埃奎斯[1]：一类基于非对称(M)LWE和(M)SIS的数字签名和密钥封装机制

Aigis: A famIly of siGnatures and key encapsulatIon mechaniSms from asymmetric (M)LWE and (M)SIS

本文档为数字签名方案部分

[1] 埃奎斯(Aigis，希腊古典文字)：希腊神话中宙斯和雅典娜使用的盾牌都叫埃奎斯（其中前者又叫宙斯之盾，后者又叫雅典娜之盾），是希腊神话中唯一能抵抗宙斯“雷霆”攻击的法器。

基本信息

算法名称： 埃奎斯数字签名方案 (Aigis-sig)

第一设计者：张江，密码科学技术国家重点实验室
联系电话：15110204521 电子邮箱：jiangzhang09@gmail.com

其他设计者：郁昱，上海交通大学，yyuu@sjtu.edu.cn
范淑琴，密码科学技术国家重点实验室，shuqinfan78@163.com
张振峰，中国科学院软件研究所，zfzhang@tca.iscas.ac.cn
杨糠，密码科学技术国家重点实验室, yangk@sklc.org

算法联系人：杨糠，密码科学技术国家重点实验室
通信地址：北京市海淀区永翔北路9号
联系电话：18510249902 电子邮箱：yangk@sklc.org

提交日期： 2019年2月28日

目录

1 引言

当前，格上公钥密码系统的安全性大多是建立在小整数解问题（Small Integer Solutions[2,34]，SIS）和带错误的学习问题（Learning with Errors[41]，LWE）的困难性之上。简单来说，小整数解问题和带错误的学习问题都与求解模整数方程有关系。令$n, m, q \in \mathbb{Z}$为正整数，α, β为正实数，$\chi_\alpha \subseteq \mathbb{Z}$是以$\alpha \in \mathbb{R}$为参数的错误分布（通常为高斯分布，或与其相近的二项分布）。无穷范数（ℓ_∞）小整数解问题$\mathrm{SIS}^{\infty}_{n,m,q,\beta}$就是给定矩阵$\mathbf{A} \in \mathbb{Z}_q^{n\times m}$，计算非零向量$\mathbf{x} \in \mathbb{Z}^m$使其满足$\mathbf{Ax} = \mathbf{0} \bmod q$并且$\|\mathbf{x}\|_\infty \leq \beta$；而对应的计算性带错误的学习问题$\mathrm{LWE}_{n,m,q,\alpha}$就是给定样本$(\mathbf{A}, \mathbf{b} = \mathbf{As}+\mathbf{e}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$，求解$\mathbf{s} \in \mathbb{Z}_q^n$，其中$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}, \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e} \xleftarrow{\$} \chi_\alpha^m$。判定性LWE问题是区分$(\mathbf{A}, \mathbf{b} = \mathbf{As}+\mathbf{e})$和$\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$上的均匀随机元素。在一定参数下，判定性LWE问题和计算性LWE问题在多项式时间意义下是等价的[41,32]。此外，SIS问题和LWE问题在一定意义上互为对偶问题。

虽然SIS问题和LWE问题看起来比较简单，但在特定参数下求解这两个问题在平均情况下的复杂度都比求解格上某些问题（例如，最短向量问题）在最坏情况的复杂度还高[41,34]。[1] 由于目前已知的格上困难问题的量子求解算法与传统经典求解算法相比在计算复杂度上并没有本质的降低，以至于大多数国内外研究学者都倾向于相信格上问题是困难的，以及基于格上困难问题设计的密码系统能够抵抗量子计算机攻击。此外，当秘密向量$\mathbf{s}$并不是随机均匀地选自于$\mathbb{Z}_q^n$时，相应LWE的变种问题（称之为正规形LWE问题）也是困难的。特别地，当$\mathbf{s} \xleftarrow{\$} \chi_\alpha^n$与噪音向量$\mathbf{e}$选自于同样的分布时，正规形LWE问题和标准的LWE问题在多项式时间的意义上是等价的[7]。由于正规形LWE问题能够更好的控制噪音增长，因此在文献中被广泛用于设计加密方案[17,15]。

