名方案

作为已经被标准化为ML-DSA签名算法的原型，CRYSTALS-Dilithium方案是基于Fiat-Shamir范式的无陷门签名方案。其算法设计文档[21]已经指出，该方案的优点包括但不限于容易安全地实现，不同级别的安全性之间容易切换等。在介绍Dilithium签名的具体方案之前，先在此简单描述其中需要用到的支撑算法。将它们作用到多项式、向量和矩阵时，表示将对应操作分别独立地作用到每个系数。

支撑算法 1. 二次幂分解算$Power2Round_q(r,d)$

该算法输入整数$r\in\mathbb{Z}_q$和一个小的正整数d，分解得到(r_1,r_0)，满足$r=r_1\cdot 2^d+r_0$，其中$\|r_0\|_\infty\leqslant 2^{d-1}$。按照下述3个步骤依次运行：

1. 计算$r\ mod^+ q$落到区间$0\leqslant r<q$，得到$r:=r\ mod^+ q$；
2. 计算$r\ mod^{\pm}2^d$落到区间$-2^{d-1}<r\leqslant 2^{d-1}$，得到$r_0:=r\ mod^{\pm}2^d$；
3. 令$r_1:=(r-r_0)/2^d$，输出(r_1,r_0)。

支撑算法 2. 高低位比特分解算法$Decompose_q(r,\alpha)$

该算法输入整数$r\in\mathbb{Z}_q$和一个小的正整数α，满足$\alpha|(q-1)$，将r分解为$r=r_1\cdot\alpha+r_0$，其中$0\leqslant r_1<\dfrac{q-1}{\alpha}$，$\|r_0\|_\infty\leqslant\dfrac{\alpha}{2}$。并称$HighBits_q(r,\alpha)=r_1$

为r的高位比特，$LowBits_q(r,\alpha)=r_0$为r的低位比特。按照下述3个步骤依次运行：

1. 计算$r \bmod^+ q$落到区间$0\leqslant r<q$中，得到r：$=r \bmod^+ q$；
2. 计算$r \bmod^\pm \alpha$落到区间$-\frac{\alpha}{2}<r\leqslant\frac{\alpha}{2}$(或$-\frac{\alpha-1}{2}\leqslant r\leqslant\frac{\alpha-1}{2}$)中，得到$r_0$：$=r \bmod^\pm \alpha$；
3. 如果$r-r_0=q-1$则令r_1：$=0, r_0$：$=r_0-1$，否则令r_1：$=(r-r_0)/\alpha$，输出(r_1,r_0)。

使用$Decompose_q(r,\alpha)$算法，能根据由R_q中的多项式组成的向量r和小范数向量s，在不保存s的情况下恢复$r+s$的高位比特，其正确性由下述引理1保证。

引理1[21]. $\boldsymbol{r},\boldsymbol{s}$是由$R_q$中的多项式组成的向量，若满足$\|\boldsymbol{s}\|_\infty\leqslant\beta$且$\|LowBits_q(\boldsymbol{r},\alpha)\|_\infty<\frac{\alpha}{2}-\beta$，则有如下等式成立：

$$HighBits_q(\boldsymbol{r},\alpha)=HighBits_q(\boldsymbol{r}+\mathbf{s},\alpha)。$$

支撑算法3. 提示位生成算法$MakeHint_q(z,r,\alpha)$

该算法输入整数$r,z\in\mathbb{Z}_q$和一个小的正整数α，能生成提示位h，其中h为0或1，用于后续判断r和$r+z$的高位比特是否相等。按照下述3个步骤依次运行。

1. 计算r的高位比特r_1：$=HighBits_q(r,\alpha)$；
2. 计算$r+z$的高位比特v_1：$=HighBits_q(r+z,\alpha)$；
3. 如果$r_1\neq v_1$则h：$=1$，否则h：$=0$，输出h。

支撑算法4. 提示位使用算法$UseHint_q(h,r,\alpha)$

该算法输入整数$r\in\mathbb{Z}_q$、提示位h和一个小的正整数α，能恢复$r+z$的高位比特。按照下述3个步骤依次运行，其正确性由引理2保证。

1. 计算m：$=(q-1)/\alpha$；
2. 计算(r_1,r_0)：$=Decompose_q(r,a)$；
3. 如果$h=1$且$r_0>0$则令v_1：$=(r_1+1)\bmod^+ m$，如果$h=1$且$r_0\leqslant 0$则令v_1：$=(r_1-1)\bmod^+ m$，否则令v_1：$=r_1$，输出v_1。

引理2[21]. 假设q和α是正整数，满足$q>2\alpha$，$q\equiv 1(\bmod^+\alpha)$并且α是偶数。设$\boldsymbol{r}$和$\boldsymbol{z}$是由R_q中的多项式组成的向量，$\boldsymbol{h}$和$\boldsymbol{h}'$为提示向量，其中$\|\boldsymbol{z}\|_\infty\leqslant\alpha/2$，则有如下等式成立：

(1) $UseHint_q(\mathrm{MakeHint}_q(\boldsymbol{z},\boldsymbol{r},\alpha),\boldsymbol{r},\alpha)=HighBits_q(\boldsymbol{r}+\boldsymbol{z},\alpha)$；

(2) 设$\boldsymbol{v}_1=UseHint_q(\boldsymbol{h},\boldsymbol{r},\alpha)$，则$\|\boldsymbol{r}-\boldsymbol{v}_1\cdot\alpha\|_\infty\leqslant\alpha+1$。此外，如果$\boldsymbol{h}$中1的个数为$\omega$，则除$\boldsymbol{r}-\boldsymbol{v}_1\cdot\alpha$的$\omega$个系数外，其余系数在进行$mod\,q$中心约减后，绝对值都不大于$\alpha/2$；

(3) 对于任意的$\boldsymbol{h},\boldsymbol{h}'$，若满足$UseHint_q(\boldsymbol{h},\boldsymbol{r},\alpha)=UseHint_q(\boldsymbol{h}',\boldsymbol{r},\alpha)$，则有$\boldsymbol{h}=\boldsymbol{h}'$。

Dilithium签名方案包括三个算法：密钥生成算法、签名算法和验证算法，这里介绍考虑公钥压缩的版本。所涉及的符号以及含义如表1所示。

表1 参数/符号及含义

符号	含义
$ExpandA$	均匀随机选取$\boldsymbol{A}\leftarrow_{\$}R_q^{k\times l}$
$ExpandS$	均匀随机选取$s_1\leftarrow_{\$}S_\eta^l, s_2\leftarrow_{\$}S_\eta^k$
$ExpandMask$	均匀随机选取$\boldsymbol{y}\leftarrow_{\$}\tilde{S}_{\gamma_1}^l$
$SampleInBall$	均匀随机选取$c\leftarrow_{\$}B_\tau$
H	Hash函数
$\oplus$	异或运算(XOR)
$\|\|$	级联运算(将两个字符串拼接)

密钥生成算法：该算法产生签名需要用到的公钥和私钥，主要思想为首先使用随机种子ζ拓展得到(ρ,ρ',K)，再利用其中的ρ拓展生成一个$k\times l$维的矩阵$\boldsymbol{A}$，这里$\boldsymbol{A}$中的元素都是多项式环R_q中多项式的NTT域表示。利用ρ'对随机私钥向量$\boldsymbol{s}_1,\boldsymbol{s}_2$进行拓展，这些私钥向量的每个元素都在多项式环$R_q$中，且多项式系数的绝对值大小不超过$\eta$。计算$\boldsymbol{t}$：$=\boldsymbol{A}\boldsymbol{s}_1+\boldsymbol{s}_2$，并利用二次幂分解算法$Power2Round_q(\boldsymbol{t},d)$进行分解，得到公钥$pk=(\rho,\boldsymbol{t}_1)$，私钥$sk=(\rho,K,tr,\boldsymbol{s}_1,\boldsymbol{s}_2,\boldsymbol{t}_0)$。

算法1. Dilithium-密钥生成算法

输入：随机种子$\zeta\in\{0,1\}^{256}$；

输出：公钥$pk=(\rho,\boldsymbol{t}_1)$，

私钥$sk=(\rho,K,tr,\boldsymbol{s}_1,\boldsymbol{s}_2,\boldsymbol{t}_0)$.

