计 算 机 研 究 与 发 展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202220533
60(9): 2137−2151, 2023

Aitps: 基于非对称模格问题的两方协同签名方案

文嘉明[1,2] 王后珍[1,2,3] 刘金会[1,2,4] 张焕国[1,2]

[1](武汉大学国家网络安全学院 武汉 430072)

[2](空天信息安全与可信计算教育部重点实验室(武汉大学) 武汉 430072)

[3](密码科学技术国家重点实验室 北京 100878)

[4](西北工业大学网络空间安全学院 西安 710072)

(wenjm@whu.edu.cn)

Aitps: A Two-Party Signature Scheme from Asymmetry Module Lattice Problems

Wen Jiaming[1,2], Wang Houzhen[1,2,3], Liu Jinhui[1,2,4], and Zhang Huanguo[1,2]

[1](*School of Cyber Science and Engineering*, *Wuhan University*, *Wuhan* 430072)

[2](*Key Laboratory of Aerospace Information Security and Trusted Computing* (*Wuhan University*), *Ministry of Education*, *Wuhan* 430072)

[3](*State Key Laboratory of Cryptology*, *Beijing* 100878)

[4](*School of Cyber Security*, *Northwestern Polytechnical University*, *Xi'an* 710072)

Abstract Recent years, with the advancement of the IoT and blockchain, multi-party signature protocols have received renewed attention. Multi-party signature is a special digital signature that requires users to interact with each other to jointly generate a signature for a message and achieve the authentication. Compared with each user signing respectively, the advantage is that the key size can be greatly decreased, and every party cannot get a legal signature only by itself, which can be used to prevent the danger of being impersonated when user's key is lost or hijacked. On the other hand, the progress of quantum computers poses a potential threat to the traditional public key cryptography scheme, the PQC(post-quantum cryptography) project was organized by the NIST(National Institute of Standards and Technology) in the US since 2016, and it determined the algorithm that was standardized in July 2022. At the same time, the multi-party signature based on its candidate digital signature schemes (such as CRYSTALS-Dilithium) also appeared. Chinese Association for Cryptologic Research(CACR) also held a national cryptographic algorithm design competition in 2019, Aigis-sig, which is the first prize signature algorithm, adopts the similar structure with Dilithium. In this paper, Aitps is proposed, which is a two-party signature based on Aigis-sig. Compared with the existing Dilithium-based two-party signatures, Aitps has better key sizes and signature sizes. For example, the signature sizes can be reduced by more than 20% at the same security level. Lastly, Aitps can also be extended to multi-party signature.

Key words digital signature; two-party signature; lattice-based cryptography; post-quantum cryptography; key protection

摘 要 物联网和区块链等技术的兴起和发展,使得多方协同签名协议重新受到了关注.多方协同签名是一种特殊的数字签名,要求多个用户进行交互后共同对一个消息产生合法的签名,以达到认证的目的.优

收稿日期: 2022-06-11; **修回日期**: 2022-10-10

基金项目: 国家重点研发计划项目(2022YFB4500800); 国家自然科学基金项目(62272385, 62272389, U19B2021); 中央高校基本科研业务费专项资金(2042022kf0021)

This work was supported by the National Key Research and Development Program of China(2022YFB4500800), the National Natural Science Foundation of China (62272385, 62272389, U19B2021), and the Fundamental Research Funds for the Central Universities(2042022kf0021).

通信作者: 王后珍(whz@whu.edu.cn)

点在于相比起每个用户分别进行签名可以缩短尺寸,同时使用分布式的方法,任何一方都无法独自进行签名,防范了因为单个用户的密钥丢失或被劫持而导致被冒充身份的隐患.另一方面,量子计算机的进展对传统的公钥密码方案构成了潜在的威胁,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(post-quantum cryptography, PQC)的国际标准征集项目,并于2022年7月确定了被选为标准的算法.同时,基于其入选的数字签名方案(例如CRYSTALS-Dilithium)的协同签名方案也已经陆续出现.2019年,中国密码学会举办了全国密码算法设计竞赛,其中公钥组获得一等奖的Aigis-sig签名方案采用了与Dilithium类似的结构.基于Aigis-sig数字签名方案设计了一种两方协同签名方案,称之为Aitps,并根据其提供的参数进行了实例化和对比,得到了相比已有的所有基于Dilithium的两方协同签名方案更优的密钥和签名大小,例如在同等的安全强度下签名尺寸可缩减20%以上.此外,该方案也可以扩展为多方协同签名.

关键词 数字签名;两方协同签名;基于格的密码学;抗量子密码;密钥保护

中图法分类号 TP309

随着互联网的飞速发展,手机等移动设备的适用范围不断扩大.根据中国互联网络信息中心发布的第49次中国互联网报告[1],截至2021年12月,我国网民规模已达10.32亿人次,其中即时通信用户规模达10.07亿人次,网络支付用户规模达9.04亿人次,网络购物用户规模达8.42亿人次,在线办公用户规模达4.69亿人次.移动设备日渐丰富的功能也带来了更严重的隐私信息泄露问题.作为传统签名的替代,数字签名伴随着网络安全的需求出现,为用户提供身份认证和数据完整性认证等,在移动设备的各个功能中发挥重要作用.

然而,数字签名的安全性建立在签名密钥安全的基础上,如果密钥不慎泄露,或者被恶意的网站和应用程序等窃取,可能会出现恶意者冒充等情况,导致隐私信息被滥用、财产损失甚至威胁到国家安全.目前用来保护签名密钥安全的方法主要分为2种[2]:借助硬件设备令牌保护密钥和使用多方协同签名的思想保护密钥.受限于便利性和成本,前者难以大规模部署到移动设备或物联网设备中;而后者则是由2个及以上的设备分别存储密钥的一部分,签名时需要多个设备交互完成,任何一方都无法独立地进行签名,分布式的处理能较好地保护密钥安全.

多方协同签名是一种特殊的门限签名,参与签名的用户P_i首先运行密钥生成算法获得私钥sk_i,然后通过与其余用户的交互生成公钥pk.签名过程也需要交互,对于给定的消息μ,如果所有的用户都同意对μ进行签名并诚实参与交互,则交互后得到一个能用pk验证的合法消息签名对(μ, σ).

尽管多方协同签名协议已经被研究了很长时间,然而现有的大多数工作都集中在多方协同RSA签名[3-4]、多方协同ECDSA签名[5-7]、多方协同Schnorr签名[8-10]以及多方协同SM2签名[2]等,这些方案都基于整数分解或者离散对数困难假设,Shor[11]已经证明了这些困难假设无法抵抗量子计算机的攻击.

相比之下,基于格上的困难假设构造的密码学方案,目前被认为是抗量子的,同时具有运算快、平均情况和最坏情况困难性等价等优点[12],受到了研究者们的广泛关注.美国国家标准技术研究院(National Institute of Standards and Technology, NIST)举办的抗量子密码(post-quantum cryptography, PQC)征集项目的抗量子密钥封装和数字签名方案,就有相当一部分是基于格的方案,例如CRYSTALS-Dilithium[13], CRYSTALS-Kyber[14], Falcon[15]等.中国密码学会在2019年举办的全国密码算法设计竞赛中的获奖算法大部分也是基于格困难问题构造的,例如Aigis-enc/sig[16]和LAC.PKE[17]等.其中Aigis设计团队观察到Dilithium签名方案实际上可以非对称地选择参数,在保证总的重复次数相当的情况下,通过改变前后2个拒绝抽样的条件及其决定的重复次数,能取得更佳的效果.因而,该团队使用了参数选取更加灵活的非对称模格问题AMLWE/AMSIS(asymmetry module learning with errors/short integer solutions)来代替Dilithium中的模格问题MLWE/MSIS(module learning with errors/short integer solutions),设计出Aigis-sig签名方案,在安全性不变或略强的前提下达到更好的综合效果,特别是拥有更短的公钥、私钥和签名长度,具体参见1.5节.值得一提的是,基于AMLWE/AMSIS的Aigis方案不仅性能优秀,它还是我国学者自主设计的抗量子密码方案,也已经出现在不同平台上对其实现进行优化的相关工作[18-19].除了提升算法本身

和优化实现之外，基于其设计特殊类型签名，例如能更好地保护密钥安全的多方协同签名，对研究国产抗量子密码算法、维护网络安全和国家安全也具有重要的意义.

2019 年，Cozzo 等人[20] 对 NIST 征集的抗量子数字签名方案中所有进入第 2 轮的算法转换成多方协同签名（分布式签名）进行了评估，得出的结论是：如果直接使用已有的安全多方计算的通用技术，基于格的方案将需要用到线性秘密共享和混淆电路等，以及它们之间的互相转换，会带来较大的计算开销，需要较长的时间.

2022 年，Damgard 等人[21] 利用 Lyubashevsky[22] 提出的构造基于格的数字签名的“FSwA（Fiat-Shamir with aborts）”范式，得到了 2 个交互轮数较小的基于格的多方协同签名协议（使用加法同态承诺方案需要交互 3 轮，使用陷门加法同态承诺方案仅需要交互 2 轮），文中的秘密向量是从离散高斯分布中选取的，承诺方案则来自于文献 [23] 及其扩展方案，该协同签名方案可以看作 Dilithium 签名方案的分布式版本，文献 [23] 中给出了完整的安全性证明，其安全性基于 MSIS 问题和 MLWE 问题.

Vakarjuk 等人[24] 也给出了一个 3 轮的两方协同签名方案——Dilizium，与文献 [21] 中方案的不同之处在于 Dilizium 使用 SWIFFT 同态 Hash 函数[25] 替换加法同态承诺方案，虽然得到了更小的密钥和签名尺寸，但是其缺点在于依赖 Rejected MLWE 困难假设，是一种启发式的困难假设[26]，同时 SWIFFT 同态 Hash 函数并不是对所有输入都是加法同态的. 现在也已经出现了对具备同态性的 Hash 函数的量子攻击方法[27]，这可能会威胁到安全性，尽管如此，我们仍将该方案纳入对比.

此外，文献 [21, 24] 中基于 Dilithium 设计的两方协同签名方案均未考虑 Dilithium 中用到的签名尺寸压缩技巧[28]，而最新方案 Dilizium 2.0[29] 采用了类似压缩技术，使其更接近于 NIST 标准化的 Dilithium 算法，然而该工作并没有评估实际的效率（重复次数）、密钥和签名尺寸等，也未进行参数选取和实例化.

本文的主要贡献包括 3 个方面：

1）采用 AMSIS/AMLWE 问题对基于 MSIS/MLWE 问题的 Dilizium 2.0 两方协同签名方案进行了修改和推广，得到新方案 Aitps. 充分利用 AMSIS/AMLWE 的非对称特性能够更灵活地选择参数，从而在安全性、计算效率、密钥和签名长度这 3 个方面达到更好的权衡，得到更优的综合性能. 值得注意的是，Aitps 方案可以被看作是 Dilizium 2.0 方案的一个扩展，在选取适当参数时，Aitps 和 Dilizium 2.0 本质上等价，即 Dilizium 2.0 可以看成 Aitps 的一个特例. 除此以外，Dilithium 设计团队和 Aigis 设计团队对其方案的后续优化，也适用于本文提出的方案.

2）使用“FSwA”范式构造基于格的多方签名在安全性上需要解决的一个关键问题是防止未通过拒绝抽样时的私钥信息泄露，我们使用同态承诺解决了这个问题（见第 2 节）. 对于新设计的 Aitps 两方协同签名方案，我们提供了完整的安全性证明，结果表明其可以有效保护各方的签名密钥，具备两方协同签名在选择消息攻击下的存在性不可伪造性[5].

3）为了对比效果，本文给出了 Aitps 方案的重复次数、密钥和签名大小的评估方法，该评估方法也适用于 Dilizium 2.0 方案. 我们使用 Dilithium 第 2 轮的参数[13] 和 Aigis-sig 的参数[16] 将 Dilizium 2.0 和 Aitps 全部进行了实例化，并将 Dilizium[24]（其密钥和签名尺寸优于文献 [21]）也进行计算并纳入对比，相比之下，本文方案的密钥以及签名尺寸优于现有的所有基于 Dilithium 的两方协同签名方案，例如在同等的安全强度下，签名尺寸可缩减 20% 以上. 据我们所知，该结果也是基于格的两方协同签名方案中最优的.

1 预备知识

1.1 符号说明

本文用$\boldsymbol{A}^{\mathrm{T}}$表示矩阵$\boldsymbol{A}$的转置，$\boldsymbol{I}_k$表示$k$阶单位矩阵，$\lceil x\rceil$表示大于等于实数$x$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，$n-1$即为环中的多项式的最高次数，实际方案中一般选取n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方案的其他参数和工程实现的效率. 表 1 给出了一套具体参数的例子.

对于多项式$w=w_0+w_1x+\cdots+w_{n-1}x^{n-1}\in R$，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$\boldsymbol{p}=(p_1,p_2,\cdots,p_k)^{\mathrm{T}}\in R^k$，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. R_q中的定义类似.

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

Table 1 Recommended Parameters of Dilithium and Aigis-sig

表 1 Dilithium 和 Aigis-sig 的推荐参数

参数类型	Dilithium	安全级别			参数类型	Aigis-sig	安全级别		
		128 b	192 b	256 b			128 b	192 b	256 b
输入参数设置	多项式次数 n	256	256	256	输入参数设置	多项式次数 n	256	256	256
	矩阵行数 k	4	5	6		矩阵行数 k	4	5	6
	矩阵列数 l	3	4	5		矩阵列数 l	3	4	5
	模数 q	8 380 417	8 380 417	8 380 417		模数 q	2 021 377	3 870 721	3 870 721
	私钥 sk_1, sk_2 范围η	6	5	3		私钥 sk_1 范围η	2	2	1
						私钥 sk_2 范围η'	3	5	5
	c 中±1个数τ	60	60	60		c 中±1个数τ	60	60	60
	签名截取参数β	325	275	175		签名截取参数β	120	120	60
						签名截取参数β'	175	275	275
	签名范围参数γ	523 776	523 776	523 776		签名范围参数γ	131 072	131 072	131 072
	签名范围参数 γ'	261 888	261 888	261 888		签名范围参数 γ'	168 448	322 560	322 560
输出参数性能	公钥大小/ B	1 184	1 472	1 760	输出参数性能	公钥大小/ B	1 056	1 312	1 568
	私钥大小/ B	2 800	3 504	3 856		私钥大小/ B	2 448	3 376	3 888
	签名大小/ B	2 044	2 701	3 366		签名大小/ B	1 852	2 445	3 046
	期望重复次数	5.90	6.60	4.30		期望重复次数	5.86	7.61	6.67
	经典安全性	100	141	174		经典安全性	99	141	179
	量子安全性	91	128	158		量子安全性	90	128	163

项式$w \in R$(或R_q)组成的集合.