一般来说，SIS问题大多被用于设计数字签名方案，而LWE问题则常常用于设计公钥加密方案。但我们研究发现标准的SIS问题以及LWE问题并不能非常好地满足我们对于密码系统性能，特别是最重要的通信性能的要求。根据对已有格上密码系统设计技术的深入思考和理解，我们提出了非对称的SIS和LWE变形问题。简单来说，非对称SIS问题(Asymmetric SIS, ASIS)$\mathrm{ASIS}^{\infty}_{n,m_1,m_2,q,\beta_1,\beta_2}$就是给定矩阵$\mathbf{A} \in \mathbb{Z}_q^{n\times(m_1+m_2)}$，计算非零向量$\mathbf{x} = (\mathbf{x}_1^T, \mathbf{x}_2^T)^T \in \mathbb{Z}^{m_1+m_2}$使其满足$\mathbf{Ax} = \mathbf{0} \bmod q$, $\|\mathbf{x}_1\|_\infty \leq \beta_1$, 并且$\|\mathbf{x}_2\|_\infty \leq \beta_2$。显然，求解$\mathrm{ASIS}^{\infty}_{n,m_1,m_2,q,\beta_1,\beta_2}$问题不会比求解$\mathrm{SIS}^{\infty}_{n,m_1+m_2,q,\min(\beta_1,\beta_2)}$问题更困难，但同样也不会比求解$\mathrm{SIS}^{\infty}_{n,m_1+m_2,q,\max(\beta_1,\beta_2)}$问题更容易。换句话说，在计算困难性上，我们有如下关系

$$\mathrm{SIS}^{\infty}_{n,m_1+m_2,q,\max(\beta_1,\beta_2)} \leq \mathrm{ASIS}^{\infty}_{n,m_1,m_2,q,\beta_1,\beta_2} \leq \mathrm{SIS}^{\infty}_{n,m_1+m_2,q,\min(\beta_1,\beta_2)}$$

[1] 这种平均困难性到最坏困难性的联系特性实际上是基于格上困难问题的密码方案相对于基于其他困难问题的密码方案独有的优势之一。

成立。这种关系实际上为我们基于ASIS设计密码方案建立了理论基础。此外，我们还提出了一类针对ASIS问题的分析方法，并给出了ASIS问题不同安全强度下的参数选取方法，从而解决了基于ASIS问题密码系统的实际参数选取问题。

相应地，非对称LWE问题（Asymmetric LWE, ALWE）$\mathrm{ALWE}_{n,m,q,\alpha_1,\alpha_2}$是给定样本$(\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$，求解$\mathbf{s} \in \mathbb{Z}_q^n$，其中$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}, \mathbf{s} \xleftarrow{\$} \chi_{\alpha_1}^n, \mathbf{e} \xleftarrow{\$} \chi_{\alpha_2}^m$。由于求解算法可能会利用秘密向量$\mathbf{s}$的分布信息，我们不能简单地像比较ASIS与SIS一样比较ALWE和LWE的关系，但对于目前已知最好的求解算法我们仍然有关系

$$\mathrm{LWE}_{n,m,q,\min(\alpha_1,\alpha_2)} \leq \mathrm{ALWE}_{n,m,q,\alpha_1,\alpha_2} \leq \mathrm{LWE}_{n,m,q,\max(\alpha_1,\alpha_2)}$$

成立。[2] 更重要的是，文献[25,16,33]研究表明只要$\chi_{\alpha_1}^n$具有足够高的熵（例如，$\{0,1\}^n$上的均匀分布），那么我们总可以选择其他参数使得$\mathrm{ALWE}_{n,m,q,\alpha_1,\alpha_2}$达到标准LWE问题的困难强度。这就为我们基于ALWE设计密码系统提供了理论基础。此外，Cheon等人[19]实际上使用了与ALWE相关的变种问题，即$\mathbf{s}$和$\mathbf{e}$选自于不同分布（注意，我们定义的ALWE仅仅指$\mathbf{s}$和$\mathbf{e}$选自于参数不同的相同分布）来设计加密方案。通过综合比较分析和优化最新的LWE求解算法，我们给出了ALWE问题和LWE问题参数之间的近似关系，以及ALWE问题不同安全强度下的参数选取方法，从而解决了基于ALWE问题密码系统的实际参数选取问题。