1. $(\rho,\rho',K)\in\{0,1\}^{256}\times\{0,1\}^{512}\times\{0,1\}^{256}$：$=H(\zeta)$；
2. $\boldsymbol{A}\in R_q^{k\times l}$：$=ExpandA(\rho)$；
3. $(\boldsymbol{s}_1,\boldsymbol{s}_2)\in S_\eta^l\times S_\eta^k$：$=ExpandS(\rho')$；
4. $\boldsymbol{t}$：$=\boldsymbol{A}\boldsymbol{s}_1+\boldsymbol{s}_2$；
5. $(\boldsymbol{t}_1,\boldsymbol{t}_0)$：$=Power2Round_q(\boldsymbol{t},d)$；
6. $\mathrm{tr}\in\{0,1\}^{256}$：$=H(\rho\|\boldsymbol{t}_1)$；
7. RETURN$(pk=(\rho,\boldsymbol{t}_1),sk=(\rho,K,tr,\boldsymbol{s}_1,\boldsymbol{s}_2,\boldsymbol{t}_0))$

签名算法：该算法的对消息M进行签名，主要

思想为首先均匀随机选取$\boldsymbol{y}\leftarrow_{\$}\tilde{S}_{\gamma_1}^{l}$并计算$\boldsymbol{w}:=\boldsymbol{A}\boldsymbol{y}$，使用高低位比特分解算法$Decompose_q(r,\alpha)$得到$\boldsymbol{w}$的高位比特$\boldsymbol{w}_1$和低位比特$\boldsymbol{w}_0$。使用Hash函数$H$和球内采样函数$SampleInBall(\cdot)$计算挑战值$c\in B_\tau$，从而得到$\boldsymbol{z}:=\boldsymbol{y}+c\boldsymbol{s}_1$。通过前面两个拒绝抽样后，使用提示位生成算法$MakeHint_q(\cdot)$计算提示向量$\boldsymbol{h}$。最后，在通过后面两个拒绝抽样后输出合法的签名$\sigma=(\tilde{c},\boldsymbol{z},\boldsymbol{h})$，其中$\beta$满足$\|c\boldsymbol{s}_1\|_\infty\leqslant\beta$，$\|c\boldsymbol{s}_2\|_\infty\leqslant\beta$。

算法2. Dilithium-签名算法

输入：私钥$sk=(\rho,K,tr,\boldsymbol{s}_1,\boldsymbol{s}_2,\boldsymbol{t}_0)$，消息$M$；

输出：签名$\sigma=(\tilde{c},\boldsymbol{z},\boldsymbol{h})$.

1. $\boldsymbol{A}\in R_q^{k\times l}:=\mathrm{ExpandA}(\rho)$；
2. $\mu\in\{0,1\}^{512}:=H(\mathrm{tr}\|M)$；
3. $\kappa:=0$，$(\boldsymbol{z},\boldsymbol{h}):=\perp$；
4. $\rho'\in\{0,1\}^{512}:=H(K\|\mu)$；
   /* 确定性版本，第三轮征集时新增随机性版本，即随机选择$\rho'\leftarrow_{\$}\{0,1\}^{512}$ */
5. WHILE $(\boldsymbol{z},\boldsymbol{h})=\perp$ DO
6. $\quad\boldsymbol{y}\in\tilde{S}_{\gamma_1}^{l}:=\mathrm{ExpandMask}(\rho',\kappa)$；
7. $\quad\boldsymbol{w}:=\boldsymbol{A}\boldsymbol{y}$；
8. $\quad\boldsymbol{w}_1:=HighBits_q(\boldsymbol{w},2\gamma_2)$；
9. $\quad\tilde{c}\in\{0,1\}^{256}:=H(\mu\|\boldsymbol{w}_1)$；
10. $\quad c\in B_\tau:=SampleInBall(\tilde{c})$；
11. $\quad\boldsymbol{z}:=\boldsymbol{y}+c\boldsymbol{s}_1$；
12. $\quad\boldsymbol{r}_0:=LowBits_q(\boldsymbol{w}-c\boldsymbol{s}_2,2\gamma_2)$；
13. $\quad$IF $\|\boldsymbol{z}\|_\infty\geqslant\gamma_1-\beta$ or $\|\boldsymbol{r}_0\|_\infty\geqslant\gamma_2-\beta$
14. $\quad\quad(\boldsymbol{z},\boldsymbol{h}):=\perp$；
15. $\quad$ELSE
16. $\quad\quad\boldsymbol{h}:=MakeHint_q(-c\boldsymbol{t}_0,\boldsymbol{w}-c\boldsymbol{s}_2+c\boldsymbol{t}_0,2\gamma_2)$；
17. $\quad\quad$IF $\|c\boldsymbol{t}_0\|_\infty\geqslant\gamma_2$ or $\|\boldsymbol{h}\|_1>\omega$
18. $\quad\quad\quad(\boldsymbol{z},\boldsymbol{h}):=\perp$；
19. $\quad\quad$END IF
20. $\quad$END IF
21. $\kappa:=\kappa+l$
22. END WHILE
23. RETURN $\sigma=(\tilde{c},\boldsymbol{z},\boldsymbol{h})$

验证算法：该算法验证消息签名对的合法性，主要思想为，在收到消息M以及签名$\sigma=(\tilde{c},\boldsymbol{z},\boldsymbol{h})$后，首先利用ExpandA$(\cdot)$算法获得矩阵$\boldsymbol{A}$，然后计算$\mu$，挑战值$c$以及$\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t}$的高位比特。最后根据相应的条件判断签名合法性。

算法3. Dilithium-验证算法

输入：公钥$pk=(\rho,\boldsymbol{t}_1)$，消息$M$，签名$\sigma=(\tilde{c},\boldsymbol{z},\boldsymbol{h})$；

输出：签名验证通过输出1，否则输出0.

1. $\boldsymbol{A}\in R_q^{k\times l}:=ExpandA(\rho)$；
2. $\mu\in\{0,1\}^{512}:=H(H(\rho\|\boldsymbol{t}_1)\|M)$；
3. $c:=SamplelnBall(\tilde{c})$；
4. $\boldsymbol{w}_1':=UseHint_q(\boldsymbol{h},\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t}_1\cdot 2^d,2\gamma_2)$；
5. IF $\|\boldsymbol{z}\|_\infty<\gamma_1-\beta$ and $\tilde{c}=H(\mu\|\boldsymbol{w}_1')$ and $\|\boldsymbol{h}\|_1\leqslant\omega$
6. $\quad$RETURN 1
7. ELSE
8. $\quad$RETURN 0
9. END IF

更详细的Dilithium签名方案的正确性以及安全性分析请参见其算法设计文档[21]，不同安全等级的Dilithium方案参数选择如表2所示。

表2　Dilithium方案推荐参数

参数	NIST安全等级		
	2	3	5
模数q	8 380 417	8 380 417	8 380 417
t抛弃比特数d	13	13	13
c中±1的个数τ	39	49	60
挑战值空间的熵$\log\binom{256}{\tau}+\tau$	192	225	257
y的系数范围γ_1	2^{17}	2^{19}	2^{19}
低位舍入范围γ_2	$(q-1)/88$	$(q-1)/32$	$(q-1)/32$
矩阵$\boldsymbol{A}$的维数(k,l)	(4,4)	(6,5)	(8,7)
私钥的范围η	2	4	2
$\beta=\tau\times\eta$	78	196	120
$\boldsymbol{h}$中1的个数最大值ω	80	55	75
重复次数	4.25	5.1	3.85

4　Olithium：本文在线/离线签名方案

本节详细说明本文提出的基于格的无陷门在线/离线签名方案，为了方便讨论，将其命名为Olithium。所涉及的符号以及含义和Dilithium一致。根据2.3节的定义，在线/离线签名方案包括四个算法：密钥生成算法、离线签名算法、在线签名算法和验证算法。为了便于理解，在描述具体方案和算法之前，先对相关思想进行概览。

4.1 设计思想

在深入分析Dilithium方案的各个运算环节后，发现签名过程中较为耗时的多项式矩阵乘法操作$\boldsymbol{w}:=\boldsymbol{A}\boldsymbol{y}$需要在收到待签名消息$M$后才能进行(在线阶段)。其原因在于，随机多项式向量$\boldsymbol{y}$的生成依赖于消息M，一旦触发格签名的拒绝抽样(中止)，则需要重启整个过程，导致签名时间明显增加。此外，将Dilithium方案修改为更适合于在线/离线签名的设置，另一个值得考虑的问题是，如何减少在在线签名过程中调用效率低下的高低位比特分解算法$Decompose_q(r,\alpha)$的次数。