对于集合(或分布)D, 用$x \leftarrow_{\$} D$表示从集合(或按照分布)$D$中均匀随机选取$x$.

1.2 格上的困难问题

本文直接采用正规型(A)MLWE/(A)MSIS 问题及假设, 定义如下.

定义 1. MLWE 问题[13, 30]. 参数为(q, k, l, η)的MLWE 问题指的是对于给定的正整数q, k, l以及$\eta > 0$, 定义分布D_1和D_2为:

1) $D_1: \boldsymbol{A} \leftarrow_{\$} R_q^{k\times l}, (\boldsymbol{s}_1, \boldsymbol{s}_2) \leftarrow_{\$} S_\eta^l \times S_\eta^k$, 计算得到$\boldsymbol{t} = \boldsymbol{A}\boldsymbol{s}_1 + \boldsymbol{s}_2 \in R_q^k$.

2) $D_2: \boldsymbol{A} \leftarrow_{\$} R_q^{k\times l}, \boldsymbol{t} \leftarrow_{\$} R_q^k$.

由D_1中的$(\boldsymbol{A}, \boldsymbol{t})$恢复$(\boldsymbol{s}_1, \boldsymbol{s}_2)$称为计算性 MLWE (computational MLWE)问题, 区分 2 个分布D_1和D_2中的$(\boldsymbol{A}, \boldsymbol{t})$称为判定性 MLWE(decisional MLWE)问题.

定义 2. AMLWE 问题[16]. 参数为(q, k, l, η, η')的AMLWE 问题指的是对于给定的正整数q, k, l以及$\eta, \eta' > 0$, 定义分布D_3和D_4为:

1) $D_3: \boldsymbol{A} \leftarrow_{\$} R_q^{k\times l}, (\boldsymbol{s}_1, \boldsymbol{s}_2) \leftarrow_{\$} S_\eta^l \times S_{\eta'}^k$, 计算得到$\boldsymbol{t} = \boldsymbol{A}\boldsymbol{s}_1 + \boldsymbol{s}_2 \in R_q^k$.

2) $D_4: \boldsymbol{A} \leftarrow_{\$} R_q^{k\times l}, \boldsymbol{t} \leftarrow_{\$} R_q^k$.

由D_3中的$(\boldsymbol{A}, \boldsymbol{t})$恢复$(\boldsymbol{s}_1, \boldsymbol{s}_2)$称为计算性 AMLWE 问题(computational AMLWE), 区分 2 个分布D_3和D_4中的$(\boldsymbol{A}, \boldsymbol{t})$称为判定性 AMLWE 问题(decisional AMLWE).

Zhang 等人[16]已经证明, 对于目前已知最好的求解算法, 有困难性关系:

$$\mathrm{MLWE}_{q,k,l,\min(\eta,\eta')} \leqslant \mathrm{AMLWE}_{q,k,l,\eta,\eta'} \leqslant \mathrm{MLWE}_{q,k,l,\max(\eta,\eta')}.$$

定义 3. MSIS 问题[13]. 参数为(q, k, l, δ)的 MSIS 问题指的是对于正整数q, k, l以及$\delta > 0$和矩阵$\boldsymbol{A} \in R_q^{k\times l}$, 计算非零向量$\boldsymbol{x} \in R_q^{k+l}$使其满足$[\boldsymbol{A} \| \boldsymbol{I}_k]\boldsymbol{x} = \boldsymbol{0} \bmod q$且$\|\boldsymbol{x}\|_\infty \leqslant \delta$.

定义 4. AMSIS 问题[16]. 参数为$(q, k, l, \delta, \delta')$的AMSIS 问题指的是对于正整数$q, k, l$以及$\delta, \delta' > 0$和矩阵$\boldsymbol{A} \in R_q^{k\times l}$, 计算非零向量$\boldsymbol{x} = (\boldsymbol{x}_1^{\mathrm{T}}, \boldsymbol{x}_2^{\mathrm{T}}) \in R_q^{k+l}$使其满足$[\boldsymbol{A} \| \boldsymbol{I}_k]\boldsymbol{x} = \boldsymbol{0} \bmod q$且$\|\boldsymbol{x}_1\|_\infty \leqslant \delta, \|\boldsymbol{x}_2\|_\infty \leqslant \delta'$.

Zhang 等人[16]已经证明, 对于目前已知最好的求解算法, 有困难性关系:

$$\mathrm{MSIS}_{q,k,l,\max(\delta,\delta')} \leqslant \mathrm{AMSIS}_{q,k,l,\delta,\delta'} \leqslant \mathrm{MSIS}_{q,k,l,\min(\delta,\delta')}.$$

1.3 (同态)承诺方案

承诺方案最早由 Chor 等人[31]在研究可验证的秘密分享时提出, 现在已经被用在各种特殊类型签名中, 例如环签名[32-33]、群签名[34]等. 在承诺方案中, 对于给定的消息$\boldsymbol{m} \in \mathcal{S}_m$, 承诺者选择随机向量$\boldsymbol{r} \in \mathcal{S}_r$用来计算$\boldsymbol{m}$的承诺值$\boldsymbol{com} = Commit_{\boldsymbol{ck}}(\boldsymbol{m}; \boldsymbol{r})$, 其中$\boldsymbol{ck} \in \mathcal{S}_{\boldsymbol{ck}}$

是承诺密钥，并将承诺值$\boldsymbol{com}$发送给接收者. 在承诺打开阶段，承诺者将与承诺值相关的信息发送给接收者，接收者能利用承诺值以及相关信息计算得到一个打开值(opening)，并验证其确实是最初承诺的消息值.

除了具备正确性之外，承诺方案还需要具备隐藏性和绑定性2个安全属性.

1) 隐藏性(hiding)指的是承诺值不会泄露被承诺值的任何信息，即通过$\boldsymbol{com}=Commit_{\boldsymbol{ck}}(\boldsymbol{m};\boldsymbol{r})$无法得到关于$\boldsymbol{m}$或$\boldsymbol{r}$的信息.

2) 绑定性(binding)指的是打开一个承诺不能得到2个不同的值，即打开$\boldsymbol{com}=Commit_{\boldsymbol{ck}}(\boldsymbol{m};\boldsymbol{r})$得到$(\boldsymbol{m}',\boldsymbol{r}')$，则$(\boldsymbol{m}',\boldsymbol{r}')\neq(\boldsymbol{m},\boldsymbol{r})$的概率是可忽略的.

根据敌手的能力和计算时间进行分类：若敌手拥有无限计算时间与资源则称为统计隐藏性(绑定性)，若限制在多项式时间则称为计算隐藏性(绑定性).

本文使用Esgin等人[35]提出的基于格的承诺方案.先产生承诺密钥$\boldsymbol{ck}=(\boldsymbol{A}_1\|\boldsymbol{A}_2)$，其中$\boldsymbol{A}_1=[\boldsymbol{I}_k\|\boldsymbol{A}_1']$，$\boldsymbol{A}_1'\leftarrow_{\$}R_q^{n\times(k-n)}$，$\boldsymbol{A}_2\leftarrow_{\$}R_q^{n\times l}$都是均匀随机选取的. 在对消息$\boldsymbol{m}\in R_q^l$进行承诺时，选取随机向量$\boldsymbol{r}\leftarrow_{\$}\mathcal{S}_r=S_\alpha^\kappa$，得到承诺$\boldsymbol{com}=\boldsymbol{A}_1\cdot\boldsymbol{r}+\boldsymbol{A}_2\cdot\boldsymbol{m}$. 在承诺打开阶段，用消息$\boldsymbol{m}$、随机数$\boldsymbol{r}$以及承诺值$\boldsymbol{com}$，验证$\boldsymbol{com}=\boldsymbol{A}_1\cdot\boldsymbol{r}+\boldsymbol{A}_2\cdot\boldsymbol{m}$且$\|(\boldsymbol{r},\boldsymbol{m})\|_2$不超过某个阈值，若均通过则输出1，否则输出0.

在满足正确性和安全性后，该承诺方案还具备3个性质[21]：

1) 密钥均匀性(uniform key). 产生承诺密钥的算法得到的承诺密钥$\boldsymbol{ck}\in\mathcal{S}_{\boldsymbol{ck}}$在$\mathcal{S}_{\boldsymbol{ck}}$上均匀分布.

2) ξ-bits最小熵.称一个承诺方案至少有ξ-bits最小熵，如果对于$\forall\boldsymbol{ck}\in\mathcal{S}_{\boldsymbol{ck}}$和$\forall\boldsymbol{m}\in\mathcal{S}_{\boldsymbol{m}}$，均有

$$\xi\leqslant-\mathrm{lb}\max_{\boldsymbol{com}\in\mathcal{S}_{com}}Pr[Commit_{\boldsymbol{ck}}(\boldsymbol{m};\boldsymbol{r})=\boldsymbol{com}:\boldsymbol{r}\leftarrow\mathcal{S}_r].$$

3) 加法同态性. 对于任意的$\boldsymbol{m}_1,\boldsymbol{m}_2\in\mathcal{S}_{\boldsymbol{m}}$以及随机数$\boldsymbol{r}_1,\boldsymbol{r}_2\in\mathcal{S}_r$，计算得到$\boldsymbol{com}_1=Commit_{\boldsymbol{ck}}(\boldsymbol{m}_1;\boldsymbol{r}_1)$和$\boldsymbol{com}_2=Commit_{\boldsymbol{ck}}(\boldsymbol{m}_2;\boldsymbol{r}_2)$，满足加法同态性，即

$$Open_{\boldsymbol{ck}}(\boldsymbol{m}_1+\boldsymbol{m}_2,\boldsymbol{r}_1+\boldsymbol{r}_2,\boldsymbol{com}_1+\boldsymbol{com}_2)=1$$

1.4 CRYSTALS-Dilithium数字签名方案

2022年7月，NIST宣布了PQC项目第3轮的评审结果[36]，确定了4种即将标准化的算法，其中最为推荐的是CRYSTALS-Kyber(密钥封装)和CRYSTALS-Dilithium(数字签名)，此外，另一个基于格的签名方案Falcon和基于Hash的签名方案SPHINCS+[37]也将标准化. 在数字签名方面，根据设计团队向NIST提交的材料以及官方评价显示，Falcon是利用“Hash-and-Sign”范式构造，虽然拥有更短的尺寸，但其签名算法内部逻辑较复杂，实现难度较大；SPHINCS+是一类基于Hash的无状态签名方案，其安全性依赖于底层Hash函数的安全性，提供可靠的安全保证，然而会导致性能上的巨大成本. 相对而言，Dilithium拥有很强的安全性和优秀的性能，能胜任绝大多数场景需求.

Dilithium是一类基于格上的困难问题构造的数字签名算法，算法的设计用到了“Fiat-Shamir with Aborts”范式，并使用了一些压缩技巧[28, 38]，主要具备3个优点[13]：

1) 容易安全地实现. 此前的基于格的数字签名方案[39-40]等需要从离散高斯分布中抽样得到秘密值，效率较低，还容易遭到侧信道攻击而导致实现的不安全[41-42]. 与它们不同，Dilithium签名方案只需要进行均匀抽样，且除了抽样之外，其余的运算操作(例如多项式乘法和舍入)都可以在恒定时间内完成，这有利于增强实现上的安全性.

2) 公钥+签名尺寸优. 为了能长期使用，Dilithium团队提交给NIST的参数选取非常保守，即便如此，Dilithium方案的“公钥+签名”的尺寸也是现有的不使用离散高斯抽样的格签名方案中最小的.

3) 模块化切换. 只需在环上进行更多或更少的操作，或者修改其中所用的可扩展输出长度Hash函数(XOF，推荐使用SHAKE-128或SHAKE-256)就可以切换不同级别的安全性. 换句话说，一旦获得某个安全级别的更优化实现，就很容易获得其他安全级别的更优化的实现.

在介绍Dilithium签名方案之前，先描述其中需要使用的高低位比特分解算法[13]$Decompose_q(\theta,\lambda)$，该算法输入整数$\theta\in\mathbb{Z}_q$和一个小的正整数$\lambda$，满足$\rho|(q-1)$，按照3步操作将$\theta$分解得到$\theta=\theta_{\mathrm{H}}\cdot\lambda+\theta_{\mathrm{L}}$，其中$0\leqslant\theta_{\mathrm{H}}<\dfrac{q-1}{\lambda}$且$\|\theta_{\mathrm{L}}\|_\infty\leqslant\dfrac{\lambda}{2}$. 并将$\theta_{\mathrm{H}}=HighBits_q(\theta,\lambda)$称为$\theta$的高位比特，将$\theta_{\mathrm{L}}=LowBits_q(\theta,\lambda)$称为$\theta$的低位比特.

1) 将θ取mod q落到区间$0\leqslant\theta<q$中，得到$\theta:=\theta\bmod^+q$.

2) 将步骤1)得到的θ取mod λ落到区间$-\dfrac{\lambda}{2}<\theta\leqslant\dfrac{\lambda}{2}$(或$-\dfrac{\lambda-1}{2}<r\leqslant\dfrac{\lambda-1}{2}$)中，得到$\theta_{\mathrm{L}}:=\theta\bmod^\pm\lambda$.

3) 如果$\theta-\theta_{\mathrm{L}}=q-1$则令$\theta_{\mathrm{H}}:=0,\theta_{\mathrm{L}}:=\theta_{\mathrm{L}}-1$，否则令$\theta_{\mathrm{H}}:=(\theta-\theta_{\mathrm{L}})/\lambda$，输出$(\theta_{\mathrm{H}},\theta_{\mathrm{L}})$.