显然，以上非对称性变种问题的定义可以很自然地推广到环LWE问题/SIS问题（RLWE/RSIS）和模LWE问题/SIS问题（Module-LWE/SIS，MLWE/MSIS）问题。鉴于众多研究已经表明MLWE/MSIS问题能够在计算效率和通信代价方面实现较好的平衡[21,14]，我们选择使用非对称MLWE问题（Asymmetric MLWE, AMLWE）和非对称MSIS问题（Asymmetric MSIS，AMSIS）来设计具体的密码系统。特别地，通过在签名方案中充分利用AMSIS和AMLWE的非对称特性（详见埃奎斯签名方案文档第1.1的设计原理部分），我们给出了比文献中已有的格上签名方案具有更好综合性能的数字签名方案，并将之命名为埃奎斯签名方案（Aigis-sig）；通过在密钥封装方案中充分利用AMLWE困难问题的非对称特性（详见埃奎斯密钥封装机制文档第1.1节的设计原理部分），我们给出了比文献中已有格上密钥封装方案具有更好综合性能的密钥封装方案，并将之命名为埃奎斯密钥封装机制（Aigis-enc）。

1.1 埃奎斯签名方案的设计原理

目前一大类格上的签名方案都可以看成著名的Schnorr签名[42]在格上的变种。技术上，该类签名方案在设计上往往是通过Fiat-Shamir技术[22]将三轮的格

[2] 实际上，对于特定的分布（例如高斯分布[41]），我们也可以从理论上证明这种困难关系成立。

上身份证明协议转换成数字签名方案的。考虑如下基于标准$\mathrm{SIS}^{\infty}_{n,m,q,\beta}$问题的身份证明协议：用户$A$拥有公钥$pk=(\mathbf{A},\mathbf{t}=\mathbf{Ax})\in\mathbb{Z}_q^{n\times m}\times\mathbb{Z}_q^n$和私钥$sk=\mathbf{x}\in\mathbb{Z}_q^m$，用户$B$知道$A$的公钥$pk$并想要用户$A$证明其拥有对应的私钥。为了达成这个目的，用户$A$首先从某个分布里面随机选择一个元素$\mathbf{y}\in\mathbb{Z}_q^m$，然后计算$\mathbf{w}=\mathbf{Ay}$并将$\mathbf{w}$发送给用户$B$；当收到消息$\mathbf{w}\in\mathbb{Z}_q^n$后，用户$B$随机选择$c\in\{0,1\}$并将其作为挑战发送给$A$；此后用户$A$计算$\mathbf{z}=\mathbf{y}+c\mathbf{x}$给$B$；当收到消息$\mathbf{z}\in\mathbb{Z}_q^m$，用户$B$通过验证等式$\mathbf{Az}=\mathbf{w}+c\mathbf{t}$是否成立来确定是否接受用户$A$的证明。

为了保证以上协议的合理性（即用户A不能欺骗用户B），用户B还需要验证向量$\beta_2=\|\mathbf{z}\|_\infty$是否足够小（即保证对应的$\mathrm{SIS}^{\infty}_{n,m,q,\beta}$问题足够困难），否则任何人都可以通过求解线性方程组完成以上证明。此外，为了保证以上身份证明协议的安全性（即用户B不能从公钥中计算出私钥，且也不能通过交互得到用户A的私钥），又必须要求$\beta_1=\|\mathbf{x}\|_\infty$要足够小，但$\|\mathbf{y}\|_\infty\gg\|\mathbf{x}\|_\infty$（因此$\|\mathbf{z}\|_\infty\gg\|\mathbf{x}\|_\infty$）以至于随机向量$\mathbf{y}$可以掩盖$\mathbf{z}$中的私钥信息$\mathbf{x}$。为了提供有意义的安全性，我们通常需要要求$\beta_2/\beta_1>2^{\omega(\log\kappa)}$，即$\beta_2$必须是$\beta_1$的超多项式倍，其中$\kappa$是安全参数。由此得到的身份证明协议，以及通过Fiat-Shamir技术转换得到的签名方案都具有非常大的参数。为了解决这个问题，大多数文献都使用了拒绝采样技术[30,31]，即用户A只有在确认$\mathbf{z}$不会泄露$\mathbf{x}$的信息时才输出$\mathbf{z}$，其他时候则直接终止并重新运行一个新的协议直到成功为止。虽然这种方法可以极大地降低系统的参数（只需要要求$\beta_2/\beta_1=\mathsf{poly}(\kappa)$即可），但对于一个交互的协议来说却带来了较大的效率损失。但幸运的是，当我们用Fiat-Shamir技术将该类身份证明协议转换为数字签名方案时，这种损失被极大地弱化了。