为了解决第一个问题，本文在设计Olithium方案的过程中，基于Γ-协议[16]的思想，对底层的零知识协议进行了改进。在不影响参数设置和安全性的前提下，使随机多项式向量$\boldsymbol{y}$的生成不再依赖于消息M。该改进使得原本需在在线阶段执行的耗时操作(如多项式矩阵乘法$\boldsymbol{w}:=\boldsymbol{A}\boldsymbol{y}$)以及相关计算被移至离线阶段进行预计算，从而显著减少了在线签名阶段的计算量，提升了签名效率。

为了解决第二个问题，Olithium方案采用了一种比Dilithium方案更适合在线/离线设置的方法，用于计算提示向量$\boldsymbol{h}$和$\boldsymbol{w}-c\boldsymbol{s}_2$的低比特位$\boldsymbol{r}_0$。通过这一改进设计，将签名过程中每次拒绝抽样循环中调用高低位比特分解算法$Decompose_q(r,\alpha)$的次数从3次降到1次。在保证输出一致的前提下，此优化显著缩短了产生有效签名所需要的(在线)签名时间。该优化在工程实现上提升了效率，同时保持原有的参数选择，且未影响底层问题的计算困难性与方案安全性。

4.2 Olithium方案描述

密钥生成算法：该算法产生签名需要用到的公钥私钥，和Dilithium方案的密钥生成算法一致。

算法4. Olithium-密钥生成算法

输入：随机种子$\zeta\in\{0,1\}^{256}$；

输出：公钥$pk=(\rho,\boldsymbol{t}_1)$，

私钥$sk=(\rho,K,tr,\boldsymbol{s}_1,\boldsymbol{s}_2,\boldsymbol{t}_0)$.

1. $(\rho,\rho',K)\in\{0,1\}^{256}\times\{0,1\}^{512}\times\{0,1\}^{256}:=H(\zeta)$；
2. $\boldsymbol{A}\in R_q^{k\times l}:=ExpandA(\rho)$；
3. $(\boldsymbol{s}_1,\boldsymbol{s}_2)\in S_\eta^l\times S_\eta^k:=ExpandS(\rho')$；
4. $\boldsymbol{t}:=\boldsymbol{A}\boldsymbol{s}_1+\boldsymbol{s}_2$；
5. $(\boldsymbol{t}_1,\boldsymbol{t}_0):=Power2Round_q(\boldsymbol{t},d)$；
6. $tr\in\{0,1\}^{256}:=H(\rho\|\boldsymbol{t}_1)$；
7. RETURN$\big(pk=(\rho,\boldsymbol{t}_1),sk=(\rho,K,tr,\boldsymbol{s}_1,\boldsymbol{s}_2,\boldsymbol{t}_0)\big)$

离线签名算法：签名设备在未收到消息时的离线阶段运行该算法，预计算并存储中间值φ。

算法5. Olithium-离线签名算法

输入：私钥$sk=(\rho,K,tr,\boldsymbol{s}_1,\boldsymbol{s}_2,\boldsymbol{t}_0)$；

输出：中间值数组ξ.

1. $\boldsymbol{A}\in R_q^{k\times l}:=ExpandA(\rho)$；
2. $\mu\in\{0,1\}^{512}:=H(tr)$；
3. $\kappa:=0$；
4. $\xi:=\{\}$；/* 初始时为空 */
5. $\rho'\in\{0,1\}^{512}:=H(K\|\mu)$；
6. WHILE the number of φ is insufficient DO
7. $\quad\boldsymbol{y}\in\tilde{S}_{\gamma_1}^l:=\mathrm{ExpandMask}(\rho',\kappa)$；
8. $\quad\boldsymbol{w}:=\boldsymbol{A}\boldsymbol{y}$；
9. $\quad(\boldsymbol{w}_1,\boldsymbol{w}_0):=Decompose_q(\boldsymbol{w},2\gamma_2)$；
10. $\quad\tilde{c}\in\{0,1\}^{256}:=H(\mu\|\boldsymbol{w}_1)$；
11. $\quad\kappa:=\kappa+l$；
12. $\quad\varphi:=(\tilde{c},\boldsymbol{y},\boldsymbol{w}_0,\boldsymbol{w}_1)$；
13. $\quad\xi:=\xi\cup\{\varphi\}$；
14. END WHILE
15. RETURN ξ

在线签名算法：该算法由签名设备在收到消息后执行，用来产生有效的签名。具体而言，收到消息M后，签名者提取预先存储的中间值$\varphi=(\tilde{c},\boldsymbol{y},\boldsymbol{w}_0,\boldsymbol{w}_1)$完成相关计算，然后输出签名。如果签名触发了拒绝抽样(中止)，签名者只需要在预先存储的$\xi=\{\varphi_1,\varphi_2,\cdots,\varphi_N\}$中提取一组不同的中间值，并重新启动签名算法的在线阶段。已经被提取使用过的中间值φ在之后的签名过程不再使用。

算法6. Olithium-提示位生成算法$MakeHint_q$

输入：整数$r,z\in\mathbb{Z}_q$和一个小的正整数α；

输出：r和$r+z$的高位比特不相等则输出1，相等则输出0.

1. $z:=z\ mod^+q$
2. IF $z\leqslant a$ or $z>q-a$ or ($z=q-a$ and $r=0$)
3. $\quad$RETURN 0；
4. END IF
5. RETURN 1

算法7. Olithium-在线签名算法

输入：中间值$\varphi=(\tilde{c}, \boldsymbol{y}, \boldsymbol{w}_0, \boldsymbol{w}_1)$，私钥$sk=(\rho, K, tr, \boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{t}_0)$，消息$M$；

输出：签名$\sigma=(\tilde{c}, \boldsymbol{z}, \boldsymbol{h})$.

1. $\tilde{k}\in\{0,1\}^{256}:=H(M)$
2. WHILE$(\boldsymbol{z}, \boldsymbol{h})=\perp$ DO
3. $\quad\tilde{c}:=\tilde{k}\oplus\tilde{c}$
4. $\quad c\in B_\tau:=$SamplelnBall$(\tilde{c})$;
5. $\quad\boldsymbol{z}:=\boldsymbol{y}+c\boldsymbol{s}_1$;
6. $\quad\boldsymbol{r}_0:=\boldsymbol{w}_0-c\boldsymbol{s}_2$;
7. $\quad$IF $\|\boldsymbol{z}\|_\infty\geqslant\gamma_1-\beta$ or $\|\boldsymbol{r}_0\|_\infty\geqslant\gamma_2-\beta$
8. $\quad\quad(\boldsymbol{z}, \boldsymbol{h}):=\perp$;
9. $\quad$ELSE
10. $\quad\boldsymbol{h}:=MakeHint_q(\boldsymbol{r}_0+c\boldsymbol{t}_0, \boldsymbol{w}_1, \gamma_2)$;
11. $\quad\quad$IF $\|c\boldsymbol{t}_0\|_\infty\geqslant\gamma_2$ or $\|\boldsymbol{h}\|_1>\omega$
12. $\quad\quad\quad(\boldsymbol{z}, \boldsymbol{h}):=\perp$;
13. $\quad\quad$END IF
14. $\quad$END IF
15. END WHILE
16. RETURN $\sigma=(\tilde{c}, \boldsymbol{z}, \boldsymbol{h})$

验证算法：该算法由验证者执行，验证消息签名对的合法性。

算法8. Olithium-验证算法

输入：公钥$pk=(\rho, \boldsymbol{t}_1)$，消息$M$，签名$\sigma=(\tilde{c}, \boldsymbol{z}, \boldsymbol{h})$；

输出：验证成功输出1，验证失败输出0.