将$Decompose_q(\cdot)$算法作用于多项式(例如环R_q中的元素)或由多项式组成的向量、矩阵时，表示对应操作被分别独立地作用到多项式的每个系数. 使用

$Decompose_q(\cdot)$算法，能对任意的由R_q中的多项式组成的向量$\boldsymbol{\Theta}$和小范数向量$\boldsymbol{\Lambda}$，在不保存$\boldsymbol{\Theta}$的情况下恢复$\boldsymbol{\Theta}+\boldsymbol{\Lambda}$的高位比特，其正确性由引理 1 保证：

引理 1[13]. $\boldsymbol{\Theta},\boldsymbol{\Lambda}$是由$R_q$中的多项式组成的向量，$\rho_1$和$\rho_2$是正整数，如果$\|\boldsymbol{\Lambda}\|_\infty \leqslant \rho_2$且$\|LowBits_q(\boldsymbol{\Theta},\rho_1)\|_\infty < \frac{\rho_1}{2}-\rho_2$，则等式$HighBits_q(\boldsymbol{\Theta},\rho_1) = HighBits_q(\boldsymbol{\Theta}+\boldsymbol{\Lambda},\rho_1)$成立.

Dilithium 签名方案包括 3 个子算法：密钥生成算法、签名算法和验证算法，这里介绍不考虑压缩公钥$\boldsymbol{t}$的简化版本.

1）密钥生成算法. 首先使用种子生成矩阵$\boldsymbol{A}$，然后均匀随机选取$sk_1=\boldsymbol{s}_1\leftarrow_{\$}S_\eta^l$，$sk_2=\boldsymbol{s}_2\leftarrow_{\$}S_\eta^k$并计算$\boldsymbol{t}=\boldsymbol{A}\boldsymbol{s}_1+\boldsymbol{s}_2$，公钥$pk=(\boldsymbol{A},\boldsymbol{t})$，私钥$sk=(\boldsymbol{A},\boldsymbol{t},\boldsymbol{s}_1,\boldsymbol{s}_2)$.

2）签名算法. 对消息μ进行签名时，首先均匀随机选取$\boldsymbol{y}\leftarrow_{\$}S_{\gamma-1}^l$并计算$\boldsymbol{w}:=\boldsymbol{A}\boldsymbol{y}$，使用$Decompose_q(\cdot)$算法得到$\boldsymbol{w}$的高位比特$\boldsymbol{w}_{\mathrm{H}}$和低位比特$\boldsymbol{w}_{\mathrm{L}}$. 使用 Hash 函数$H_0$计算挑战值$c\in C$，从而得到$\boldsymbol{z}:=\boldsymbol{y}+c\boldsymbol{s}_1$，在通过拒绝抽样后输出合法的签名$\sigma=(z,c)$，其中$\beta$满足$\|c\boldsymbol{s}_1\|_\infty\leqslant\beta,\|c\boldsymbol{s}_2\|_\infty\leqslant\beta$.

算法 1. Dilithium-签名算法.

输入：消息μ，私钥$sk=(\boldsymbol{A},\boldsymbol{t},\boldsymbol{s}_1,\boldsymbol{s}_2)$；

输出：签名$\sigma=(z,c)$.

① 初始时设置$\boldsymbol{z}:=\perp$；

② while $\boldsymbol{z}=\perp$ do

③ $\quad\boldsymbol{y}\leftarrow_{\$}S_{\gamma-1}^l$；

④ $\quad\boldsymbol{w}:=\boldsymbol{A}\boldsymbol{y}$；

⑤ $\quad\boldsymbol{w}_{\mathrm{H}}=HighBits_q(\boldsymbol{w},2\gamma')$；

⑥ $\quad c:=H_0(\mu\|\boldsymbol{w}_{\mathrm{H}})\in C$；

⑦ $\quad\boldsymbol{z}:=\boldsymbol{y}+c\boldsymbol{s}_1$；

⑧ $\quad$if $\|\boldsymbol{z}\|_\infty\geqslant\gamma-\beta$ 或

$\qquad\left\|LowBits_q(\boldsymbol{A}\boldsymbol{y}-c\boldsymbol{s}_2,2\gamma')\right\|_\infty\geqslant\gamma'-\beta$ then

⑨ $\qquad\boldsymbol{z}:=\perp$；

⑩ $\quad$end if

⑪ end while

⑫ return $\sigma=(z,c)$.

3）验证算法. 在得到μ和$\sigma=(z,c)$后，首先利用$Decompose_q(\cdot)$算法计算$\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t}$的高位比特，然后根据$\boldsymbol{z}$的范围和挑战值$c$的正确性判断签名合法性.

算法 2. Dilithium-验证算法.

输入：消息μ以及对应的签名$\sigma=(z,c)$，公钥$pk=(\boldsymbol{A},\boldsymbol{t})$；

输出：1(接受签名)/0(拒绝签名).

① $\boldsymbol{w}'_{\mathrm{H}}:=HighBits_q(\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t},2\gamma')$；

② if $\|\boldsymbol{z}\|_\infty<\gamma-\beta$ 且 $c=H_0(\mu\|\boldsymbol{w}'_{\mathrm{H}})$ then

③ $\quad$return 1；

④ else return 0；

⑤ end if

更详细的 Dilithium 签名方案的正确性以及安全性分析参见文献 [13].

1.5 Aigis-sig 数字签名方案

2020 年 1 月，中国密码学会发布了全国密码算法设计竞赛的结果，Aigis-sig 数字签名方案是公钥密码组获得一等奖的 3 个方案中唯一的签名方案，并且其对应的密钥封装方案 Aigis-enc 同样也获得了一等奖，在国内抗量子公钥密码设计中具有高度评价，经过进一步的改进后，将成为我国自主设计的重要抗量子密码方案.

Aigis-sig 的主要设计思想与 Dilithium 类似，改进之处在于使用了更加灵活的非对称的格上困难问题——AMSIS 问题和 AMLWE 问题，通过改变私钥s_2的选取范围以及拒绝抽样的条件，得到了更优的公钥尺寸或签名尺寸，以及更强的安全性. Aigis-sig 签名方案包括 3 个子算法：密钥生成算法、签名算法和验证算法，这里介绍不考虑压缩公钥$\boldsymbol{t}$的简化版本.

1）密钥生成算法. 首先使用种子生成矩阵$\boldsymbol{A}$，然后均匀随机选取$sk_1=\boldsymbol{s}_1\leftarrow_{\$}S_\eta^l$，$sk_2=\boldsymbol{s}_2\leftarrow_{\$}S_\eta^k$并计算$\boldsymbol{t}=\boldsymbol{A}\boldsymbol{s}_1+\boldsymbol{s}_2$，公钥$pk=(\boldsymbol{A},\boldsymbol{t})$，私钥$sk=(\boldsymbol{A},\boldsymbol{t},\boldsymbol{s}_1,\boldsymbol{s}_2)$.

2）签名算法. 对消息μ进行签名，首先均匀随机选取$\boldsymbol{y}\leftarrow_{\$}S_{\gamma-1}^l$并计算$\boldsymbol{w}:=\boldsymbol{A}\boldsymbol{y}$，使用$Decompose_q(\cdot)$算法得到$\boldsymbol{w}$的高位比特$\boldsymbol{w}_{\mathrm{H}}$和低位比特$\boldsymbol{w}_{\mathrm{L}}$. 使用 Hash 函数$H_0$计算挑战值$c\in C$，从而得到$\boldsymbol{z}:=\boldsymbol{y}+c\boldsymbol{s}_1$以及$\boldsymbol{u}:=\boldsymbol{w}-c\boldsymbol{s}_2$，在通过拒绝抽样后输出合法的签名$\sigma=(z,c)$，其中$\beta$和$\beta'$满足$\|c\boldsymbol{s}_1\|_\infty\leqslant\beta,\|c\boldsymbol{s}_2\|_\infty\leqslant\beta'$.

算法 3. Aigis-sig-签名算法.

输入：消息μ，私钥$sk=(\boldsymbol{A},\boldsymbol{t},\boldsymbol{s}_1,\boldsymbol{s}_2)$；

输出：签名$\sigma=(z,c)$.

① 初始时设置$\boldsymbol{z}:=\perp$；

② while $\boldsymbol{z}=\perp$ do

③ $\quad\boldsymbol{y}\leftarrow_{\$}S_{\gamma-1}^l$；

④ $\quad\boldsymbol{w}:=\boldsymbol{A}\boldsymbol{y}$；

⑤ $\quad\boldsymbol{w}_{\mathrm{H}}=HighBits_q(\boldsymbol{w},2\gamma')$；

⑥ $\quad c:=H_0(\mu\|\boldsymbol{w}_{\mathrm{H}})\in C$；

⑦ $\quad\boldsymbol{z}:=\boldsymbol{y}+c\boldsymbol{s}_1$，$\boldsymbol{u}:=\boldsymbol{w}-c\boldsymbol{s}_2$；

⑧ $\quad(\boldsymbol{u}_{\mathrm{H}},\boldsymbol{u}_{\mathrm{L}})=Decompose_q(\boldsymbol{u},2\gamma')$；

⑨ $\quad$if $\|\boldsymbol{z}\|_\infty\geqslant\gamma-\beta$ 或 $\|\boldsymbol{u}_{\mathrm{L}}\|_\infty\geqslant\gamma'-\beta'$ 或

$\qquad\boldsymbol{u}_{\mathrm{H}}\neq\boldsymbol{w}_{\mathrm{H}}$ then

⑩ $\qquad\boldsymbol{z}:=\perp$；

⑪ $\quad$end if

⑫ end while

⑬ return $\sigma=(z,c)$.

3）验证算法. 在得到μ和$\sigma=(z,c)$后，首先利用$Decompose_q(\cdot)$算法计算$\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t}$的高位比特，然后根据$z$的范围和挑战值$c$的正确性判断签名合法性，这里的$\beta$（以及未显式表达的$\beta'$）与算法 2 中的$\beta$不同.

算法 4. Aigis-sig-验证算法.

输入：消息μ以及对应的签名$\sigma=(z,c)$，公钥$pk=(\boldsymbol{A},\boldsymbol{t})$;

输出：1（接受签名）/ 0（拒绝签名）.

① $\boldsymbol{w}'_{\mathrm{H}}=HighBits_q(\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t},2\gamma')$;

② if $\|z\|_\infty<\gamma-\beta$ 且 $c=H_0(\mu\|\boldsymbol{w}'_{\mathrm{H}})$ then

③ return 1;

④ else return 0;

⑤ end if

更详细的 Aigis-sig 签名方案的正确性以及安全性分析参见文献 [16, 43].

2 两方协同签名方案

本文基于 AMSIS/AMLWE 困难问题提出的两方协同签名方案有 2 个.

1）密钥生成算法. 和 Dilithium 算法一样，用户$P_i(i=1,2)$在选取随机的矩阵$\boldsymbol{A}_i$时，可以直接使用 256 b 的种子.

① P_1随机选取$\boldsymbol{A}_1\leftarrow_{\$}R_q^{k\times l}$并计算得到 Hash 值$g_1=H_1(\boldsymbol{A}_1)$，将$g_1$发送给$P_2$，$P_2$进行同样的操作，将$g_2=H_1(\boldsymbol{A}_2)$发送给$P_1$.

② P_1收到P_2发送的g_2后，将$\boldsymbol{A}_1$发送给P_2，P_2进行同样的操作，将$\boldsymbol{A}_2$发送给P_1.

③ P_1收到P_2发送的$\boldsymbol{A}_2$后，验证$H_1(\boldsymbol{A}_2)=g_2$，若不成立则中止，否则计算公共矩阵$\overline{\boldsymbol{A}}:=[\boldsymbol{A}\|\boldsymbol{I}_k]\in R_q^{k\times(l+k)}$，其中$\boldsymbol{A}=\boldsymbol{A}_1+\boldsymbol{A}_2$.

④ P_1随机选取$sk_{1,1}\leftarrow_{\$}S_\eta^l$，$sk_{1,2}\leftarrow_{\$}S_{\eta'}^k$，并计算得到$\boldsymbol{t}_1=\overline{\boldsymbol{A}}\cdot sk_1=\boldsymbol{A}\cdot sk_{1,1}+sk_{1,2}\in R_q^k$以及 Hash 值$g'_1=H_2(\boldsymbol{t}_1)$，将$g'_1$发送给$P_2$，$P_2$也进行同样的操作，将$g'_2=H_2(\boldsymbol{t}_2)$发送给$P_1$.

⑤ P_1收到P_2发送的g'_2后，将$\boldsymbol{t}_1$发送给P_2，P_2也进行同样的操作，将$\boldsymbol{t}_2$发送给P_1.

⑥ P_1收到P_2发送的$\boldsymbol{t}_2$后，验证$H_2(\boldsymbol{t}_2)=g'_2$，若不成立则中止，否则计算$\boldsymbol{t}=\boldsymbol{t}_1+\boldsymbol{t}_2$.

如果上述过程都没有中止，则可以得到P_i的私钥为$sk_i=(\boldsymbol{A},\boldsymbol{t}_i,sk_{i,1},sk_{i,2})$，公钥为$pk=(\boldsymbol{A},\boldsymbol{t})$.

注：在交换$\boldsymbol{A}_i(\boldsymbol{t}_i)$之前先交换 Hash 值$g_i(g'_i)$，是为了防止恶意的$P_2$在收到诚实的$P_1$发送的$\boldsymbol{A}_1(\boldsymbol{t}_1)$后，再自适应地选择$\boldsymbol{A}_2(\boldsymbol{t}_2)$.

2）签名算法. 我们以用户P_1的视角为例介绍两方协同签名协议，对消息$\mu\in\mathcal{M}$进行签名的具体过程如图 1 所示，运行协议后得到z_1. 同样地，P_2也会得到z_2，将两者合起来即得到签名$\sigma=(z,c,\boldsymbol{h})$.

和 Dilithium 一样，签名算法的第 1 步是计算身份协议所需要用到的挑战值$c\in C=B_\tau$（B_τ的定义与文献 [13,16] 相同，表示环R中恰好有τ个系数为±1且其余系数为0的元素组成的集合），该值应该由双方一起生成，因而需要双方交换多个消息，但是并不能直接交换$\boldsymbol{w}_1$和$\boldsymbol{w}_2$（或其对应的高位比特），主要原因有 2 个方面：

① 如果P_1将$\boldsymbol{w}_1$发送给了P_2，而之后未通过拒绝抽样，签名过程中止，$(\boldsymbol{w}_1,z_1)$可能会泄露关于私钥$sk_{1,1}$的信息. 之前的工作[26]的解决方法是利用非标准的格困难假设——Rejected MLWE 假设，然而这只是一个启发式假设，可能存在安全性问题.