对于任意正整数n,q，令环$R_q=\mathbb{Z}_q[x]/(x^n+1)$。环$R_q$上的无穷范数（$\ell_\infty$）MSIS问题$\mathrm{MSIS}^{\infty}_{n,q,k,\ell,\beta}$就是给定矩阵$\mathbf{A}\in R_q^{k\times\ell}$，计算非零向量$\mathbf{x}\in R^\ell$使其满足$\mathbf{Ax}=\mathbf{0}\bmod q$并且$\|\mathbf{x}\|_\infty\le\beta$。对于任意正整数$\eta$，定义$S_\eta\subset R_q$是由无穷范数小于等于$\eta$的多项式构成的集合（即多项式中每个系数的绝对值都小于等于η）。令$\mathsf{H}:\{0,1\}^*\to R_2$是一个杂凑函数。将以上身份证明协议中的技术推广到MSIS问题上，并且应用Fiat-Shamir技术即可得到如下签名方案：

- **密钥生成算法：**随机选择矩阵$\mathbf{A}\xleftarrow{\$}R_q^{k\times\ell}$，计算$\mathbf{t}=\mathbf{Ax}$，最终输出公钥$pk=(\mathbf{A},\mathbf{t})$，私钥$sk=(\mathbf{x},pk)$，其中$\mathbf{x}\xleftarrow{\$}S_\eta^\ell$。
- **签名生成算法：**给定私钥$sk=(\mathbf{x},pk)$和消息$\mu\in\{0,1\}^*$，
  1. 随机选择$\mathbf{y}\xleftarrow{\$}S_{\gamma-1}^\ell$；
  2. 计算$\mathbf{w}=\mathbf{Ay}$和$c=\mathsf{H}(\mathbf{w}\|\mu)$；
  3. 计算$\mathbf{z}=\mathbf{y}+c\mathbf{x}$；
  4. 如果$\|\mathbf{z}\|_\infty\ge\gamma-\beta$，重新回到第一步开始计算，否则输出签名$\sigma=(\mathbf{z},c)$，其中$\|c\mathbf{x}\|_\infty\le\beta$对于所有可能的$c$和$\mathbf{x}$都成立。

– **签名验证算法**：给定公钥$pk=(\mathbf{A},\mathbf{t})$，消息$\mu\in\{0,1\}^*$和签名$\sigma=(\mathbf{z},c)$，如果$\|\mathbf{z}\|_\infty<\gamma-\beta$且$c=\mathsf{H}(\mathbf{Az}-c\mathbf{t}\|\mu)$，返回1（即接受签名），否则返回0。