1. $\boldsymbol{A}\in R_q^{k\times l}:=$ExpandA$(\rho)$;
2. $\mu\in\{0,1\}^{512}:=H(H(\rho\|\boldsymbol{t}_1))$;
3. $\tilde{k}\in\{0,1\}^{256}:=H(M)$;
4. $c:=$SamplelnBall$(\tilde{c})$;
5. $\boldsymbol{w}_1':=UseHint_q(\boldsymbol{h}, \boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t}_1\cdot 2^d, 2\gamma_2)$;
6. IF $\|\boldsymbol{z}\|_\infty<\gamma_1-\beta$ AND $\tilde{c}=H(\mu\|\boldsymbol{w}_1')\oplus\tilde{k}$ AND $\|\boldsymbol{h}\|_1\leqslant\omega$
7. $\quad$RETURN 1
8. ELSE
9. $\quad$RETURN 0
10 END IF

4.3　Olithium方案的正确性分析

由于Olithium方案采用了Dilithium相同的方式计算提示向量$\boldsymbol{h}$和$\boldsymbol{w}-c\boldsymbol{s}_2$的低位比特$\boldsymbol{r}_0$，因此其正确性和Dilithium一样以$Decompose_q$函数的性质为基础。本文通过简单回顾相关引理，以及Dilithium方案的正确性（更多细节请读者参考其算法文档[21]），并说明Olithium方案在设计过程中所进行的修改，来证明Olithium方案的正确性。

引理3[21]. $\boldsymbol{r}, \boldsymbol{s}$是由$R_q$中多项式组成的向量，

$$(\boldsymbol{w}_1, \boldsymbol{w}_0)=Decompose_q(\boldsymbol{r}-\boldsymbol{s}, \alpha),$$
$$(\boldsymbol{r}_1, \boldsymbol{r}_0)=Decompose_q(\boldsymbol{r}, \alpha),$$
$$\|\boldsymbol{s}\|_\infty\leqslant\beta,$$

则有如下等价关系成立

$$\|\boldsymbol{w}_0+\boldsymbol{s}\|_\infty<\frac{\alpha}{2}-\beta\Leftrightarrow\boldsymbol{w}_1=\boldsymbol{r}_1\wedge\|\boldsymbol{r}_0\|_\infty<\alpha/2-\beta。$$

根据引理3，计算$(\boldsymbol{r}_1, \boldsymbol{r}_0)=Decompose_q(\boldsymbol{w}-c\boldsymbol{s}_2, 2\gamma_2)$，并检查$\|\boldsymbol{r}_0\|_\infty<\gamma_2-\beta$和$\boldsymbol{r}_1=\boldsymbol{w}_1$，实际上等价于检查$\|\boldsymbol{w}_0-c\boldsymbol{s}_2\|_\infty<\gamma_2-\beta$，其中$\boldsymbol{w}_0$是$\boldsymbol{w}$的低位比特，满足$(\boldsymbol{w}_1, \boldsymbol{w}_0)=Decompose_q(\boldsymbol{w}, 2\gamma_2)$和$\|c\boldsymbol{s}_2\|_\infty\leqslant\beta$。若该检查能通过，则意味着$\boldsymbol{w}_0-c\boldsymbol{s}_2$是$\boldsymbol{w}-c\boldsymbol{s}_2$的低位比特。

在Dilithium方案的提示向量$\boldsymbol{h}$的计算过程中，根据$MakeHint_q$函数的定义，如果$\boldsymbol{w}-c\boldsymbol{s}_2$和$\boldsymbol{w}-c\boldsymbol{s}_2+c\boldsymbol{t}_0$的相应系数的高位比特不同，则其签名算法（对应本文算法2）中的$\boldsymbol{h}$的相应系数是非零的，此时该算法已经在步骤8中计算出$\boldsymbol{w}$的分解为$\boldsymbol{w}=\alpha\boldsymbol{w}_1+\boldsymbol{w}_0$，可知$\alpha\boldsymbol{w}_1+(\boldsymbol{w}_0-c\boldsymbol{s}_2)$是$\boldsymbol{w}-c\boldsymbol{s}_2$的正确分解。

另一方面，$\alpha\boldsymbol{w}_1+(\boldsymbol{w}_0-c\boldsymbol{s}_2+c\boldsymbol{t}_0)$是$\boldsymbol{w}-c\boldsymbol{s}_2+c\boldsymbol{t}_0$的正确分解（即高位比特是$\boldsymbol{w}_1$）当且仅当$\boldsymbol{w}_0-c\boldsymbol{s}_2+c\boldsymbol{t}_0$中每个多项式系数都在区间$(-\gamma_2, \gamma_2]$内，或者当某个多项式系数是$-\gamma_2$时相应的$\boldsymbol{w}_1$系数为0（即考虑$Decompose_q$函数的边界情况）。因此，对于提示向量$\boldsymbol{h}$，只需检查$\boldsymbol{w}_0-c\boldsymbol{s}_2+c\boldsymbol{t}_0$的系数上的相应条件。在验证提示向量$\boldsymbol{h}$后，Dilithium方案的验证算法（对应本文算法3）退化为普通的基于格的无陷门签名方案，其正确性不难验证，限于篇幅，更多细节请参见其文档[21]。

在Olithium方案的改进中，本文也对$MakeHint_q$函数进行了修改（对应本文算法6）并以$\boldsymbol{r}_0+c\boldsymbol{t}_0$, $\boldsymbol{w}_1$, γ_2作为输入，其中$\boldsymbol{r}_0=\boldsymbol{w}_0-c\boldsymbol{s}_2$。具体而言，如果$\boldsymbol{r}_0+c\boldsymbol{t}_0$中多项式每个系数都在区间$(-\gamma_2, \gamma_2]$内，或者当多项式系数是$-\gamma_2$时相应的$\boldsymbol{w}_1$系数为0（即考虑$Decompose_q$函数的边界情况），则$\boldsymbol{w}-c\boldsymbol{s}_2$和$\boldsymbol{w}-c\boldsymbol{s}_2+c\boldsymbol{t}_0$相应系数的高位比特相同，反之则不相同。

在Olithium方案的验证算法（对应本文算法8）中，在计算$\boldsymbol{w}_1'$之后，通过验证下述三个条件则来检查签名的合法性：

$$\|\boldsymbol{z}\|_{\infty}<\gamma_1-\beta,$$
$$H(\mu\|\boldsymbol{w}_1')\oplus\tilde{k}=\tilde{c},$$
$$\|\boldsymbol{h}\|_1\leqslant\omega。$$

事实上，只需要证明$\boldsymbol{w}_1'=\boldsymbol{w}_1$。根据引理2的(1)可知，当$\|c\boldsymbol{t}_0\|_{\infty}\leqslant\gamma_2$时，有

$$UseHint_q(\boldsymbol{h},\boldsymbol{w}-c\boldsymbol{s}_2+c\boldsymbol{t}_0,2\gamma_2)=HighBits_q(\boldsymbol{w}-c\boldsymbol{s}_2,2\gamma_2),$$

结合参数之间满足的关系$\boldsymbol{w}=\boldsymbol{A}\boldsymbol{y}$，$\boldsymbol{z}=\boldsymbol{y}+c\boldsymbol{s}_1$以及$\boldsymbol{t}=\boldsymbol{A}\boldsymbol{s}_1+\boldsymbol{s}_2$，可得

$$\boldsymbol{w}-c\boldsymbol{s}_2=\boldsymbol{A}\boldsymbol{y}-c\boldsymbol{s}_2=\boldsymbol{A}(\boldsymbol{z}-c\boldsymbol{s}_1)-c\boldsymbol{s}_2=\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t}。$$

另一方面，根据$\boldsymbol{t}=\boldsymbol{t}_1\cdot2^d+\boldsymbol{t}_0$可得

$$\boldsymbol{w}-c\boldsymbol{s}_2+c\boldsymbol{t}_0=\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t}_1\cdot2^d。$$

因此验证者可以计算得到

$$\boldsymbol{w}_1'=UseHint_q(\boldsymbol{h},\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t}_1\cdot2^d,2\gamma_2)=UseHint_q(\boldsymbol{h},\boldsymbol{w}-c\boldsymbol{s}_2+c\boldsymbol{t}_0,2\gamma_2)=HighBits_q(\boldsymbol{w}-c\boldsymbol{s}_2,2\gamma_2)。$$

此外，由于$\|c\boldsymbol{s}_2\|_{\infty}\leqslant\beta$，$LowBits_q(\boldsymbol{w}-c\boldsymbol{s}_2,2\gamma_2)<\gamma_2-\beta$，以及引理1，有

$$HighBits_q(\boldsymbol{w}-c\boldsymbol{s}_2,2\gamma_2)=HighBits_q(\boldsymbol{w},2\gamma_2)=\boldsymbol{w}_1。$$