② 如果P_2知道了$(\boldsymbol{w}_1,z_1)$，可以从z_1中提取得到$c\cdot sk_{1,2}$从而得到P_1的私钥的一部分$sk_{1,2}$，P_1同理，这也可能会导致安全性上的威胁.

我们用同态承诺方案来解决这个问题. P_1和P_2均可以用公钥$pk=(\boldsymbol{A},\boldsymbol{t})$和消息$\mu$计算得到消息对应的承诺密钥$\boldsymbol{ck}\leftarrow H_{ck}(\mu,pk)$，先互相交换承诺值$\boldsymbol{com}_i=Commit_{\boldsymbol{ck}}(\boldsymbol{w}_{i,\mathrm{H}};\boldsymbol{r}_i)$，由于使用的是同态承诺，因此可以将不同用户的承诺值聚合在一起，并用来在签名生成阶段计算挑战值. 同时和密钥生成算法一样，在互相交换承诺值$\boldsymbol{com}_i$之前需要先交换承诺值对应的 Hash 值$H_3(\boldsymbol{com}_i)$，如果缺少了这一步，恶意的P_2可以在看到P_1的承诺值$\boldsymbol{com}_1$自适应地选择$\boldsymbol{com}_2'$.

在第 3 轮通信中，双方交换$(\boldsymbol{z}_i,\boldsymbol{r}_i)$. 并验证对方提供的$\boldsymbol{com}_i$是否确实是用同态承诺方案计算得到，验证通过后，计算得到完整的$\boldsymbol{z}=\boldsymbol{z}_1+\boldsymbol{z}_2$.

3）验证算法. 收到消息μ对应的签名$\sigma=(\boldsymbol{z},c,\boldsymbol{h})$以及用来计算承诺的$\boldsymbol{r}$，作如下验证.

1）验证$\|\boldsymbol{z}\|_\infty<2\gamma-2\beta$.

2）用公钥$pk=(\boldsymbol{A},\boldsymbol{t})$和消息$\mu$计算得到消息对应的承诺密钥$\boldsymbol{ck}\leftarrow H_{ck}(\mu,pk)$.

3）计算$\boldsymbol{w}:=\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t}$，从而得到高位比特$\boldsymbol{w}_{\mathrm{H}}:=HighBits_q(\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t},4\gamma')$和$\widetilde{\boldsymbol{w}}_{\mathrm{H}}:=\boldsymbol{w}_{\mathrm{H}}-\boldsymbol{h}\bmod\frac{q-1}{2\gamma'}$.

4）计算$\boldsymbol{com}:=Commit_{\boldsymbol{ck}}(\widetilde{\boldsymbol{w}}_{\mathrm{H}};\boldsymbol{r})$.

5）验证$c=H_0(\mu,\boldsymbol{com})$，若通过，返回 1（即接受签名），否则返回 0（即拒绝签名）.

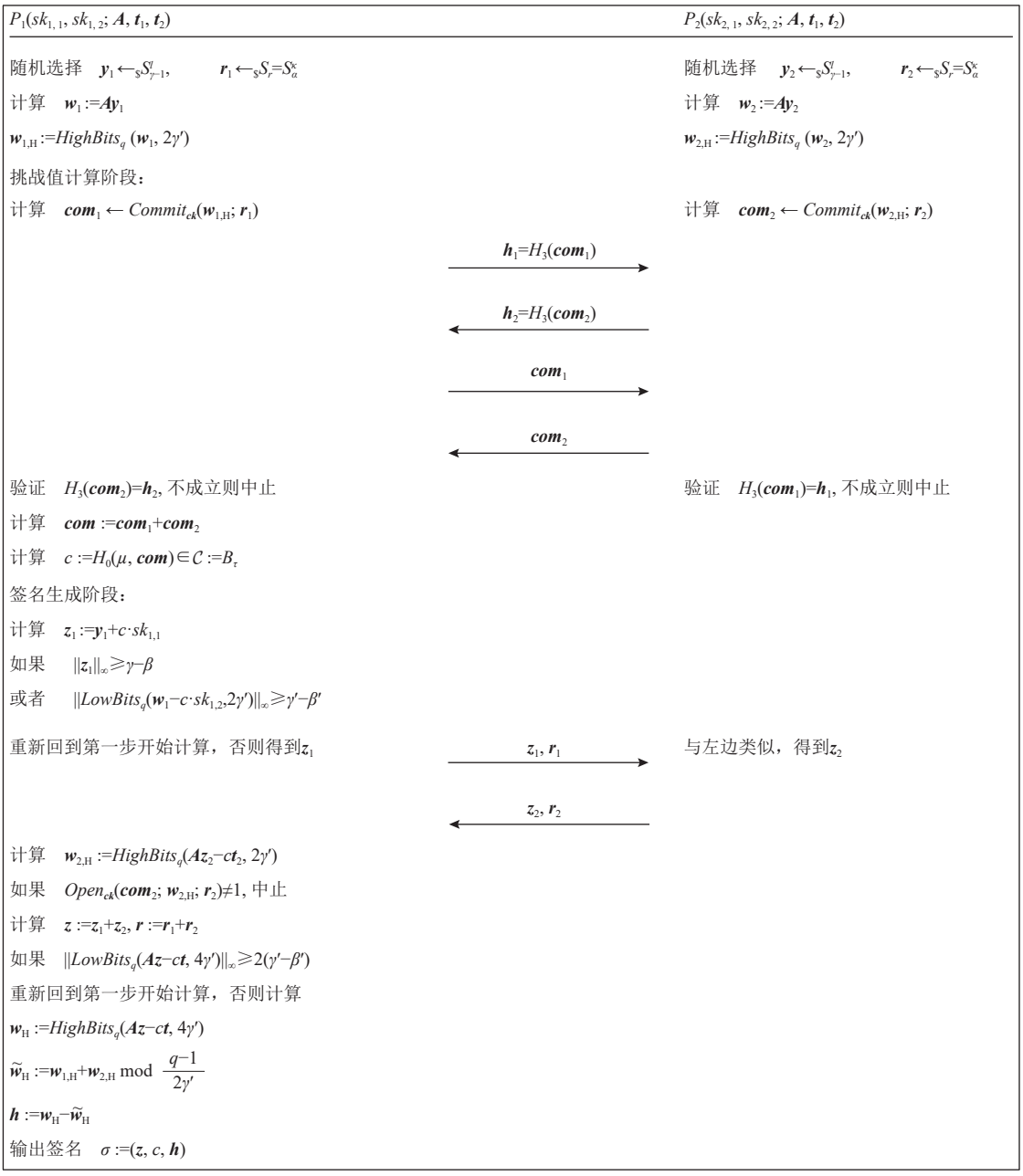

Fig. 1 Illustration of two-party signing

图 1 两方协同完成签名的过程

2.1 方案的正确性分析

证明本文提出方案的正确性，实际上就是证明 3 个结论：

1）$\|z\|_\infty < 2(\gamma - \beta)$

由签名算法的定义有$\|z_i\|_\infty \leqslant \gamma - \beta$，因此有$\|z\|_\infty = \|z_1 + z_2\|_\infty \leqslant \|z_1\|_\infty + \|z_2\|_\infty < 2(\gamma - \beta)$.

2）$\boldsymbol{w}_{\mathrm{H}} = HighBits_q(\boldsymbol{A}z - c\boldsymbol{t}, 4\gamma')$

由$\boldsymbol{A}z_i - c\boldsymbol{t}_i = \boldsymbol{A}(\boldsymbol{y}_i + c \cdot sk_{i,1}) - c(\boldsymbol{A} \cdot sk_{i,1} + sk_{i,2}) = \boldsymbol{A}\boldsymbol{y}_i - c \cdot sk_{i,2}$ 有 $\boldsymbol{A}z - c\boldsymbol{t} = \boldsymbol{A}(z_1 + z_2) - c(\boldsymbol{t}_1 + \boldsymbol{t}_2) = \boldsymbol{A}\boldsymbol{y}_1 + c \cdot sk_{1,2} + \boldsymbol{A}\boldsymbol{y}_2 + c \cdot sk_{2,2} - c(\boldsymbol{A} \cdot sk_{1,1} + sk_{1,2} + \boldsymbol{A} \cdot sk_{2,1} + sk_{2,2}) = \boldsymbol{A}\boldsymbol{y}_1 + \boldsymbol{A}\boldsymbol{y}_2 - (c \cdot sk_{1,2} + c \cdot sk_{2,2})$. 又因为$\|c \cdot sk_{1,2}\|_\infty < \beta'$以及$\|c \cdot sk_{2,2}\|_\infty < \beta'$ 有 $\|c \cdot sk_{1,2} + c \cdot sk_{2,2}\|_\infty \leqslant \|c \cdot sk_{1,2}\|_\infty + \|c \cdot sk_{2,2}\|_\infty < 2\beta'$.

同时根据由签名算法的定义得到$LowBits_q(\boldsymbol{A}z - c\boldsymbol{t}, 4\gamma') < 2\gamma' - 2\beta'$，由引理 1 有

$$HighBits_q(\boldsymbol{A}z - c\boldsymbol{t}, 4\gamma') = HighBits_q(\boldsymbol{A}\boldsymbol{y} - (c \cdot sk_{1,2} + c \cdot sk_{2,2}), 4\gamma') = HighBits_q(\boldsymbol{A}\boldsymbol{y}, 4\gamma') = HighBits_q(\boldsymbol{w}, 4\gamma') = \boldsymbol{w}_{\mathrm{H}}.$$

3）$Open_{ck}(\boldsymbol{com}, \widetilde{\boldsymbol{w}}_{\mathrm{H}}, \boldsymbol{r}) = 1$

由于$\boldsymbol{com}$是在签名算法中通过成功运行“挑战值

计算阶段”以及承诺方案的加法同态性质生成的，因此有

$$\begin{aligned}&\boldsymbol{com}=\boldsymbol{com}_1+\boldsymbol{com}_2=\\&Commit_{\boldsymbol{ck}}(\boldsymbol{w}_{1,\mathrm{H}};\boldsymbol{r}_1)+Commit_{\boldsymbol{ck}}(\boldsymbol{w}_{2,\mathrm{H}};\boldsymbol{r}_2)=\\&Commit_{\boldsymbol{ck}}(\boldsymbol{w}_{1,\mathrm{H}}+\boldsymbol{w}_{2,\mathrm{H}};\boldsymbol{r}_1+\boldsymbol{r}_2)\end{aligned}$$

是$\boldsymbol{w}_{1,\mathrm{H}}+\boldsymbol{w}_{2,\mathrm{H}}$对应的一个承诺值，即为$\widetilde{\boldsymbol{w}}_{\mathrm{H}}$对应的一个承诺.

验证方计算得到$\boldsymbol{w}_{\mathrm{H}}=HighBits_q(\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t},4\gamma')$和签名算法是相同的，再利用$\boldsymbol{h}=\boldsymbol{w}_{\mathrm{H}}-\widetilde{\boldsymbol{w}}_{\mathrm{H}}$进行“纠正”得到$\widetilde{\boldsymbol{w}}_{\mathrm{H}}$，因此有$Open_{\boldsymbol{ck}}(\boldsymbol{com},\widetilde{\boldsymbol{w}}_{\mathrm{H}},\boldsymbol{r})=1$.

2.2 方案的安全性分析

本节给出两方协同签名的安全性定义以及本文所提方案的安全性证明. 和文献 [21] 一样，我们沿用 Lindell 给出的两方协同签名的基于游戏的安全性定义 [5]，在该定义中，假设敌手只能调用一次密钥生成算法，而多个签名会话可以同时执行.

我们证明本文提出的两方协同签名方案具有分布式签名的选择消息攻击下的存在性不可伪造性（distributed signature with existential unforgeability under chosen message attack, DS-EU-CMA），需要用到如下的实验$Exp^{\mathrm{DS-EU-CMA}}(\mathcal{A})$：

初始时定义签名消息集合$\mathcal{M}$为空集，敌手可以询问协同签名的谕言机O^{DS}，得到若干个消息签名对(μ_i,σ_i)，并将μ_i添加到$\mathcal{M}$中，最后敌手需要产生一个新的(μ^*,σ^*)，其中$\mu^*\neq\mu_i$.

两方协同签名方案 DS-EU-CMA 安全意味着对于任意概率多项式时间的敌手$\mathcal{A}$，成功伪造签名的优势$Adv^{\mathrm{DS-EU-CMA}}(\mathcal{A})$是可忽略的，其中优势的定义为

$$Adv^{\mathrm{DS-EU-CMA}}(\mathcal{A}):=Pr[Exp^{\mathrm{DS-EU-CMA}}(\mathcal{A})\rightarrow 1].$$

我们的安全证明的主要思想与文献 [21, 29] 类似，我们证明了：如果敌手能以不可忽略的概率成功进行一个有效的伪造，则可以攻破承诺方案的计算绑定性或者 AMLWE/AMSIS 假设，即能以概率$\varepsilon_{\mathrm{Binding}}$攻破承诺方案的计算绑定性，或能以概率$\varepsilon_{\mathrm{D\text{-}AMLWE}_{q,k,l,\eta,\eta'}}$解决参数为$(q,k,l,\eta,\eta')$的判定性 AMLWE 问题，或能以概率$\varepsilon_{\mathrm{AMSIS}_{q,k,l+1,\delta,\delta'}}$解决参数为$(q,k,l+1,\delta,\delta')$的 AMSIS 问题，其中$\varepsilon$均表示不可忽略的概率值. 证明过程中需要用到文献 [44] 提出的分叉引理，在此先进行介绍.

引理 2. 分叉引理（general forking lemma）[44]. 对于固定的询问次数$Q\geqslant 1$以及集合$\mathcal{H}$（$\mathcal{H}$中的元素个数$|\mathcal{H}|\geqslant 2$），令$\mathcal{B}$是一个随机化算法，满足$(i,\sigma_{\mathrm{out}})\leftarrow\mathcal{B}(x,h_1,h_2,\cdots,h_Q)$，其输入$h_1,h_2,\cdots,h_Q\in\mathcal{H}$，$x$是由一个随机化的输入值生成算法$\mathcal{IG}$产生，输出$i\in\{0,1,\cdots,Q\}$，$\sigma_{\mathrm{out}}$是一个辅助输出.