简单来说，以上签名方案的安全性依赖于$\mathrm{MSIS}^\infty_{n,q,k,\ell,\eta}$问题来保证不能从公钥$pk$中计算出私钥$sk$和$\mathrm{MSIS}^\infty_{n,q,k,\ell,2\gamma}$问题来保证不能伪造出签名。由于$\|c\mathbf{x}\|_\infty\leq\beta$总是成立，所以对于任意可能的$c,\mathbf{x}$和任意给定的$\mathbf{z}$，都存在$\mathbf{y}\in S^\ell_\gamma$使得$\mathbf{z}=\mathbf{y}+c\mathbf{x}$，这就保证了签名不会泄露私钥的信息。在效率上，拒绝采样技术可能会使签名算法运行约$\left(\frac{2(\gamma-\beta)-1}{2\gamma-1}\right)^{-n\cdot\ell}$次才能成功生成签名，且签名长度约等于$n\ell\lceil\log_2(2(\gamma-\beta)-1)\rceil+n$。即在固定维数$n$的情况下，签名长度主要取决于$\ell$和$\log_2(2(\gamma-\beta)-1)$的值。显然，为了提高计算效率，我们希望$\beta\ll\gamma$且$\ell$很小，为了提高通信效率，我们希望$\gamma$和$\ell$都很小。但为了保证MSIS问题的困难性，我们并不能使用太小的ℓ。

为了解决以上问题，文献中建议使用LWE问题及其变种问题来产生签名方案的密钥，从而降低密钥和签名的长度。特别地，令环R_q上（均匀噪音分布）的MLWE问题$\mathrm{MLWE}_{n,k,\ell,q,\alpha}$就是给定样本$(\mathbf{A},\mathbf{b}=\mathbf{As}_1+\mathbf{s}_2)\in R_q^{k\times\ell}\times R_q^k$，求解$\mathbf{s}_1\in R_q^\ell$，其中$\mathbf{s}_1\xleftarrow{\$}S^\ell_\eta,\mathbf{s}_2\xleftarrow{\$}S^k_\eta$。显然，可以将MLWE问题看成是关于矩阵$\bar{\mathbf{A}}=(\mathbf{A}\|\mathbf{I}_k)\in R_q^{k\times(\ell+k)}$的MSIS问题。由于MLWE问题可以允许使用较小的ℓ和k，我们可以改进以上签名方案：

– **密钥生成算法**：随机选择矩阵$\mathbf{A}\xleftarrow{\$}R_q^{k\times\ell}$，计算$\mathbf{t}=\mathbf{As}_1+\mathbf{s}_2$，最终输出公钥$pk=(\mathbf{A},\mathbf{t})$，私钥$sk=(\mathbf{s}_1,\mathbf{s}_2,pk)$，其中$\mathbf{s}_1\xleftarrow{\$}S^\ell_\eta,\mathbf{s}_2\xleftarrow{\$}S^k_\eta$。

– **签名生成算法**：给定私钥$sk=(\mathbf{s}_1,\mathbf{s}_2,pk)$和消息$\mu\in\{0,1\}^*$，

1. 随机选择$\mathbf{y}\xleftarrow{\$}S^{\ell+k}_{\gamma-1}$；
2. 计算$\mathbf{w}=(\mathbf{A}\|\mathbf{I}_k)\mathbf{y}$和$c=\mathsf{H}(\mathbf{w}\|\mu)$；
3. 计算$\mathbf{z}=\mathbf{y}+c\begin{pmatrix}\mathbf{s}_1\\\mathbf{s}_2\end{pmatrix}$；
4. 如果$\|\mathbf{z}\|_\infty\geq\gamma-\beta$，重新回到第一步开始计算，否则输出签名$\sigma=(\mathbf{z},c)$，其中$\left\|c\begin{pmatrix}\mathbf{s}_1\\\mathbf{s}_2\end{pmatrix}\right\|_\infty\leq\beta$对于所有可能的$c,\mathbf{s}_1,\mathbf{s}_2$都成立。

– **签名验证算法**：给定公钥$pk=(\mathbf{A},\mathbf{t})$，消息$\mu\in\{0,1\}^*$和签名$\sigma=(\mathbf{z},c)$，如果$\|\mathbf{z}\|_\infty<\gamma-\beta$且$c=\mathsf{H}((\mathbf{A}\|\mathbf{I}_k)\mathbf{z}-c\mathbf{t}\|\mu)$，返回1（即接受签名），否则返回0。