综上所述，验证者在验证算法(对应本文算法8)中计算Hash函数所输入的$\boldsymbol{w}_1'$，与签名者在离线签名算法(对应本文算法5)中计算Hash函数所输入的$\boldsymbol{w}_1$相同。根据Fiat-Shamir变换的过程，可知Olithium方案的正确性成立。

4.4 Olithium方案的安全性分析

本节证明，在量子随机谕言模型下，Olithium方案满足SUF-CMA(定义4)，其基于$\mathrm{MLWE}_{q,m,k,D}$和$\mathrm{MSIS}_{q,m,k,\gamma}$问题。由于Olithium方案可以视为将Dilithium方案先改为在线/离线设置，再使用额外的技巧进行工程实现上的优化，其安全性应在本质上等价，证明思想类似。限于篇幅，本文省略一些相同的证明部分，请感兴趣的读者参考Dilithium方案的算法设计文档[21]。

根据Olithium方案的密钥生成算法(对应本文算法4)以及$\mathrm{MLWE}_{q,m,k,D}$假设，公钥$(\boldsymbol{A},\boldsymbol{t}=\boldsymbol{A}\boldsymbol{s}_1+\boldsymbol{s}_2)$与均匀随机选取的$(\boldsymbol{A},\boldsymbol{t})$是计算不可区分的。因此要证明Olithium方案的安全性，只需要分析下述实验：

敌手$\mathcal{A}$收到均匀随机选取的$(\boldsymbol{A},\boldsymbol{t})$，目标是伪造一个合法的消息/签名对$\{M,(\boldsymbol{z},\boldsymbol{h},c)\}$满足：

(1) $\|\boldsymbol{z}\|_{\infty}<\gamma_1-\beta$；

(2) $H(\mu\|\boldsymbol{w}_1,M)=c$；

(3) $\|\boldsymbol{h}\|_1\leqslant\omega$。

其中，$\boldsymbol{w}_1=UseHint_q(\boldsymbol{h},\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t}_1\cdot2^d,2\gamma_2)$。

根据引理2的2)，有

$$\|\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t}_1\cdot2^d-UseHint_q(\boldsymbol{h},\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t}_1\cdot2^d,2\gamma_2)\cdot2\gamma_2\|_{\infty}\leqslant2\gamma_2+1,$$

令$\boldsymbol{w}_1=(\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t}_1\cdot2^d+\boldsymbol{u})/2\gamma_2$，其中$\|\boldsymbol{u}\|_{\infty}\leqslant2\gamma_2+1$。此外，$\boldsymbol{u}$至多有$\omega$个系数的绝对值大于$\gamma_2$。因为$\boldsymbol{t}=\boldsymbol{t}_1\cdot2^d+\boldsymbol{t}_0$，其中$\|\boldsymbol{t}_0\|_{\infty}\leqslant2^{d-1}$，有

$$\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t}_1\cdot2^d+\boldsymbol{u}=\boldsymbol{A}\boldsymbol{z}-c(\boldsymbol{t}-\boldsymbol{t}_0)+\boldsymbol{u}=\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t}+(c\boldsymbol{t}_0+\boldsymbol{u})=\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t}+\boldsymbol{u}',$$

$$\boldsymbol{w}_1=(\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t}+\boldsymbol{u}')/2\gamma_2=\frac{1}{2\gamma_2}\cdot\left[\boldsymbol{A}|\boldsymbol{t}|\boldsymbol{I}_k\right]\cdot\begin{bmatrix}\boldsymbol{z}\\c\\\boldsymbol{u}'\end{bmatrix}。$$

注意到

$$\|\boldsymbol{u}'\|_{\infty}\leqslant\|c\boldsymbol{t}_0\|_{\infty}+\|\boldsymbol{u}\|_{\infty}\leqslant\|c\|_1\cdot\|\boldsymbol{t}_0\|_{\infty}+\|\boldsymbol{u}\|_{\infty}\leqslant\tau\cdot2^{d-1}+2\gamma_2+1,$$

因此，为了伪造一个新的消息签名对，敌手需要找到M和$(\boldsymbol{z},c,\boldsymbol{u}')$，满足

$$H\left(\mu\left\|\frac{1}{2\gamma_2}\cdot\left[\boldsymbol{A}|\boldsymbol{t}|\boldsymbol{I}_k\right]\cdot\begin{bmatrix}\boldsymbol{z}\\c\\\boldsymbol{u}'\end{bmatrix},M\right.\right)=c,$$

其中，$M\in\{0,1\}^*$，且$\|\boldsymbol{z}\|_{\infty}<\gamma_1-\beta$，$\|c\|_{\infty}=1$，$\|\boldsymbol{u}'\|_{\infty}\leqslant\tau\cdot2^{d-1}+2\gamma_2+1$，$\boldsymbol{u}'$至多有$\omega$个系数的绝对值大于$\tau\cdot2^{d-1}+\gamma_2$。为了方便描述，定义函数$H'$使得$H(\mu\|x,M)=H'(\mu\|2\gamma_2x,M)$，上述等式变为

$$H'\left(\mu\left\|\left[\boldsymbol{A}|\boldsymbol{t}|\boldsymbol{I}_k\right]\cdot\begin{bmatrix}\boldsymbol{z}\\c\\\boldsymbol{u}'\end{bmatrix},M\right.\right)=c。$$

根据Hash函数结构与输出的独立性[21]，对于固定的M，该问题仍然困难。此时等式变为

$$H''\left(\mu\left\|\left[\boldsymbol{A}|\boldsymbol{t}|\boldsymbol{I}_k\right]\cdot\begin{bmatrix}\boldsymbol{z}\\c\\\boldsymbol{u}'\end{bmatrix}\right.\right)=c。$$

由于$(\boldsymbol{A},\boldsymbol{t})$是均匀随机选取的，且$\mu$只依赖于公钥$(\rho,\boldsymbol{t}_1)$，因此和伪造Dilithium方案的合法签名一样，需要选择合适的$\boldsymbol{w}$，使得$H''(\mu\|\boldsymbol{w})=c$，并找到$(\boldsymbol{z},\boldsymbol{u}')$使得$\boldsymbol{A}\boldsymbol{z}+\boldsymbol{u}'=\boldsymbol{w}+c$，满足$\|\boldsymbol{z}\|_{\infty}<\gamma_1-\beta\ll\tau\cdot2^{d-1}+2\gamma_2+1$并且$\boldsymbol{u}'$至多有$\omega$个系数的绝对值可以大于$\tau\cdot2^{d-1}+\gamma_2$。根据SelfTargetMSIS假设可知伪造满足$\boldsymbol{A}\boldsymbol{z}+\boldsymbol{u}'=\boldsymbol{t}'$的解$(\boldsymbol{z},\boldsymbol{u}')$是困难的(该困难问题的具体描述以及参数与Dilithium算法

文档[21]相同）。因此，Olithium方案在量子随机谕言模型下具备选择消息攻击下的强存在性不可伪造性。

证毕.

5　性能分析与对比

5.1　理论分析

5.1.1　重复次数

影响基于格的无陷门签名的效率的一个重要指标是拒绝抽样过程中的重复，本节计算Olithium方案不同安全等级重复次数的期望值。

由于参数的设置使得在线签名阶段（对应本文算法7）中的步骤10导致$(\boldsymbol{z},\boldsymbol{h}):=\perp$的概率在1%到2%之间。因此，循环重启主要是由在线签名阶段中的步骤6引起，只需计算步骤6中$(\boldsymbol{z},\boldsymbol{h})$被设置为$\perp$的概率。

在步骤6中，当$\|\boldsymbol{z}\|_\infty<\gamma_1-\beta$时，对于$c\boldsymbol{s}_1$的每个系数$\sigma$，都有$\boldsymbol{y}_i$的相应系数应该落在区间$[-\gamma_1+\beta+1-\sigma,\gamma_1-\beta-1-\sigma]$。该范围的大小为$2(\gamma_1-\beta)-1$。另一方面，$\boldsymbol{y}$中每个多项式的系数都是从$(-\gamma_1,\gamma_1)$中均匀随机选取，共$2\gamma_1-1$个值。因此，$\|\boldsymbol{z}\|_\infty<\gamma_1-\beta$的概率为

$$\left(\frac{2(\gamma_1-\beta)-1}{2\gamma_1-1}\right)^{256\cdot l}=\left(1-\frac{\beta}{\gamma_1-1/2}\right)^{256\cdot l}\approx e^{-256\cdot\beta l/\gamma_1}$$