$\mathcal{F}_{\mathcal{B}}(x)$是与$\mathcal{B}$相关联的分叉算法，定义为：

1）$\mathcal{B}$选取ρ作为随机的硬币（coins），以随机化.

2）随机选取$h_1,h_2,\cdots,h_Q\leftarrow_{\$}\mathcal{H}$.

3）运行算法$\mathcal{B}$：$(i,\sigma_{\mathrm{out}})\leftarrow\mathcal{B}(x,h_1,h_2,\cdots,h_Q;\rho)$.

4）如果$i=0$，则返回$(0,\perp,\perp)$，否则有$1\leqslant i\leqslant Q$重新随机选取$h'_i,h'_{i+1},\cdots,h'_Q\leftarrow_{\$}\mathcal{H}$.

5）$(i',\sigma'_{\mathrm{out}})\leftarrow\mathcal{B}\left(x,h_1,h_2,\cdots,h_{i-1},h'_i,h'_{i+1},\cdots,h'_Q;\rho\right)$.

6）如果$i=i'$且$h_i\neq h'_i$，返回$(1,\sigma_{\mathrm{out}},\sigma'_{\mathrm{out}})$，否则返回$(0,\perp,\perp)$.

定义$\mathcal{B}$的接受概率acc和分叉概率frk.

$$\begin{aligned}acc:=&Pr[i\neq 0:x\leftarrow\mathcal{IG},h_1,h_2,\cdots,h_Q\leftarrow\mathcal{H},(i,\sigma_{\mathrm{out}})\leftarrow\\&\mathcal{B}(x,h_1,h_2,\cdots,h_Q)],\\frk:=&Pr[b=1:x\leftarrow\mathcal{IG},(b,\sigma_{\mathrm{out}},\sigma'_{\mathrm{out}})\leftarrow\mathcal{F}_{\mathcal{B}}(x)],\end{aligned}$$

则有$frk\geqslant acc\cdot\left(\frac{acc}{Q}-\frac{1}{|\mathcal{H}|}\right)$，也可以表示为$acc\leqslant\frac{Q}{|\mathcal{H}|}+\sqrt{Q\cdot frk}$.

在本方案中，定义$\mathcal{IG}$的输出为$(\boldsymbol{A},\boldsymbol{t},\boldsymbol{ck}^*)$.

定理 1. 假设承诺方案具有计算隐藏性、计算绑定性、密钥均匀性、加法同态性和ξ-bits 最小熵. 那么对于任意的可以询问 1 次密钥生成随机谕言机，Q_{S}次签名谕言机和Q_{H}次随机谕言机$H_0,H_1,H_2,H_3,H_{\boldsymbol{ck}}$的概率多项式时间敌手，该两方协同签名方案是 DS-EU-CMA 安全的，其安全性依赖于参数为(q,k,l,η,η')的 AMLWE 假设以及参数为$(q,k,l+1,\delta,\delta')$的 AMSIS 假设.

证明. 对于一个能攻破该两方协同签名方案的敌手$\mathcal{A}$，我们首先构造一个关于$\mathcal{A}$的模拟器$\mathcal{S}$，能在不使用诚实用户的密钥的前提下模拟协议中诚实用户P_n的行为. $\mathcal{S}$用到了$SimKeyGen(\cdot)$及$SimSign(\cdot)$谕言机[21, 29]. 我们通过一系列的中间游戏$Game_i$来定义$\mathcal{S}$，并比较了每一个游戏和前一个游戏的差别. 在得到$\mathcal{S}$后，我们利用$\mathcal{S}$构造了算法$\mathcal{B}$，并通过调用分叉引理获得针对 2 个不同的挑战值的伪造签名，这使得我们能攻破承诺方案的计算绑定性或者 AMLWE/AMSIS 假设.

$Game_0$：分为随机谕言机模拟（random oracle simulation）、诚实用户谕言机模拟和伪造 3 个部分.

1）随机谕言机模拟. 本方案中用到 5 个 Hash 函数$H_0,H_1,H_2,H_3,H_{\boldsymbol{ck}}$，分别模拟：

① $H_1:\{0,1\}^*\rightarrow\{0,1\}^{l_1}$（用在密钥生成算法中的公钥矩阵生成）. 构造空的 Hash 列表HT_1，当向HT_1询问x'时，若列表中已经有$HT_1(x')$则返回$HT_1(x')$，否则从$\{0,1\}^{l_1}$中均匀随机地选取一个值y'当作$HT_1(x')$，并将

(x', y')填入HT_1.

② $H_2: \{0,1\}^* \to \{0,1\}^{l_2}$(用在密钥生成算法中的私钥和$\boldsymbol{t}$的生成). 构造空的 Hash 列表$HT_2$, 当向$HT_2$询问$x'$时, 若列表中已经有$HT_2(x')$则返回$HT_2(x')$, 否则从$\{0,1\}^{l_2}$中均匀随机地选取一个值$y'$当作$HT_2(x')$, 并将$(x', y')$填入$HT_2$.

③ $H_3: \{0,1\}^* \to \{0,1\}^{l_3}$(用在签名算法中的交换承诺值). 构造空的 Hash 列表HT_3, 当向HT_3询问x'时, 若列表中已经有$HT_3(x')$则返回$HT_3(x')$, 否则从$\{0,1\}^{l_3}$中均匀随机地选取一个值y'当作$HT_3(x')$, 并将(x', y')填入HT_3.

④ $H_0: \{0,1\}^* \to C$(用在签名算法中的计算挑战值). 构造空的 Hash 列表HT_0, 并令$ctr = 0$. 注意到H_0的输入为$(\mu, \boldsymbol{com})$, 因此需要先询问随机谕言机$H_3(\mu, pk)$, 若列表中已经有$HT_0(\mu, \boldsymbol{com})$则返回$HT_0(\mu, \boldsymbol{com})$, 否则设$ctr = ctr + 1$, 并将$h_{ctr}$当作$HT_0(\mu, \boldsymbol{com})$, 将$(\mu, \boldsymbol{com}, h_{ctr})$填入$HT_0$. 其中, $\{h_1, h_2, \cdots, h_{Q_S+Q_H+1}\}$是$\mathcal{S}$收到的作为输入的随机挑战值.

⑤ $H_{ck}: \{0,1\}^* \to \mathcal{S}_{ck}$(用在计算承诺密钥). 构造空的 Hash 列表$H_{ck}$, 由于$H_{ck}$的输入为$(\mu, pk)$, 故向$HT_{ck}$询问$(\mu, pk)$, 若列表中已经有$HT_{ck}(\mu, pk)$, 则返回$HT_{ck}(\mu, pk)$, 否则从承诺密钥空间$\mathcal{S}_{ck}$中均匀随机地选取一个值$\boldsymbol{y}$当作$HT_{ck}(\mu, pk)$, 并将$(\mu, pk, \boldsymbol{y})$填入$HT_{ck}$.

搜索 Hash 列表算法$SearchHash(HT, \boldsymbol{h})$. 在构造 Hash 列表$HT$后, 可以定义算法$SearchHash(HT, \boldsymbol{h})$: 对于$\boldsymbol{h}$, 在列表$HT$中寻找原像$\widetilde{m}$满足$HT(\widetilde{m}) = \boldsymbol{h}$, 如果不存在原像, 则返回$flag$ = alert并设置$m = \perp$, 如果存在不止一个原像, 则返回$flag$ = bad.

2) 诚实用户谕言机模拟(honest party oracle simulation). 在这个游戏中, 敌手$\mathcal{A}$的行为和两方协同签名中的一个诚实用户相同. 与文献[21, 29]类似, 当$\mathcal{A}$询问诚实用户谕言机时, $\mathcal{S}$可以直接按照密钥生成算法和签名算法的定义进行模拟, 限于篇幅, 在此处不展开, 感兴趣的读者可以参见文献[21, 29].

3) 伪造(forgery). 敌手$\mathcal{A}$输出一个伪造的消息签名对$(\mu^*, \sigma^* = (\boldsymbol{z}^*, c^*, \boldsymbol{h}^*))$. 对于给定的伪造, $\mathcal{S}$操作为:

如果$\mu^* \notin \mathcal{M}$, 返回$(0, \perp)$; 并计算承诺密钥$\boldsymbol{ck}^* \leftarrow H_{ck}(\mu^*, pk)$和挑战值$c^* \leftarrow H_0(\mu^*, \boldsymbol{com}^*)$以及$\widetilde{\boldsymbol{w}}_H = HighBits_q\left(\boldsymbol{A}\boldsymbol{z}^* - c^*\boldsymbol{t}, 4\gamma'\right) - \boldsymbol{h}^* \bmod \frac{q-1}{4\gamma'}$.

如果承诺的打开值$Open_{ck}(\boldsymbol{com}^*; \widetilde{\boldsymbol{w}}_H; \boldsymbol{r}^*) \neq 1$或者$\|z^*\|_\infty \geqslant 2(\gamma - \beta)$, 则返回$(0, \perp)$; 否则可以找到下标$i_f \in \{1, 2, \cdots, Q_S + Q_H + 1\}$使得$c^* = h_{i_f}$, 返回

$$(i_f, \sigma_{out}) = (\boldsymbol{z}^*, c^*, \boldsymbol{h}^*, \boldsymbol{r}^*, \boldsymbol{com}^*, \mu^*, \boldsymbol{ck}^*).$$

将$\mathcal{S}$在第i个游戏中不输出$(0, \perp)$的概率记作$Pr[Game_i]$, 有$Pr[Game_0] := Adv^{DS-EU-CMA}(\mathcal{A})$.

$Game_1$: 相较于$Game_0$, $Game_1$仅改变$\mathcal{S}$的签名过程, 主要的变化在于挑战值c改为从集合C中均匀随机选取, 因此$\mathcal{S}$能在不与敌手$\mathcal{A}$交互的情况下计算出$\boldsymbol{z}_n$. 在收到挑战值h_i后, $\mathcal{S}$运行搜索 Hash 列表算法$SearchHash(HT_3, h_i)$, 找到原像$\boldsymbol{com}_i$并计算$\boldsymbol{com} = \boldsymbol{com}_n + \boldsymbol{com}_i$, 最后, $\mathcal{S}$用c当作对随机谕言机H_0的询问$(\mu, \boldsymbol{com})$的回答. 除了 3 种情况之外, $Game_1$和$Game_0$是等同的.

1) 在$\mathcal{S}$或$\mathcal{A}$对H_3的至多$Q_H + 2Q_S$次询问中, 产生了至少一次碰撞, 发生的概率为$\frac{(Q_H + 2Q_S + 1)^2}{2^{l_3+1}}$.

2) 在对H_0的Q_S次询问中, 至少失败了 1 次, 分 2 种情况:

① $H_3(\boldsymbol{com}_n)$在$\mathcal{A}$对H_3的$Q_H + 2Q_S$次询问中已经被询问过, 这意味着$\mathcal{A}$已经猜到了$Commit_{ck}$的输出, 发生概率为$\frac{Q_S \cdot (Q_H + 2Q_S)}{2^\xi}$.

② Hash 列表HT_0中$(\mu, \boldsymbol{com})$对应的 Hash 值已经被$\mathcal{S}$或$\mathcal{A}$在之前的对H_0的至多$Q_H + Q_S$次询问时设置过, 发生概率为$\frac{Q_S \cdot (Q_H + Q_S)}{2^\xi}$.

3) $\mathcal{A}$在没有询问H_3时, 就已经猜到了H_3的 2 个输出中的至少 1 个, 发生概率为$\frac{2Q_S}{2^{l_3}}$.

由此可得

$$|Pr[Game_1] - Pr[Game_0]| \leqslant \frac{(Q_H + 2Q_S + 1)^2}{2^{l_3+1}} + Q_S\left(\frac{2Q_H + 3Q_S}{2\xi} + \frac{2}{2^{l_3}}\right).$$

$Game_2$: 相对于$Game_1$, $Game_2$继续改变$\mathcal{S}$的签名过程, 改变方式取决于$\|z_1\|_\infty$的大小: 如果$\|z_1\|_\infty \geqslant \gamma - \beta$, 则均匀随机地选取$\boldsymbol{w}_n \leftarrow_\$ R_q^k$; 否则令$\boldsymbol{w}_n = \boldsymbol{A}\boldsymbol{y}_n$, 其中$\boldsymbol{y} \leftarrow_\$ S_{\gamma-1}^l$.

和之前的游戏一样, $\mathcal{S}$将$\boldsymbol{w}_{n,H}$进行承诺得到$\boldsymbol{com}_n = Commit_{ck}(\boldsymbol{w}_{n,H}, \boldsymbol{r}_n)$, 其中$\boldsymbol{r}_n \leftarrow_\$ \mathcal{S}_r$, 并将$H_3(\boldsymbol{com}_n)$发送给敌手$\mathcal{A}$.

$Game_2$和$Game_1$之间的区别是$\mathcal{A}$在进行Q_S次询问后, 区分一个模拟的承诺方案和一个真实的承诺方案的优势不同, 即攻破承诺方案的隐藏性的优势不同. 因此有$|Pr[Game_2] - Pr[Game_1]| \leqslant Q_S \cdot \varepsilon_{Hiding}$.

$Game_3$: 在这个游戏中, $\mathcal{S}$并不直接生成$\boldsymbol{z}_n$, 也不第一个拒绝抽样$\|z_1\|_\infty < \gamma - \beta$. 相反, $\mathcal{S}$模拟拒绝抽样: 以$1 - \frac{|S_{\gamma-\beta-1}^k|}{|S_{\gamma-1}^l|}$的概率随机选取$\boldsymbol{w}_n \leftarrow_\$ R_q^k$. 以$\frac{|S_{\gamma-\beta-1}^k|}{|S_{\gamma-1}^l|}$的概率随机选取$\boldsymbol{z}_n \leftarrow_\$ S_{\gamma-\beta-1}^l$并定义$\boldsymbol{w}_n = \boldsymbol{A}\boldsymbol{z}_n - c \cdot (\boldsymbol{t}_n - sk_{n,2})$.