进一步，考虑到$\mathbf{w}=(\mathbf{A}\|\mathbf{I}_k)\mathbf{y}=\mathbf{Ay}_1+\mathbf{y}_2$以及$\gamma\ll q$的性质，我们有$\mathbf{w}$中每个系数表示成比特串时的高位比特基本由$\mathbf{Ay}_1$的高位比特决定。因此，文献[10,21]提出了只利用$\mathbf{Ay}_1$的高位比特来压缩签名长度的方法。特别地，令$\mathsf{HighBits}(\mathbf{z},2\gamma_2)$表示取向量$\mathbf{z}$中每个多项式中的系数高位比特（具体取多少比特由参数γ_2来控制）。对应地，令$\mathsf{LowBits}(\mathbf{z},2\gamma_2)$表示取向量$\mathbf{z}$中每个多项式中的系数低位比特。以下修改的签名方案将依赖于HighBits和LowBits来保证安全性和正确性：

- **密钥生成算法**：随机选择矩阵$\mathbf{A} \xleftarrow{\$} R_q^{k\times\ell}$，计算$\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$，最终输出公钥$pk = (\mathbf{A}, \mathbf{t})$，私钥$sk = (\mathbf{s}_1, \mathbf{s}_2, pk)$，其中$\mathbf{s}_1 \xleftarrow{\$} S_\eta^\ell, \mathbf{s}_2 \xleftarrow{\$} S_\eta^k$。
- **签名生成算法**：给定私钥$sk = (\mathbf{s}_1, \mathbf{s}_2, pk)$和消息$\mu \in \{0,1\}^*$，
  1. 随机选择$\mathbf{y} \xleftarrow{\$} S_{\gamma_1-1}^\ell$；
  2. 计算$\mathbf{w} = \mathbf{A}\mathbf{y}$和$c = \mathsf{H}(\mathsf{HighBits}(\mathbf{w}, 2\gamma_2)\|\mu)$；
  3. 计算$\mathbf{z} = \mathbf{y} + c\mathbf{s}_1$；
  4. 如果$\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$或者$\mathsf{LowBits}(\mathbf{A}\mathbf{y} - c\mathbf{s}_2, 2\gamma_2) \geq \gamma_2 - \beta$，重新回到第一步开始计算，否则输出签名$\sigma = (\mathbf{z}, c)$，其中$\|c\mathbf{s}_1\|_\infty, \|c\mathbf{s}_2\|_\infty \leq \beta$对于所有可能的$c, \mathbf{s}_1, \mathbf{s}_2$都成立。
- **签名验证算法**：给定公钥$pk = (\mathbf{A}, \mathbf{t})$，消息$\mu \in \{0,1\}^*$和签名$\sigma = (\mathbf{z}, c)$，如果$\|\mathbf{z}\|_\infty < \gamma_1 - \beta$且$c = \mathsf{H}(\mathsf{HighBits}(\mathbf{A}\mathbf{z} - c\mathbf{t}, 2\gamma_2)\|\mu)$，返回1（即接受签名），否则返回0。

本质上，签名算法第4步的拒绝采样提供两项功能：1）在安全性上，保证了签名不会泄露私钥$\mathbf{s}_1, \mathbf{s}_2$的信息；2）在正确性上，保证了$\mathsf{HighBits}(\mathbf{A}\mathbf{z} - c\mathbf{t}, 2\gamma_2) = \mathsf{HighBits}(\mathbf{A}\mathbf{y} - c\mathbf{s}_2, 2\gamma_2) = \mathsf{HighBits}(\mathbf{A}\mathbf{y}, 2\gamma_2)$，因为$\mathbf{A}\mathbf{y} = \mathbf{A}\mathbf{y} - c\mathbf{s}_2 + c\mathbf{s}_2$，$\|c\mathbf{s}_2\|_\infty \leq \beta$且$\mathsf{LowBits}(\mathbf{A}\mathbf{y} - c\mathbf{s}_2, 2\gamma_2) < \gamma_2 - \beta$。如果我们取$\gamma_1 = 2\gamma_2$，那么上述改进的签名方案依赖于$\mathrm{MLWE}_{n,k,\ell,q,\eta}$问题来保证敌手不能从公钥$pk$中计算出私钥$sk$，并且依赖于变形的$\mathrm{MSIS}^\infty_{n,k,(\ell+k+1),q,2\gamma_1+2}$问题来保证敌手不能伪造出签名。