该近似关系用到$\gamma_1\gg 1/2$。

其次，还需要计算下列不等式成立的概率

$$\|\boldsymbol{r}_0\|_\infty=\|LowBits_q(\boldsymbol{w}-c\boldsymbol{s}_2,2\gamma_2)\|_\infty<\gamma_2-\beta$$

假设$\boldsymbol{r}_0$中多项式的每个系数都服从在模$2\gamma_2$下的均匀分布，那么$\|\boldsymbol{r}_0\|_\infty<\gamma_2-\beta$的概率为

$$\left(\frac{2(\gamma_2-\beta)-1}{2\gamma_2}\right)^{256\cdot k}\approx e^{-256\cdot\beta k/\gamma_2},$$

该近似关系用到$\beta\gg 1/2$。

综上，在线签名阶段步骤6通过的概率约为

$$e^{-256\cdot\beta\left(\frac{l}{\gamma_1}+\frac{k}{\gamma_2}\right)},$$

代入具体数值的结果如表3和表4所示。

表3　Olithium方案的重复次数期望值

安全等级	NIST安全等级		
	2	3	5
重复次数的期望值	4.25	5.1	3.85

表4　Olithium方案的在线签名成功概率

存储中间值组数N	1	2	3	4	5	6	7	8	9
Olithium-2在线签名成功概率(%)	23.50	41.48	55.23	65.75	73.80	79.96	84.67	88.27	91.03
Olithium-3在线签名成功概率(%)	19.63	35.41	48.09	58.28	66.47	73.05	78.34	82.59	86.01
Olithium-5在线签名成功概率(%)	25.96	45.18	59.41	69.95	77.75	83.52	87.80	90.97	93.31

5.1.2　密钥和签名尺寸

在相同安全等级下，Olithium在线/离线签名方案与Dilithium数字签名方案使用同样的安全参数，密钥和签名尺寸也保持一致。根据其文档中的计算方法，公钥大小为$32+320k$字节，私钥大小为$96+32((k+l)\cdot\lceil\log(2\eta+1)\rceil+13k)$字节，签名大小为$32l\cdot(1+\log\gamma_1)+\omega+k+32$字节。将具体数值代入计算公式后，可以得到不同安全等级的密钥和签名尺寸如表5所示。

表5　Olithium方案与Dilithium方案的密钥和签名尺寸

安全等级	NIST安全等级		
	2	3	5
公钥尺寸(Byte)	1312	1952	2592
私钥尺寸(Byte)	2528	4000	4864
签名尺寸(Byte)	2420	3293	4595

5.1.3　存储空间

在计算Olithium方案执行各阶段的临时变量总空间前，先分析其在离线签名阶段存储一组中间值所需空间，即$\varphi=(\tilde{c},\boldsymbol{y},\boldsymbol{w}_0,\boldsymbol{w}_1)$，其中：

(1) $\tilde{c}$需要256/8=32字节。

(2) $\boldsymbol{y}$是多项式向量，多项式的每个系数可以用γ_1-c表示，其中$c\in\{0,1,\cdots,2\gamma_1-1\}$，需要对$c$进行存储。对于NIST安全等级2，将$\boldsymbol{y}$位打包需要$l\cdot256\cdot18/8=576\cdot l$字节；对于NIST安全等级3和5，将$\boldsymbol{y}$位打包需要$l\cdot256\cdot20/8=640\cdot l$字节。

(3) $\boldsymbol{w}_0$是由k个多项式$w_{0,0},w_{0,1},\cdots,w_{0,k-1}$组成的向量，多项式的每个系数可以用$\gamma_2-c$表示，需要对$c$进行存储。对于NIST安全等级2，满足$c\in\{0,1,\cdots,190\,464\}$，可以用18比特存储，将$\boldsymbol{w}_0$位打包需要$k\cdot256\cdot18/8=576\cdot k$字节；对于NIST安

全等级3和5，满足$c\in\{0,1,\cdots,523\,776\}$，可以用19比特存储，将$\boldsymbol{w}_0$位打包需要$k\cdot256\cdot19/8=608\cdot k$字节。

（4）$\boldsymbol{w}_1$是由k个多项式$w_{1,0},w_{1,1},\cdots,w_{1,k-1}$组成的向量，对于NIST安全等级2，多项式的每个系数属于$\{0,1,\cdots,43\}$，可以用6比特存储，将$\boldsymbol{w}_1$位打包需要$k\cdot256\cdot6/8=192\cdot k$字节。对于NIST安全等级3和5，多项式的每个系数属于$\{0,1,\cdots,15\}$，可以用4比特存储，将$\boldsymbol{w}_1$位打包需要$k\cdot256\cdot4/8=128\cdot k$字节。

结合表2可得具体数值（表6），根据此计算临时变量总空间，并与Dilithium方案对比（表7）。本质上，Dilithium方案可以视为仅生成1组中间值的Olithium方案，并直接使用该值进行签名，如果签名失败，则需重新启动整个签名过程，且整个过程都需要在收到消息后才能进行（在线）。尽管Olithium方案的在线签名阶段和离线签名阶段的临时变量加起来占用略大的存储空间，但其签名时间显著缩短。

表6 Olithium方案存储一组中间值所需空间

安全等级	NIST安全等级		
	2	3	5
$\tilde{c}$的存储空间（Byte）	32	32	32
$\boldsymbol{y}$的存储空间（Byte）	2304	3200	4480
$\boldsymbol{w}_0$的存储空间（Byte）	2304	3648	4864
$\boldsymbol{w}_1$的存储空间（Byte）	768	768	1024
总存储空间（Byte）	5408	7648	10 400

表7 Olithium方案与Dilithium方案的临时变量总空间（与表4相同，N表示存储中间值组数）

安全等级	NIST安全等级		
	2	3	5
Olithium/Dilithium密钥生成（KB）	36	58	94
Olithium-在线签名（KB）	5.3N+26.7	7.5N+44.5	10.2N+75.8
Olithium-离线签名（KB）	20	28	38
Dilithium-签名（KB）	52	80	123
Olithium/Dilithium验证（KB）	33	54	88

由表7可知，增加部分处于可接受范围内，特别是对于存储空间较为充足的签名设备。而对于存储空间较为有限的设备，可以选择仅存储少数几组离线阶段中间值，并在签名过程中同时进行离线阶段计算以及（使用之前产生的中间值）在线阶段计算。在这种情况下，Olithium方案在签名过程中的存储空间需求相比Dilithium方案不会增加太多。此外，为了达到理想的性能，还可以根据设备的存储空间限制，以及表4中的在线签名成功概率，权衡决定存储的离线阶段中间值的组数，具体参见6.1节中的讨论。

5.2 实验分析

本节通过仿真实验对Olithium方案进行性能评估。基于Dilithium设计团队提交至NIST第三轮标准征集的源代码[①]，我们采用C语言实现Olithium方案，并公开了源代码[②]。测试平台配置包括处理器为11th Gen Intel®Core™ i7-11700@2.50GHz，内存为32 GB，操作系统为Ubuntu 18.04.6 LTS版本的64位。在对比中，Olithium-i和Dilithium-i（$i=2,3,5$）表示算法的不同安全等级，待签名的消息随机产生。

由于Olithium方案并未改变Dilithium方案的密钥生成算法，实验所测得的两个方案的密钥生成算法的执行时间如表8所示。

表8 Olithium方案与Dilithium方案的密钥生成时间

安全等级	NIST安全等级		
	2	3	5
密钥生成时间（μs）	67	114	170

表9表示在不同NIST安全等级下，Olithium方案在离线阶段生成一组中间值的时间，该过程可以在签名设备未收到待签名消息就进行。

表9 Olithium方案的离线阶段生成一组中间值时间

安全等级	NIST安全等级		
	2	3	5
Olithium生成一组中间值时间（μs）	36	49	75

表10是Olithium方案与Dilithium方案在不同安全等级的在线签名效率对比，实验将（因为格签名的拒绝抽样导致的）不同重复次数的签名过程各进行了十万次，取平均效率作为结果。这里在线签名效率的评估过程，包括先运行足够多次数的离线部分，然后开始在线部分，并统计其时间，即和在线/离线签名的现实使用场景相同。由于Dilithium方案的部分过程也可以离线完成，出于比较的公平性，对比实验中也只统计了其在线部分的时间。从表中可