$Game_3$和$Game_2$之间的区别是$\mathcal{A}$区分一个随机选取的签名和一个真实的签名之间的优势，类似于文献[29]，我们可以得到这2个$Game$中生成$\boldsymbol{w}_n$的方式是相同的，即有$Pr[Game_3]=Pr[Game_2]$.

$Game_4$：在这个游戏中，我们的目的是完全移除签名过程中的私钥. 在没有拒绝抽样时（概率为$\frac{|S^k_{\gamma-\beta-1}|}{|S^l_{\gamma-1}|}$），$\boldsymbol{w}'_n=\boldsymbol{A}\boldsymbol{z}_n-c\boldsymbol{t}_n$因$\|c\cdot sk_{n,2}\|<\beta$且$\|LowBits_q(\boldsymbol{A}z_n-c\boldsymbol{t}_n,2\gamma')\|_\infty<\gamma'-\beta$，由引理1可知$Game_3$中的$\boldsymbol{A}\boldsymbol{z}_n-c(\boldsymbol{t}_n-sk_{n,2})$和$Game_4$中的$\boldsymbol{A}\boldsymbol{z}_n-c\boldsymbol{t}_n$在模为$2\gamma'$时具有相同的高位比特，即$Game_4$中$\boldsymbol{w}_n$的高位比特和$Game_3$中$\boldsymbol{w}_n$的高位比特的分布相同，因此有$Pr[Game_4]=Pr[Game_3]$.

接下来修改密钥产生过程：

$Game_5$：在这个游戏中，开始修改密钥产生过程. 模拟器$\mathcal{S}$拥有$\boldsymbol{A}\leftarrow_\$R_q^{k\times l}$后，随机选取$hk_n\leftarrow_\$\{0,1\}^{l_1}$并将其发送给敌手$\mathcal{A}$，在收到$hk_i$后，$\mathcal{S}$运行$SearchHash(HT_1,hk_i)$找到原像$\boldsymbol{A}_i$并计算$\boldsymbol{A}_n=\boldsymbol{A}-\boldsymbol{A}_i$，以$hk_n$作为$H_1(\boldsymbol{A}_n)$的回答. $Game_5$和$Game_4$之间的公共矩阵$\boldsymbol{A}$并没有发生变化. 除了3种情况之外，$Game_5$和$Game_4$是等同的：

1）在$\mathcal{S}$或$\mathcal{A}$对H_1的至多Q_H次询问中，产生了至少1次碰撞，发生概率为$\frac{(Q_\mathrm{H}+1)Q_\mathrm{H}}{2^{l_1+1}}$.

2）$\mathcal{S}$以hk_n作为$H_1(\boldsymbol{A}_n)$的回答失败. 若在之前$\mathcal{A}$对随机谕言机H_1的Q_H次询问中，$H_1(\boldsymbol{A}_n)$已经被$\mathcal{A}$询问过，则会出现这种情况，发生概率为$\frac{Q_\mathrm{H}}{q^{d\cdot k\cdot l}}$.

3）$\mathcal{A}$在没有询问H_1时，就已经猜到了H_1的2个输出中的至少1个，发生概率为$\frac{2Q_\mathrm{S}}{2^{l_1}}$. 由此可得

$$|Pr[Game_5]-Pr[Game_4]|\leqslant\frac{(Q_\mathrm{H}+1)Q_\mathrm{H}}{2^{l_1+1}}+\frac{Q_\mathrm{H}}{q^{d\cdot k\cdot l}}+\frac{2}{2^{l_1}}.$$

$Game_6$：在这个游戏中，继续修改密钥生成过程，$\mathcal{S}$产生$\boldsymbol{t}_n\leftarrow_\$R_q^k$来代替$\boldsymbol{t}_n=\boldsymbol{A}\cdot sk_{n,1}+sk_{n,2}$，如果敌手$\mathcal{A}$可以区分$Game_5$和$Game_6$，根据定义2，$\mathcal{A}$可以攻破参数为$(q,k,l,\eta,\eta')$的Decisional-AMLWE问题，因此有$|Pr[Game_6]-Pr[Game_5]|\leqslant\varepsilon_{\mathrm{D-AMLWE}_{q,k,l,\eta,\eta'}}$.

$Game_7$：在这个游戏中，$\mathcal{S}$以公钥$\boldsymbol{t}\in R_q^k$作为输入，随机选取$comk_n\leftarrow_\$\{0,1\}^{l_2}$并将其发送给$\mathcal{A}$. 在收到$\mathcal{A}$返回的$comk_i$后，$\mathcal{S}$运行$SearchHash(HT_2,comk_i)$找到原像$\boldsymbol{t}_i$并计算$\boldsymbol{t}_n=\boldsymbol{t}-\boldsymbol{t}_i$.

最后，$\mathcal{S}$以$comk_n$作为$H_2(\boldsymbol{t}_n)$的回答. $Game_7$和$Game_6$之间$\boldsymbol{t}$和$\boldsymbol{t}_n$并没有发生变化. 除了3种情况之外，$Game_7$和$Game_6$是等同的：

1）在$\mathcal{S}$或$\mathcal{A}$对H_2的至多Q_H次询问中，产生了至少1次碰撞，发生概率为$\frac{(Q_\mathrm{H}+1)Q_\mathrm{H}}{2^{l_2+1}}$.

2）$\mathcal{S}$以$comk_n$作为$H_2(\boldsymbol{t}_n)$的回答失败. 若在之前$\mathcal{A}$对随机谕言机H_2的Q_H次询问中，$H_2(\boldsymbol{t}_n)$已经被$\mathcal{A}$询问过，则会出现这种情况，发生概率为$\frac{Q_\mathrm{H}}{q^{d\cdot k}}$.

3）$\mathcal{A}$在没有询问H_2之前，就已经猜到了H_2的2个输出中的至少1个，发生的概率为$\frac{2}{2^{l_2}}$. 由此可得

$$|Pr[Game_7]-Pr[Game_6]|\leqslant\frac{(Q_\mathrm{H}+1)Q_\mathrm{H}}{2^{l_2+1}}+\frac{Q_\mathrm{H}}{q^{d\cdot k}}+\frac{2}{2^{l_2}}.$$

$Game_8$：在这个游戏中，将AMSIS问题实例嵌入到证明中，AMSIS实例具有形式$[\boldsymbol{A}'\|\boldsymbol{I}_k]$，其中$\boldsymbol{A}'\leftarrow_\$R_q^{k\times(l+1)}$. Game_7中的公钥$(\boldsymbol{A},\boldsymbol{t})$是从$R_q^{k\times l}\times R_q^k$中均匀随机选取的，因此令$\boldsymbol{A}'=[\boldsymbol{A}\|\boldsymbol{t}]$并不影响公钥$(\boldsymbol{A},\boldsymbol{t})$的分布.

接下来嵌入承诺密钥$\boldsymbol{ck}^\#\leftarrow H_{\boldsymbol{ck}}(\mu,pk)$，和文献[21,29]类似，由于该承诺方案具有密钥均匀性，因此可以直接从承诺密钥空间$\mathcal{S}_{\boldsymbol{ck}}$上均匀选取承诺密钥，这样模拟出的承诺密钥和诚实生成的承诺密钥是不可区分的. 假设给$\mathcal{S}$一个承诺密钥$\boldsymbol{ck}^\#$作为输入，我们希望$\mathcal{S}$对除了一个询问之外的所有询问全部返回的是模拟的承诺密钥. 在使用询问进行伪造签名时，$\mathcal{S}$应该使用一个预先定义好的承诺密钥$\boldsymbol{ck}^\#$，只需要调整模拟谕言机$H_{\boldsymbol{ck}}$的方式：当收到对$H_{\boldsymbol{ck}}$的询问时，以ω的概率均匀随机选取$HT_{\boldsymbol{ck}}(\mu,pk)\leftarrow_\$\mathcal{S}_{\boldsymbol{ck}}$，以$1-\omega$的概率选取$HT_{\boldsymbol{ck}}(\mu,pk)=\boldsymbol{ck}^\#$.

如果模拟成功（即$\mathcal{S}$不返回$(0,\perp)$），则有$\boldsymbol{ck}^\#=\boldsymbol{ck}=H_{\boldsymbol{ck}}(\mu^*,pk)$. 考虑到所做的修改，我们需要调整$\mathcal{S}$不返回$(0,\perp)$的概率，即有

$$Pr[Game_8]\geqslant\omega^{Q_\mathrm{H}+Q_\mathrm{S}}\cdot(1-\omega)\cdot Pr[Game_7].$$

综上，我们依赖模拟器$\mathcal{S}$构造了一个算法$\mathcal{B}$能攻破使用承诺密钥$\boldsymbol{ck}^\#$的承诺方案的绑定性，或者找到一个输入为$\boldsymbol{A}'=[\boldsymbol{A}\|\boldsymbol{t}]$的AMSIS问题的解. 由于算法$\mathcal{B}$输入$(\boldsymbol{A},\boldsymbol{t},\boldsymbol{ck}^\#)$并调用分叉引理中的分叉算法$\mathcal{F}$，可以以$frk$的概率得到2个伪造的输出$\boldsymbol{\sigma}_{\mathrm{out}}=(\boldsymbol{z}^*,c^*,\boldsymbol{h}^*,\boldsymbol{r}^*,\boldsymbol{com}^*,\mu^*,\boldsymbol{ck}^*)$和$\boldsymbol{\sigma}'_{\mathrm{out}}=(\boldsymbol{z}',c',\boldsymbol{h}',\boldsymbol{r}',\boldsymbol{com}',\mu',\boldsymbol{ck}')$，由分叉引理有

$$Pr[Game_8]=acc\leqslant\frac{Q_\mathrm{H}+Q_\mathrm{S}+1}{|C|}+\sqrt{(Q_\mathrm{H}+Q_\mathrm{S}+1)\cdot frk}.$$

根据分叉算法的定义，所有分叉前产生的值都是相同的，即有$\boldsymbol{com}^*=\boldsymbol{com}'$，$\mu^*=\mu'$以及$\boldsymbol{ck}^*=\boldsymbol{ck}'=\boldsymbol{ck}^\#$，且满足$c^*\neq c'$. 因此有

$$Open_{ck'}\left(\boldsymbol{com}^*,\widetilde{\boldsymbol{w}}^*_\mathrm{H},\boldsymbol{r}^*\right)=Open^*_{\boldsymbol{ck}}\left(\boldsymbol{com}^*,\widetilde{\boldsymbol{w}}'_\mathrm{H},\boldsymbol{r}'\right)=1.$$

其中

$$\widetilde{\boldsymbol{w}}^*_\mathrm{H}=HighBits_q\left(\boldsymbol{A}\boldsymbol{z}^*-c^*\boldsymbol{t},4\gamma'\right)-\boldsymbol{h}^*\bmod\frac{q-1}{4\gamma'},$$

$$\widetilde{\boldsymbol{w}}'_\mathrm{H}=HighBits_q(\boldsymbol{A}\boldsymbol{z}'-c'\boldsymbol{t},4\gamma')-\boldsymbol{h}'\bmod\frac{q-1}{4\gamma'}.$$

按照$\widetilde{\boldsymbol{w}}^*_\mathrm{H}$和$\widetilde{\boldsymbol{w}}'_\mathrm{H}$是否相等2种情况分类讨论：

情形 1. $\widetilde{\boldsymbol{w}}_{\mathrm{H}}^{*} \neq \widetilde{\boldsymbol{w}}_{\mathrm{H}}'$, 此时 $\mathcal{A}$ 已经找到了承诺 $\boldsymbol{com}^{*}$ 的同一个承诺密钥 $\boldsymbol{ck}^{*}$ 对应的 2 个有效的打开, 这意味着 $\mathcal{A}$ 已经攻破了承诺方案的绑定性, 发生的概率为 $\varepsilon_{\mathrm{Binding}}$.

情形 2. $\widetilde{\boldsymbol{w}}_{\mathrm{H}}^{*} = \widetilde{\boldsymbol{w}}_{\mathrm{H}}'$, 此时有 $HighBits_q\left(\boldsymbol{Az}^{*}-c^{*}\boldsymbol{t}, 4\gamma'\right)-\boldsymbol{h}^{*}=HighBits_q(\boldsymbol{Az}'-c'\boldsymbol{t}, 4\gamma')-\boldsymbol{h}'$, 将该式的两边同时乘以 $4\gamma'$ 并结合 $HighBits_q(\cdot)$ 和 $LowBits_q(\cdot)$ 的定义有

$$(\boldsymbol{Az}^{*}-c^{*}\boldsymbol{t})-LowBits_q\left(\boldsymbol{Az}^{*}-c^{*}\boldsymbol{t}, 4\gamma'\right)-4\gamma'\cdot\boldsymbol{h}^{*}=(\boldsymbol{Az}'-c'\boldsymbol{t})-LowBits_q(\boldsymbol{Az}'-c'\boldsymbol{t}, 4\gamma')-4\gamma'\cdot\boldsymbol{h}',$$

将 $\boldsymbol{x}^{*}=LowBits_q(\boldsymbol{Az}^{*}-c^{*}\boldsymbol{t}, 4\gamma')+4\gamma'\cdot\boldsymbol{h}^{*}$ 和 $\boldsymbol{x}'=LowBits_q(\boldsymbol{Az}'-c'\boldsymbol{t}, 4\gamma')+4\gamma'\cdot\boldsymbol{h}'$ 代入, 可得

$$\boldsymbol{Az}^{*}-c^{*}\boldsymbol{t}-\boldsymbol{x}^{*}=\boldsymbol{Az}'-c'\boldsymbol{t}-\boldsymbol{x}', \text{ 即}$$

$$[\boldsymbol{A}\|\boldsymbol{t}\|\boldsymbol{I}]\cdot\begin{bmatrix}\boldsymbol{z}^{*}-\boldsymbol{z}'\\ c'-c^{*}\\ \boldsymbol{x}'-\boldsymbol{x}^{*}\end{bmatrix}=\boldsymbol{0}.$$