仔细观察以上签名方案可以发现，其计算效率由第4步中的判断$\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$而导致的拒绝采样次数和判断$\mathsf{LowBits}(\mathbf{A}\mathbf{y} - c\mathbf{s}_2, 2\gamma_2) \geq \gamma_2 - \beta$拒绝采样次数来共同决定，其中前者需要的重复次数大约为$\left(\frac{2(\gamma_1-\beta)-1}{2\gamma_1-1}\right)^{-n\cdot\ell}$，而后者需要的重复次数大约为$\left(\frac{2(\gamma_2-\beta)-1}{2\gamma_2-1}\right)^{-n\cdot k}$。**由于总的计算效率由前后两个部分来共同决定，那么我们实际上可以通过调节参数在保证总的重复次数不变的情况下改变前后两个判断导致的重复次数**。此外，该签名方案的签名长度与γ_1的大小密切相关而与γ_2无关。**特别地，我们实际上可以通过减小γ_1的值来减小签名长度**。但单纯地降低γ_1的值会极大地增加第4步中第一个判断失败的概率。为了在尽可能保证计算效率的情况下减小签名的长度，我们选择减小$\|\mathbf{s}_1\|_\infty$的值（从而减小$\|c\mathbf{s}_1\|_\infty$的值）来缓解由于减小γ_1而带来的第4步中第一个判断失败概率的增大，同时通过增加$\|\mathbf{s}_2\|_\infty$的值来保持从公钥恢复私钥的困难性（其对应困难问题恰好是AMLWE问题）。此外，我们还选择增大γ_2的值来降低第4步中第二个判断失败的概率，从而进一步缓解由于减小γ_1而带来的总计算效率的损失。特别地，单纯地减小γ_1实际上增加了伪造签名的难度，从而在一定程度上弥补了增大γ_2而带来的伪造签名的难度损失（其对应的问题恰好是AMSIS问题）。特别地，定义环R_q上的无穷范数（ℓ_∞）AMSIS问题$\mathrm{AMSIS}^\infty_{n,k,\ell_1,\ell_2,q,\beta_1,\beta_2}$就是给定矩阵$\mathbf{A} \in R_q^{k\times(\ell_1+\ell_2)}$，计算非零向

量$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \in R^{\ell_1+\ell_2}$满足$\mathbf{A}\mathbf{x} = \mathbf{0} \bmod q$并且$\|\mathbf{x}_1\|_\infty \leq \beta_1, \|\mathbf{x}_2\|_\infty \leq \beta_2$；定义$R_q$上（均匀噪音分布）的AMLWE问题$\mathrm{AMLWE}_{n,k,\ell,q,\eta_1,\eta_2}$就是给定样本$(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2) \in R_q^{k\times\ell} \times R_q^k$，求解$\mathbf{s}_1 \in R_q^\ell$，其中$\mathbf{s}_1 \xleftarrow{\$} S_{\eta_1}^\ell, \mathbf{s}_2 \xleftarrow{\$} S_{\eta_2}^k$。如引言中讨论的一样，由AMLWE和MLWE的关系，以及AMSIS和MSIS的关系，以上设计策略在理论上是可行的。实际上，通过对目前格上最好的求解算法及其变种问题的分析，与格上同类方案相比我们的方案的确能在保证安全性不变的前提下实现更好的综合效率，特别是拥有更短的公钥、私钥和签名长度。

1.2 文档结构

第2节将介绍一些基本符号和操作。第3节将给出埃奎斯签名方案的具体描述。第4节将给出埃奎斯签名方案的实现和性能。第5节给出方案的安全证明。第6节分析了算法抵抗已知攻击的能力。第7节对算法的优缺点进行了总结。