① CRYSTALS-Dilithium，NIST Submission Package for round 3，https://pq-crystals.org/dilithium/resources.shtml

② Olithium-Code https://github.com/jiamiwen/Olithium

表10　Olithium方案与Dilithium方案的不同重复次数在线签名效率对比

重复次数	1	2	3	4	5	6	7	8	9
Olithium-2在线签名速度(次/s)	27 777	20 833	16 949	13 888	10 638	9433	8196	7407	6172
Dilithium-2在线签名速度(次/s)	15 151	9259	6756	4761	4132	3521	3076	2597	2347
提升(%)	83.33	125.00	150.87	191.70	157.45	167.91	166.45	185.21	162.97
Olithium-3在线签名速度(次/s)	21 276	13 513	11 363	9803	7751	6369	5847	5025	4405
Dilithium-3在线签名速度(次/s)	10 526	6329	4716	3164	2666	2145	1972	1718	1602
提升(%)	102.13	113.51	140.95	209.83	190.74	196.92	196.50	192.49	174.97
Olithium-5在线签名速度(次/s)	15 873	12 195	8333	7246	6329	4651	3968	3236	3021
Dilithium-5在线签名速度(次/s)	7462	4291	3076	2320	1869	1610	1340	1203	1094
提升(%)	112.72	184.20	170.90	212.33	238.63	188.88	196.12	168.99	176.14

以看出，Olithium方案在在线签名效率上有显著的提升，且随着重复次数的增加，提升越来越明显，最后趋于稳定。此外，提升是波动的，经过分析认为其原因可能是Olithium方案是随机性方案，拥有多个中止条件，在线签名会在多个位置随机中止重启，具备不确定性。表10还显示随着安全等级的增加，在线签名效率提升越来越快，经过分析认为其原因可能是随着安全等级的上升，矩阵$\boldsymbol{A}$的维数变大，使本文对涉及该部分进行的优化效果更加明显。

最后，实验还分别对Olithium方案和Dilithium方案的总在线签名时间和验证时间(进行十万次取平均值)进行测试，其中每次签名的消息随机产生，彼此之间无关联，测试结果如表11和表12所示。该实验可以视为在真实场景中进行在线签名(每次签名过程中的重复次数不可预测)和验证签名。从表11和表12中可以看出，相比起Dilithium方案，Olithium方案的在线签名时间缩短约50%，而验证时间几乎相同。

表11　Olithium方案与Dilithium方案的签名时间

安全等级	NIST安全等级		
	2	3	5
Olithium在线签名时间(μs)	123	162	181
Dilithium在线签名时间(μs)	255	365	425

表12　Olithium方案与Dilithium方案的验证时间

安全等级	NIST安全等级		
	2	3	5
Olithium验证时间(μs)	67	106	174
Dilithium验证时间(μs)	64	106	173

6　讨论与应用

6.1　Olithium方案的讨论

本文提出的Olithium在线/离线签名方案不仅在设计上继承了Dilithium数字签名方案的核心思想以保证效率、尺寸和安全性，还通过改进其底层的零知识协议以及优化工程实现，使其更加适配在线/离线的设定，最终有效降低了在线签名阶段的计算负担。值得强调的是，Olithium的优化旨在提升算法性能，而不会对签名的用途、功能或适用场景造成任何影响。因此，在任何能使用Dilithium方案的密码系统中，都可以直接替换为Olithium方案。在一些硬件平台上，仅仅只需要替换部分电路和程序。

在现实场景中使用时，Olithium方案更适合于具有充足存储空间的签名设备，例如数字证书颁发场景。在这类场景中，拥有固定私钥的签名者可以离线预先计算并存储足够数量的中间值(这些中间值仅与签名者的私钥有关，而与具体消息无关，因此可以用于不同的消息)。当收到消息后，签名者使用预存的中间值进行在线计算产生签名，这样能最大限度发挥在线/离线签名优势，大幅提升在线签名效率，具体数值见表10。

对于资源受限的签名设备，其可以根据自身的存储能力，灵活地设置和调节所保存的中间值组数和空间，合理地调配设备存储资源，从而实现存储空间和签名效率之间的权衡，总体签名效率优于Dilithium，具体如下：

(1) 签名设备的存储空间仅能保存1组中间值。

Olithium方案与Dilithium方案所存储的中间值本质上等价，空间需求也完全相同。即使这种情况下，由表10和表11可知，使用Olithium方案仍能够将在线签名效率提升约100%，即在线签名时间缩短约50%。

(2) 签名设备的存储空间能保存少量几组中间值。

该设备可以在在线签名的过程中动态管理中间值。例如，抛弃因签名失败而无效的中间值并释放

相应存储空间，同时生成新的中间值以供后续签名使用。通过这种方式，相较于仅保存1组中间值的情况签名效率会进一步提升，具体组数选择和提升数值可参考表10。

(3) 签名设备的存储空间不足以保存1组中间值。

该设备无法使用本文所设计的Olithium方案以及现有的Dilithium方案进行签名。

6.2 Olithium方案的应用：数字证书颁发

在公钥密码学体系中，数字证书(Digital Certificates)在建立各参与方的信任的过程中发挥着至关重要的作用。这些证书以电子文档的形式存在，用于证明各参与方的公钥所有权和身份真实性，进而保障密码系统的安全可靠。

通常情况下，数字证书会由特定的数字证书颁发机构(Certificate Authority，CA)进行发放和管理。颁发机构作为受信任的第三方权威，其职责是利用自身的私钥对数字证书的内容进行签名，并将签名后的证书交付给请求者。数字签名技术的应用使得证书难以被伪造或篡改，从而确保证书的真实性和完整性，有效防止冒充行为的发生。随着量子时代的临近，需要设计更安全高效且适应证书颁发场景的抗量子签名方案，以用于证书的签名和颁发。本文提出的Olithium方案能够较好地满足这一需求。

具体而言，在交易过程中，无论是数据提供方、数据接收方，还是中间的数据处理平台，所有参与方都需要持有由证书颁发机构签名和颁发的数字证书，流程如图1所示。

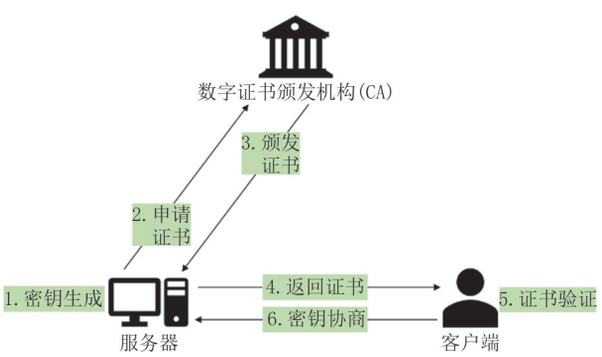

图1 数字证书颁发流程

系统模型主要包含三方实体：数字证书颁发机构、服务器和客户端。由于证书颁发机构拥有固定的密钥、充足的计算资源和存储空间，适合进行离线预计算，可以如下使用Olithium方案。

(1)数字证书颁发机构：该实体在整个系统中的作用是接收服务器请求，生成合法证书并进行签名。在未收到服务器请求或拥有空闲计算资源时，可以进行离线预计算，产生中间值并进行存储。一旦收到服务器请求，即可进行相应的在线签名。

(2)服务器：该实体在需要时向证书颁发机构发起请求，上传需要认证的信息，其中包括需签名的公钥。服务器收到颁发机构返回的证书后，将其发送给客户端以进行验证。

(3)客户端：该实体收到服务器发送的证书后，在通过合法性验证后与服务器进行密钥协商，并以此作为后续交流的基础。

在该系统中使用本文设计的Olithium方案，可以显著提升数字证书颁发机构这一频繁需要签名设备的在线签名效率。同时，由于Olithium方案是基于目前被广泛认为抗量子的格困难问题设计，即使在未来量子计算机投入使用后，该数字证书系统仍将具备值得信赖的安全性。

7 总结与展望

在传统密码方案面临量子计算威胁的背景下，后量子密码学的发展对于保障数据安全起到至关重要的作用。本文基于最新的NIST标准ML-DSA(即CRYSTALS-Dilithium)，设计了一个基于格的无陷门在线/离线签名方案Olithium。该方案允许签名者在未收到消息时(离线阶段)产生签名的一部分，并在收到消息后(在线阶段)继续完成签名。最终，在保持存储空间不明显增加的前提下，将在线阶段的签名时间缩短约50%。本文也以数字证书颁发作为例子，说明了Olithium方案在现实中的应用。