由于伪造的签名是有效的, 有 $\|LowBits_q(\boldsymbol{Az}^{*}-c^{*}\boldsymbol{t}, 4\gamma')\|_{\infty}<2(\gamma'-\beta')$ 以及 $\|LowBits_q(\boldsymbol{Az}'-c'\boldsymbol{t}, 4\gamma')\|_{\infty}<2(\gamma'-\beta')$, 同时由于 $\boldsymbol{h}^{*}, \boldsymbol{h}'\in\{-1, 0, 1\}^{k}$, 因此有 $\|\boldsymbol{x}^{*}\|_{\infty}\leqslant\|LowBits_q\left(\boldsymbol{Az}^{*}-c^{*}\boldsymbol{t}, 4\gamma'\right)\|_{\infty}+4\gamma'\|\boldsymbol{h}^{*}\|_{\infty}\leqslant 6\gamma'-2\beta'$. 同理有 $\|\boldsymbol{x}'\|_{\infty}\leqslant 6\gamma'-2\beta'$, 可以得到

$$\|\boldsymbol{x}'-\boldsymbol{x}^{*}\|_{\infty}\leqslant 12\gamma'-4\beta'.$$

又由于 $\|\boldsymbol{z}^{*}\|_{\infty}<2(\gamma-\beta)$ 以及 $\|\boldsymbol{z}'\|_{\infty}<2(\gamma-\beta)$, 有

$$\|\boldsymbol{z}^{*}-\boldsymbol{z}'\|_{\infty}<4(\gamma-\beta).$$

故 $\mathcal{A}$ 找到了参数为 $(q, k, l+1, \delta, \delta')$ 的 AMSIS 问题的解, 其中 δ 取决于 $\|\boldsymbol{z}^{*}-\boldsymbol{z}'\|_{\infty}<4(\gamma-\beta)$, δ' 取决于 $\|\boldsymbol{x}'-\boldsymbol{x}^{*}\|_{\infty}\leqslant 12\gamma'-4\beta'$, 发生的概率为 $\varepsilon_{\mathrm{AMSIS}(q,k,l+1,\delta,\delta')}$. 由分叉引理可知

$$frk\leqslant\varepsilon_{\mathrm{Binding}}+\varepsilon_{\mathrm{AMSIS}(q,k,l+1,\delta,\delta')}.$$

综上, 我们得到敌手成功伪造签名的概率优势为

$$Adv^{\mathrm{DS-EU-CMA}}(\mathcal{A})=Pr[Game_0]\leqslant|Pr[Game_8]|+\sum_{i=0}^{8}|Pr[Game_{i+1}]-Pr[Game_i]|\leqslant\frac{(Q_{\mathrm{H}}+2Q_{\mathrm{S}}+1)^2}{2^{l_3+1}}+Q_{\mathrm{S}}\left(\frac{2Q_{\mathrm{H}}+3Q_{\mathrm{S}}}{2^{\xi}}+\frac{2}{2^{l_3}}\right)+Q_{\mathrm{S}}\cdot\varepsilon_{\mathrm{Hiding}}+\frac{(Q_{\mathrm{H}}+1)Q_{\mathrm{H}}}{2^{l_1+1}}+\frac{Q_{\mathrm{H}}}{q^{d\cdot k\cdot l}}+\frac{2}{2^{l_1}}+\varepsilon_{\text{D-AMLWE}_{q,k,l,\eta,\eta'}}+\frac{(Q_{\mathrm{H}}+1)Q_{\mathrm{H}}}{2^{l_2+1}}+\frac{Q_{\mathrm{H}}}{q^{d\cdot k}}+\frac{2}{2^{l_2}}+\frac{1}{\omega^{Q_{\mathrm{H}}+Q_{\mathrm{S}}}(1-\omega)}\cdot\frac{Q_{\mathrm{H}}+Q_{\mathrm{S}}+1}{|C|}+\frac{\sqrt{(Q_{\mathrm{H}}+Q_{\mathrm{S}}+1)\cdot(\varepsilon_{\mathrm{Binding}}+\varepsilon_{\mathrm{AMSIS}_{q,k,l+1,\delta,\delta'}})}}{\omega^{Q_{\mathrm{H}}+Q_{\mathrm{S}}}(1-\omega)}.$$

因此, 若敌手以不可忽略的概率成功伪造签名, 则能以不可忽略的概率攻破承诺方案的计算绑定性或 AMLWE/AMSIS 假设. 证毕.

3 性能分析与比较

为密码方案选择合适的参数需要在多个方面权衡, 本方案参数的选择应该在保证足够安全性的前提下, 使得期望重复次数(影响通信轮数和签名时间)较小且拥有较短的签名和密钥尺寸, 因此我们主要考虑这 3 项指标, 并以此评价方案的性能.

3.1 重复次数的期望值(通信重复轮数)

签名生成算法用到了拒绝抽样技术, 以得到 $\|\boldsymbol{z}_i\|_{\infty}\leqslant\gamma-\beta-1$, $\|LowBits_q(\boldsymbol{w}_i-c\cdot sk_{i,2}, 2\gamma')\|_{\infty}\leqslant\gamma'-\beta-1$ 以及 $\|LowBits_q(\boldsymbol{Az}-c\boldsymbol{t}, 4\gamma')\|_{\infty}\leqslant 2\gamma'-2\beta-1$, 需要考虑重复的轮数, 主要由 1)~3)决定:

1) $\|\boldsymbol{z}_i\|_{\infty}\leqslant\gamma-\beta-1$

如果 $\|c\cdot sk_{i,1}\|\leqslant\beta$ 成立, 那么当 $\|\boldsymbol{y}_i\|_{\infty}\leqslant\gamma-2\beta-1$ 时, 总有 $\|\boldsymbol{z}_i\|_{\infty}=\|\boldsymbol{y}_i+c\cdot sk_{i,1}\|_{\infty}\leqslant\gamma-\beta-1$, 该范围的大小为 $2(\gamma-\beta)-1$. 由于 $\boldsymbol{y}_i\leftarrow_{\$}S_{\gamma-1}^{l}$ 中的每个多项式系数都是从 $\{-(\gamma-1), \cdots, (\gamma-1)\}$ 中均匀随机选取(即对于固定的 $c\cdot sk_{i,1}$, $\boldsymbol{z}_i$ 共有 $2\gamma-1$ 种可能值), 因此 $\|\boldsymbol{z}_i\|_{\infty}\leqslant\gamma-\beta-1$ 的概率为

$$\left(\frac{2(\gamma-\beta)-1}{2\gamma-1}\right)^{n\cdot l}=\left(1-\frac{\beta}{\gamma-1/2}\right)^{n\cdot l}\approx\mathrm{e}^{-nl\beta/\gamma},$$

其中 "≈" 利用了 $\gamma\gg\frac{1}{2}$ 的事实.

2) $\|LowBits_q(\boldsymbol{w}_i-c\cdot sk_{i,2}, 2\gamma')\|_{\infty}\leqslant\gamma'-\beta'-1$

计算 $\|LowBits_q(\boldsymbol{w}_i-c\cdot sk_{i,2}, 2\gamma')\|_{\infty}\leqslant\gamma'-\beta'-1$ 成立的概率, 由于其中的多项式的每个系数都在模 $2\gamma'$ 的剩余系中均匀分布, 因此

$$\|LowBits_q(\boldsymbol{w}_i-c\cdot sk_{i,2}, 2\gamma')\|_{\infty}\leqslant\gamma'-\beta'-1$$

成立的概率为

$$\left(\frac{2(\gamma'-\beta')-1}{2\gamma'}\right)^{n\cdot k}=\left(1-\frac{\beta'+1/2}{\gamma'}\right)^{n\cdot k}\approx\mathrm{e}^{-nk\beta'/\gamma'},$$

其中 "≈" 利用了 $\gamma\gg\frac{1}{2}$ 的事实.

3) $\|LowBits_q(\boldsymbol{Az}-c\boldsymbol{t}, 4\gamma')\|_{\infty}\leqslant 2\gamma'-2\beta'-1$

不难验证满足 2), 则 3)一定成立. 换句话说, 对于满足 2)的 $\boldsymbol{w}_1-c\cdot sk_{1,2}$ 和 $\boldsymbol{w}_2-c\cdot sk_{2,2}$, 一定有

$$\|LowBits_q(\boldsymbol{Az}-c\boldsymbol{t}, 4\gamma')\|_{\infty}\leqslant 2(\gamma'-\beta')$$

成立, 结合 $\|c\boldsymbol{t}\|_{\infty}\leqslant 2\beta'$ 及引理 1 可知不会泄露信息.

对于 2 个用户 $P_i, i=1, 2$, 当 $\|\boldsymbol{z}_i\|_{\infty}\leqslant\gamma-\beta-1$ 且 $\|LowBits_q(\boldsymbol{w}_i-c\cdot sk_{i,2}, 2\gamma')\|_{\infty}$ 时得到合法的签名, 其概率为

$$Pr[\text{success}]=(\mathrm{e}^{-nl\beta/\gamma}\cdot\mathrm{e}^{-nk\beta'/\gamma'})^2=\mathrm{e}^{-2n(l\beta/\gamma+k\beta'/\gamma')}.$$

因此重复次数的期望值可以估计为

$$E=1/Pr[\text{success}]=\mathrm{e}^{2n(l\beta/\gamma+k\beta'/\gamma')}.$$

3.2 密钥和签名大小

P_i 的私钥为 $sk_i=(\boldsymbol{A}, \boldsymbol{t}_i, sk_{i,1}, sk_{i,2})$, 公钥为 $pk=(\boldsymbol{A}, \boldsymbol{t})$, 其中 $\boldsymbol{A}\in R_q^{k\times l}$, $\boldsymbol{t}\in R_q^{k}$.

1) 计算公钥大小. $\boldsymbol{A}=\sum_{i=1}^{2}\boldsymbol{A}_i=\boldsymbol{A}_1+\boldsymbol{A}_2$, 类似于

Dilithium签名方案以及Aigis-sig签名方案，用户$P_i(i=1,2)$的$\boldsymbol{A}_i$可以由256 b的种子生成，因此只需要保存种子，共512b，公钥（单位B）共需要约

$$\frac{512+n\cdot k\cdot\lceil \mathrm{lb}q\rceil}{8}.$$

2）计算私钥大小. P_i的私钥$sk_i=(\boldsymbol{A},\boldsymbol{t}_i,sk_{i,1},sk_{i,2})$，其中$sk_{i,1}\leftarrow_{\$}S_{\eta}^{l}$，$sk_{i,2}\leftarrow_{\$}S_{\eta'}^{k}$，私钥（单位B）共需要约

$$\frac{512}{8}+\frac{n\cdot k\cdot\lceil \mathrm{lb}q\rceil}{8}+\frac{n\cdot 1\cdot(\lceil \mathrm{lb}\eta\rceil+1)}{8}+\frac{n\cdot k\cdot(\lceil \mathrm{lb}\eta'\rceil+1)}{8}.$$

这里和文献[24]一样，由于$sk_{i,1}$和$sk_{i,2}$的每个系数都可能是负的元素，因此需要额外1b来表示符号.

3）计算签名大小. 签名由$(\boldsymbol{z},c,\boldsymbol{h})$共3个部分组成：

$\boldsymbol{z}$是一个l维向量，且满足$\|\boldsymbol{z}\|_{\infty}\leqslant 2(\gamma-\beta)-1$，因此$\boldsymbol{z}$的大小约为$n\cdot l\cdot(\lceil \mathrm{lb}((\gamma-\beta)-1)\rceil+1)$.

c是挑战值，和文献[13, 16, 22]选取自相同的挑战值集合，因此c的大小为$\tau\cdot(\lceil \mathrm{lb}n\rceil+1)$.

$$\boldsymbol{h}=\boldsymbol{w}_{\mathrm{H}}-\widetilde{\boldsymbol{w}}_{\mathrm{H}}=HighBits_q(\boldsymbol{A}\boldsymbol{z}-c\boldsymbol{t},4\gamma')-\left(\boldsymbol{w}_{1,\mathrm{H}}+\boldsymbol{w}_{2,\mathrm{H}}\bmod\frac{q-1}{2\gamma'}\right),$$

$\boldsymbol{h}$是一个k维向量且$\|\boldsymbol{h}\|_{\infty}\leqslant\frac{q-1}{4\gamma'}$，$\boldsymbol{h}$的大小为$n\cdot k\cdot\left(\left\lceil \mathrm{lb}\left(\frac{q-1}{4\gamma'}\right)\right\rceil+1\right)$.

综上，签名（单位B）共需要约

$$\frac{n\cdot l\cdot(\lceil \mathrm{lb}(2(\gamma-\beta)-1)\rceil+1)}{8}+\frac{\tau(\lceil \mathrm{lb}n\rceil+1)}{8}+\frac{n\cdot k\cdot\left(\left\lceil \mathrm{lb}\left(\frac{q-1}{4\gamma'}\right)\right\rceil+1\right)}{8}.$$

特别地，当$\eta'=\eta,\beta'=\beta$时，本文的方案与Dilizium 2.0两方协同签名方案本质上等价，也就是说Dilizium 2.0可以视为本文方案的一个特例.

我们选取Dilithium第2轮的参数[13]以及Aigis相对应的参数[16]进行实例化，参数如表1所示，得到的密钥大小和签名大小的对比见表2，签名成功所需重复次数的对比见表3.

由表2可知，我们提出的两方协同签名方案比文献[24, 29]拥有更短的密钥和签名. 而在决定运行时间的期望重复次数方面，虽然大于文献[29]的方案，但由文献[16]和表3可知，即使重复次数略大，在进行工程上的优化以及使用AVX2进行加速后，Aigis-sig甚至在一些参数下快于Dilithium（具体运行时间对比参见文献[16]），因此我们合理地认为实际情况下Aitps（本文提出的基于Aigis-sig的两方协同签名方案）与Dilizium 2.0（现有最优的基于Dilithium的两方协同签名方案）效率相当.