未来，可以从以下几个方面进一步研究：

(1)目前基于Fiat-Shamir范式的签名方案在签名过程中，由于拒绝抽样会发生多次中止并导致重复，这对签名性能有较大的影响。后续可以尝试减少重复次数，继续提升在线签名效率。

(2)使用Olithium的设计思想对基于Dilithium设计的后续签名方案和标准，以及具有更加丰富功能的签名方案进行改进，提升它们在现实场景中的实用性。

(3)基于其他目前被认为抗量子的困难问题，利用利用Γ-协议以及本文设计在线/离线签名方案，为未来量子计算机时代的网络安全提供丰富、高效和安全的密码学解决方案。

参 考 文 献

[1] Shor P W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Review, 1999, 41(2): 303-332

[2] Wang Ya-Hui, Zhang Huan-Guo, Wu Wan-Qing, et al. Quantum algorithms for breaking RSA based on phase estimation and equation solving. Chinese Journal of Computers, 2017, 40(12): 2688-2699 (in Chinese)
(王亚辉，张焕国，吴万青等. 基于方程求解与相位估计攻击RSA的量子算法 计算机学报，2017，40(12)：2688-2699)

[3] Zhang Huan-Guo, Mao Shao-Wu, Wu Wan-Qing, et al. Overview of quantum computation complexity theory. Chinese Journal of Computers, 2016, 39(12): 2403-2428 (in Chinese)
(张焕国，毛少武，吴万青等. 量子计算复杂性理论综述. 计算机学报，2016，39(12)：2403-2428)

[4] Yang Ya-Tao, Chang Xin, Shi Hao-Peng, et al. CDBS: Blind signature scheme based on CRYSTALS-Dilithium algorithm. Journal on Communications, 2024, 45(7): 184-195 (in Chinese)
(杨亚涛，常鑫，史浩鹏等. CDBS:基于CRYSTALS-Dilithium算法的盲签名方案. 通信学报，2024，45(7)：184-195)

[5] Wen Jia-Ming, Wang Hou-Zhen, Liu Jin-Hui, et al. Aitps: A two-party signature scheme from asymmetry module lattice problems. Journal of Computer Research and Development, 2023, 60(9): 2137-2151 (in Chinese)
(文嘉明，王后珍，刘金会等. Aitps：基于非对称模格问题的两方协同签名方案. 计算机研究与发展，2023，60(9)：2137-2151)

[6] Dong Yi-Fan, Fang Bo-Yue, Liang Zhi-Chuang, et al. Efficient lattice-based digital signature scheme in large-galois-group prime-degree Prime-ideal field. Journal of Software, 2025, 36(2): 776-804 (in Chinese)
(董怡帆，方博越，梁志闯等. 素阶数域上的高效格基数字签名方案. 软件学报，2025，36(2)：776-804)

[7] Li Yan-Bin, Zhu Jia-Jie, Tang Ming, et al. Power analysis attacks for lattice-based cryptography. Chinese Journal of Computers, 2023, 46(2): 331-352 (in Chinese)
(李延斌，朱嘉杰，唐明等. 面向格密码的能耗分析攻击技术. 计算机学报，2023，46(2)：331-352)

[8] Even S, Goldreich O, Micali S. On-line/off-line digital signatures. Journal of Cryptology, 1996, 9: 35-67

[9] Shamir A, Tauman Y. Improved online/offline signature schemes//Proceedings of the 21st Annual International Cryptology Conference (CRYPTO 2001). BarbaraSanta, USA, 2001, 355-367

[10] Kurosawa K, Schmidt-Samoa K. New online/offline signature schemes without random oracles//Proceedings of the International Conference on Theory and Practice of Public-Key Cryptography (PKC 2006). YorkNew, USA, 2006, 330-346

[11] Ateniese G, De Medeiros B. Identity-based chameleon hash and applications//Proceedings of the 8th International Conference Financial Cryptography (FC 2008). WestKey, USA, 2008, 164-180

[12] Chen X, Zhang F, Susilo W, et al. Efficient generic on-line/off-line signatures without key exposure//Proceedings of the 5th International Conference of Applied Cryptography and Network Security (ACNS 2007). Zhuhai, China, 2007, 18-30

[13] Zheng M, Yang S J, Wu W, et al. A new design of online/offline signatures based on lattice//Proceedings of the 14th International Conference of Information Security Practice and Experience (ISPEC 2018). Tokyo, Japan, 2018, 198-212

[14] Fiat A, Shamir A. How to prove yourself: Practical solutions to identification and signature problems//Proceedings of the 6th Annual International Cryptology Conference (CRYPTO 1986). BarbaraSanta, USA, 2001, 186-194

[15] Lyubashevsky V. Lattice signatures without trapdoors//Proceedings of the 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2012). Cambridge, UK, 2012, 738-755

[16] Yao A C C, Zhao Y. Online/offline signatures for low-power devices. IEEE Transactions on Information Forensics and Security, 2012, 8(2): 283-294

[17] Zhang P, Jiang H, Zheng Z, et al. A new and efficient lattice-based online/offline signature from perspective of abort. The Computer Journal, 2022, 65(9): 2400-2410

[18] Alkim E, Barreto P S L M, Bindel N, et al. The lattice-based digital signature scheme qTESLA//Proceedings of the 18th International Conference of Applied Cryptography and Network Security (ACNS 2020). Rome, Italy, 2020, 441-460

[19] Zhang P, Yang H, Zhu L, et al. A new lattice-based online/offline signatures framework for low-power devices. Theoretical Computer Science, 2023, 962: 113942

[20] Katz J. Digital Signatures. Berlin, Germany: Springer, 2010

[21] Bai S, Ducas L, Kiltz E, et al. Crystals-Dilithium: Algorithm specifications and supporting documentation (version 3.1). round-3 submission to the nist pqc standardization project, Gaithersburg, Maryland, USA, 2021

WANG Hou-Zhen, associate professor. His main research interests include cryptography and post-quantum cryptography.

DUAN Xiao-Chao, M. S. candidate. His main research interests include cryptography and post-quantum cryptography.

WEN Jia-Ming, Ph. D. candidate. His main research interests include cryptography and post-quantum cryptography.

WANG Ya-Hui, Ph. D., lecturer. Her main research interests include cryptography and quantum computing.

ZHANG Huan-Guo, Ph. D., professor, Ph. D. supervisor. His main research interests include information security and cryptography.

Background

This paper investigates the optimized implementation of post-quantum cryptography. Currently, secure data transmission and usage rely on robust cryptographic technologies. However, recent advances in quantum algorithms and quantum computers have posed unprecedented threats to these techniques. To address this challenge, the U. S. National Institute of Standards and Technology (NIST) launched the PQC project in 2016 to solicit relevant standards, and in 2023 it released a draft of the ML-DSA post-quantum signature standard based on the CRYSTALS-Dilithium algorithm. Likewise, the Chinese Association for Cryptologic Research also organized a national cryptographic algorithm design competition in 2019, where the Aigis-sig signature scheme, which shares a similar structure to Dilithium, won first prize in the signature category.

To further enhance the signing efficiency of Dilithium and Aigis-sig, this paper proposes a novel online/offline signature scheme named Olithium. It enables the generation of a partial signature in advance (offline phase), allowing the completion time of the signature to be significantly shortened once the message is received (online phase). Although this approach requires slightly more storage, it substantially improves signing efficiency, making it well-suited for scenarios that require rapid processing of large datasets, such as big data and cloud computing applications. This paper provides a complete security proof, and details experimental results, demonstrating that average signing time across various security levels can be reduced by approximately 50%. A practical example of digital certificate issuance is provided to illustrate the real-world applicability of our scheme.

Our group has over a decade of experience in post-quantum cryptography and quantum computing, making numerous academic contributions, such as the references [2, 3, 5, 7] in this paper. Our research was supported by a closely related National Natural Science Foundation of China Key Program, "Research on Public-Key Cryptography Theory and Key Technologies for Quantum Computers" (No. 61332019), and is being supported by the National Key R&D Program of China (No. 2022YFB4500800), the Hubei Key R&D Program (No. 2022BAA041), and the China Scholarship Council Visiting PhD Student Program (No. 202306270167).