Table 2 Comparison in Key Sizes and Signature Sizes

表2 密钥大小及签名大小的比较

方案	公钥大小/B	每个用户的私钥大小/B	签名大小/B
Aitps-1024 (本文)	2 752	3 328	2 404
Dilizium 2.0-1024[29]	3 008	3 904	3 236
Dilizium-1024[24]	3 008	4 032	5 406
Aitps-1280 (本文)	3 584	4 480	3 140
Dilizium 2.0-1280[29]	3 744	4 896	4 196
Dilizium-1280[24]	3 744	5 024	6 750
Aitps-1536 (本文)	4 288	5 216	3 876
Dilizium 2.0-1536[29]	4 480	5 536	5 156
Dilizium-1536[24]	4 480	5 632	8 094

Table 3 Comparison in Expect Repeations

表3 期望重复次数比较

方案	期望重复次数	方案	期望重复次数
Aitps-1024 (本文)	$e^{3.534}$=34.3	Aigis-sig-1024[16]	5.86
Dilizium 2.0-1024[29]	$e^{3.495}$=33.0	Dilithium-1024[13]	5.90
Aitps-1280 (本文)	$e^{4.0575}$=57.8	Aigis-sig-1280[16]	7.61
Dilizium 2.0-1280[29]	$e^{3.763}$=43.1	Dilithium-1280[13]	6.60
Aitps-1536 (本文)	$e^{3.791}$=44.3	Aigis-sig-1536[16]	6.67
Dilizium 2.0-1536[29]	$e^{2.908}$=18.3	Dilithium-1536[13]	4.30

4 总结及展望

本文提出的Aitps是一种基于格的两方协同签名协议，允许2个用户在不泄露各自私钥的情况下通过交互对给定的消息进行签名，同时任何一方都无法重构密钥和独自完成整个签名过程，保证了密钥的安全性. 在协议的安全性方面，我们将其归约到非对称模格困难问题以及用到的承诺方案的计算隐藏性（本质上也是基于模格问题的困难性）. 由于现有的最优的相关方案Dilizium 2.0没有评估效果，我们也给出了Aitps方案各项评价指标的计算公式（也适用于Dilizium 2.0方案），并采用CRYSTALS-Dilithium和Aigis-sig的推荐参数进行实例化，相比之下，本文提出的方案比现有的所有基于Dilithium的两方签名方案具有更小的密钥和签名尺寸，特别是签名尺寸，能减少至少20%.

与文献[21]一样，本文提出的两方协同签名方案很容易扩展成为安全的多方协同签名方案. 在未来的工作中，我们将进一步考虑使用提交给NIST的

CRYSTALS-Dilithium 签名方案中对公钥的压缩技巧继续减小公钥尺寸, 以及尝试降低期望重复次数.

作者贡献声明: 文嘉明提出算法设计思路并撰写论文; 王后珍提出算法优化及分析思路, 并参与撰写论文; 刘金会和张焕国提出指导意见并修改论文.

参 考 文 献

[1] CNNIC. The 49th statistical report on China's Internet development [EB/OL]. (2022-02-25)[2022-07-23]. http://www.cnnic.net.cn/n4/2022/0401/c88−1131.html
(中国互联网络信息中心. 第49次《中国互联网络发展状况统计报告》[EB/OL]. (2022-02-25)[2022-07-23]. http://www.cnnic.net.cn/n4/2022/0401/c88−1131.html)

[2] Feng Qi, He Debiao, Luo Min, et al. Efficient two-party SM2 signing protocol for mobile Internet[J]. Journal of Computer Research and Development, 2020, 57(10): 2136−2146 (in Chinese)
(冯琦, 何德彪, 罗敏, 等. 移动互联网环境下轻量级 SM2两方协同签名[J]. 计算机研究与发展, 2020, 57(10): 2136−2146)

[3] Shoup V. Practical threshold signatures[C]//Proc of the 19th Int Conf on the Theory and Application of Cryptographic Techniques (EUROCRYPT). Berlin: Springer, 2000: 207−220

[4] Damgård I, Mikkelsen G L, Skeltved T. On the security of distributed multiprime RSA[C]//Proc of the 17th Int Conf on Information Security and Cryptology(ICISC). Berlin: Springer, 2014: 18−33

[5] Lindell Y. Fast secure two-party ECDSA signing[C]//Proc of the 37th Annual Int Cryptology Conf(CRYPTO). Berlin: Springer, 2017: 613−644

[6] Doerner J, Kondi Y, Lee E, et al. Secure two-party threshold ECDSA from ECDSA assumptions[C]//Proc of the 39th IEEE Symp on Security and Privacy(S&P). Piscataway, NJ: IEEE, 2018: 980−997

[7] Xue Haiyang, Au M H, Xie Xiang, et al. Efficient online-friendly two-party ECDSA signature[C]//Proc of the 27th ACM SIGSAC Conf on Computer and Communications Security(CCS). New York: ACM, 2021: 558−573

[8] Maxwell G, Poelstra A, Seurin Y, et al. Simple Schnorr multi-signatures with applications to bitcoin[J]. Designs, Codes and Cryptography, 2019, 87(9): 2139−2164

[9] Komlo C, Goldberg I. FROST: Flexible round-optimized Schnorr threshold signatures[C]//Proc of the 27th Selected Areas in Cryptography (SAC). Berlin: Springer, 2014: 34−65

[10] Garillot F, Kondi Y, Mohassel P. Threshold schnorr with stateless deterministic signing from standard assumptions[C]//Proc of the 41st Annual Int Cryptology Conf (CRYPTO). Berlin: Springer, 2021: 127−156

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

[12] Ajtai M. Generating hard instances of lattice problems (extended abstract)[C]//Proc of the 28th Annual ACM Symp on the Theory of Computing (STOC). New York: ACM, 1996: 99–108

[13] Ducas L, Durmus A, Lepoint T, et al. CRYSTALS-Dilithium: A lattice-based digital signature scheme[J]. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2018, 2018(1): 238−268

[14] Bos J, Ducas L, Kiltz E, et al. CRYSTALS-Kyber: A CCA-secure module-lattice-based KEM[C]//Proc of IEEE European Symp on Security and Privacy(EuroS&P). Piscataway, NJ: IEEE, 2018: 353–367

[15] Fouque P A, Hoffstein J, Kirchner P, et al. Falcon: Fast-Fourier lattice-based compact signatures over NTRU [EB/OL]. (2020-01-10)[2022-07-23].https://falcon-sign.info/

[16] Zhang Jiang, Yu Yu, Fan Shuqin, et al. Tweaking the asymmetry of asymmetric-key cryptography on lattices: KEMs and signatures of smaller sizes[C]//Proc of the 23rd IACR Int Conf on Practice and Theory of Public-Key Cryptography (PKC). Berlin: Springer, 2020: 37–65

[17] Lu Xianhui, Liu Yamin, Zhang Zhenfei, et al. LAC: Practical ring-LWE based public-key encryption with byte-level modulus [J/OL]. IACR Cryptology ePrint Archive, 2018 [2022-09-25].https://eprint.iacr.org/2018/1009

[18] Shen Shiyu, He Feng, Zhao Yunlei. Multi-platform efficient implementation and optimization of Aigis-enc algorithm[J]. Journal of Computer Research and Development, 2021, 58(10): 2238−2252 (in Chinese)
(沈诗羽, 何峰, 赵运磊. Aigis密钥封装算法多平台高效实现与优化[J]. 计算机研究与发展, 2021, 58(10): 2238−2252)

[19] Zhou Zhen, He Debiao, Luo Min, et al. Compact software/hardware co-design and implementation method of Aigis-sig digital signature scheme[J]. Chinese Journal of Network and Information Security, 2021, 7(2): 64−76 (in Chinese)
(周朕, 何德彪, 罗敏, 等. 紧凑的Aigis-sig 数字签名方案软硬件协同实现方法[J]. 网络与信息安全学报, 2021, 7(2): 64−76)

[20] Cozzo D, Smart N P. Sharing the LUOV: Threshold post-quantum signatures[C]//Proc of the 17th IMA Int Conf on Cryptography and Coding (IMACC). Berlin: Springer, 2019: 128–153

[21] Damgard I, Orlandi C, Takahashi A, et al. Two-round *n*-out-of-*n* and multi-signatures and trapdoor commitment from lattices[J]. Journal of Cryptology, 2022, 35(2): 1−56

[22] Lyubashevsky V. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures[C]//Proc of the 15th Int Conf on the Theory and Application of Cryptology and Information Security (ASIACRYPT). Berlin: Springer, 2009: 598–616

[23] Baum C, Damgård I, Lyubashevsky V et al. More efficient commitments from structured lattice assumptions[C]//Proc of the 11th Int Conf on Security and Cryptography for Networks (SCN). Berlin: Springer, 2018: 368–385

[24] Vakarjuk J, Snetkov N, Willemson J. Dilizium: A two-party lattice-based signature scheme[J]. Entropy, 2021, 23(8): 989−1018

[25] Lyubashevsky V, Micciancio D, Peikert C, et al. SWIFFT: A modest proposal for FFT hashing[C]//Proc of the 15th Int Conf on Fast Software Encryption (FSE). Berlin: Springer, 2008: 54–72

[26] Fukumitsu M, Hasegawa S. A lattice-based provably secure multisignature scheme in quantum random oracle model[C]//Proc of the 14th Int Conf on Provable and Security (ProvSec). Berlin:

Springer, 2020: 45–64

[27] Garcia-Escartin J C, Gimeno V, Moyano-Fernández J J. Quantum collision finding for homomorphic Hash functions [J/OL]. IACR Cryptology ePrint Archive, 2021 [2022-09-25].https://eprint.iacr.org/2021/1016

[28] Bai Shi, Galbraith S D. An improved compression technique for signatures based on learning with errors[C]//Proc of Cryptographers' Track at the RSA Conf (CT-RSA). Berlin: Springer, 2014: 28–47

[29] Laud P, Snetkov N, Vakarjuk J. DiLizium 2.0: Revisiting two-party crystals-Dilithium [J/OL]. IACR Cryptology ePrint Archive, 2022 [2022-09-25].https://eprint.iacr.org/2022/644

[30] Langlois A, Stehlé D. Worst-case to average-case reductions for module lattices[J]. Designs, Codes and Cryptography, 2015, 75(3): 565−599

[31] Chor B, Goldwasser S, Micali S. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract)[C]//Proc of the 26th Annual Symp on Foundations of Computer Science(FOCS). Los Alamitos, CA: IEEE Computer Society, 1985: 383–395

[32] Lyubashevsky V, Nguyen N K, Seiler G. SMILE: Set membership from ideal lattices with applications to ring signatures and confidential transactions[C]//Proc of the 41st Annual Int Cryptology Conf (CRYPTO). Berlin: Springer, 2021: 611–640

[33] Tian Yangtong, Zhang Huang, Xie Shaohao, et al [J]. Journal of Computer Research and Development, 2019, 56(10): 2229−2242 (in Chinese)
(田杨童, 张煌, 谢少浩, 等. 后量子的智能电表隐私保护方案[J]. 计算机研究与发展, 2019, 56(10): 2229−2242)

[34] Lyubashevsky V, Nguyen N K, Plancon M, et al. Shorter lattice-based group signatures via "almost free" encryption and other optimizations[C]//Proc of the 27th Int Conf on Theory and Application of Cryptology and Information Security (ASIACRYPT). Berlin: Springer, 2021: 218–248

[35] Esgin M F, Steinfeld R, Sakzad A, et al. Short lattice-based one-out-of-many proofs and applications to ring signatures[C]//Proc of the 17th Int Conf on Applied Cryptography and Network Security (ACNS). Berlin: Springer, 2019: 67–88

[36] NIST. PQC standardization process: Announcing four candidates to be standardized, plus fourth round candidates [EB/OL]. (2022-07-05)[2022-07-25].https://csrc.nist.gov/News/2022/pqc-candidates-to-be-standardized-and-round-4

[37] Bernstein D J, Hülsing A, Kölbl S, et al. The SPHINCS+ signature framework[C]//Proc of the 26th ACM SIGSAC Conf on Computer and Communications Security (CCS). New York: ACM, 2019: 2129−2146

[38] Güneysu T, Lyubashevsky V, Pöppelmann T. Practical lattice-based cryptography: A signature scheme for embedded systems[C]//Proc of the 14th Int Conf on Cryptographic Hardware and Embedded Systems (CHES). Berlin: Springer, 2012: 530–547

[39] Lyubashevsky V. Lattice signatures without trapdoors[C]//Proc of the 31st Int Conf on the Theory and Application of Cryptographic Techniques (EUROCRYPT). Berlin: Springer, 2012: 738−755

[40] Ducas L, Durmus A, Lepoint T, et al. Lattice signatures and bimodal Gaussians[C]// Proc of the 33rd Annual Int Cryptology Conf (CRYPTO). Berlin: Springer, 2013: 40–56

[41] Bruinderink L G, Hülsing A, Lange T, et al. Flush, gauss, and reload —A cache attack on the BLISS lattice-based signature scheme[C]// Proc of the 18th Int Conf on Cryptographic Hardware and Embedded Systems (CHES). Berlin: Springer, 2016: 323–345

[42] Pessl P, Bruinderink L G, Yarom Y. To BLISS-B or not to be: Attacking strongswan's implementation of post-quantum signatures[C]//Proc of the 24th ACM SIGSAC Conf on Computer and Communications Security (CCS). New York: ACM, 2017: 1843–1855

[43] Zhang Jiang, Yu Yu, Fan Shuqin, et al. Aigis: A family of signatures and key encapsulatIon mechanisms from asymmetric (M)LWE and (M)SIS (the part of digital signature) [EB/OL]. (2019-02-28)[2022-07-25].https://sfjs.cacrnet.org.cn/site/term/list_72_1.html

[44] Bellare M, Neven G. Multi-signatures in the plain public-key model and a general forking lemma [C]//Proc of the 13th ACM SIGSAC Conf on Computer and Communications Security (CCS). New York: ACM, 2006: 390–399

Wen Jiaming, born in 1997. PhD candidate. His main research interests include lattice-based cryptography and cryptography protocols.

文嘉明, 1997 年生. 博士研究生. 主要研究方向为基于格的密码学和密码协议.

Wang Houzhen, born in 1981. PhD, associate professor. Member of CCF. His main research interests include public key cryptography and post-quantum cryptography.

王后珍, 1981 年生. 博士, 副教授. CCF 会员. 主要研究方向为公钥密码学和抗量子密码学.

Liu Jinhui, born in 1989. PhD, associate professor. Her main research interests include cryptography and blockchain security.

刘金会, 1989 年生. 博士, 副教授. 主要研究方向为密码学和区块链安全.

Zhang Huanguo, born in 1945. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include information security, cryptography, and trusted computing.

张焕国, 1945 年生. 博士, 教授, 博士生导师. CCF 高级会员. 主要研究方向为信息安全、密码学和可信计算.