2 符号及参数

$\mathcal{B}$	8比特无符号整数的集合(字节)，即$\{0, \ldots, 255\}$
$\mathcal{B}^k$	$\underbrace{\mathcal{B} \times \cdots \times \mathcal{B}}_{k}$
$\mathcal{B}^*$	$\mathcal{B}^1 \cup \mathcal{B}^2 \cup \ldots \cup \mathcal{B}^i \cup \ldots$
a_i	对于字节或比特数组a，a_i表示数组a的第i个元素(索引从0开始)
$a + k$	对于长度为ℓ的字节数组a和整数$k \in \{0, 1, \ldots, \ell - 1\}$，$a + k$表示从$a$的第$k$个元素开始的字节数组
$\|$	对于字符串(或字节数组) a和b, $a\|b$表示a与b的级联
$:=$	赋值操作，$a := b$表示将a赋值为b
κ	安全参数
e	自然常数2.71828...
$\log$	底为e的对数函数
$\log_2$	底为2的对数函数
$\mathbb{R}$	实数集合
$\mathbb{Z}$	整数集合$\{\ldots, -2, -1, 0, 1, 2, \ldots\}$
$\mathbb{Z}_q$	商环$\mathbb{Z}/q\mathbb{Z}$，其中$q \in \mathbb{N}$为正整数

$\mathbb{Z}_q^n$	$\underbrace{\mathbb{Z}_q \times \cdots \times \mathbb{Z}_q}_{n}$
$\mathbb{N}$	正整数集合$\{1, 2, 3, \ldots\}$
$\lceil \cdot \rceil$	向上最近取整函数，对于有实数$x \in \mathbb{R}$，$\lceil x \rceil$表示大于等于x的最小整数
$\lfloor \cdot \rfloor$	向下最近取整函数，对于有实数$x \in \mathbb{R}$，$\lfloor x \rfloor$表示小于等于x的最大整数
$\mathbf{x} \xleftarrow{\$} D$	根据分布D，选取$\mathbf{x}$
$\mathbf{x} \xleftarrow{\$} \mathcal{S}$	从一个集合$\mathcal{S}$中均匀随机选取$\mathbf{x}$
$\mathsf{negl}(\cdot)$	可忽略函数
$\circ$	逐点相乘，即对于$\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{Z}_q^n$和$\mathbf{y} = (y_1, y_2, \ldots, y_n) \in \mathbb{Z}_q^n$，$\mathbf{x} \circ \mathbf{y} = (x_1 y_1, x_2 y_2, \ldots, x_n y_n)$
$\mathsf{Trunc}(\cdot, \cdot)$	函数$\mathsf{Trunc}(x, \ell) : \{0,1\}^n \to \{0,1\}^\ell$将截取比特串$x$最后$\ell \leq n$比特
R	商环$R = \mathbb{Z}[X]/(X^n + 1)$，其中$n = 2^{n'-1}$使得$X^n + 1$是$2^{n'}$次分圆多项式
R_q	商环$R_q = \mathbb{Z}_q[X]/(X^n + 1)$，其中$n = 2^{n'-1}$使得$X^n + 1$是$2^{n'}$次分圆多项式
R^k	$\underbrace{R \times \cdots \times R}_{k}$
R_q^k	$\underbrace{R_q \times \cdots \times R_q}_{k}$
$a, b, \ldots$	环R或R_q(包括环$\mathbb{Z}$和$\mathbb{Z}_q$)中的元素用小写字母表示
$\mathbf{a}$, $\mathbf{b}$, ...	粗体小写字母表示列向量
$\mathbf{A}$, $\mathbf{B}$	粗体大写字母表示矩阵
$\mathbf{a}^T$或$\mathbf{A}^T$	向量$\mathbf{a}$或矩阵$\mathbf{A}$的转置
$\mathbf{a}[i]$	向量$\mathbf{a}$的第i个元素(索引从0开始)
$\mathbf{A}[i][j]$	矩阵$\mathbf{A}$第i行, 第j列的元素(索引从0开始)
$\mathsf{BytesToBits}$	函数$\mathsf{BytesToBits}$输入长度为ℓ的字节数组，输出长度为8ℓ的比特数组，即对于$\alpha = \mathsf{BytesToBits}(a)$，那么$\alpha_i := \left(\lfloor a_{\lfloor i/8 \rfloor} / 2^{(i \bmod 8)} \rfloor \bmod 2\right)$
$\mathsf{BitsToBytes}$	函数$\mathsf{BitsToBytes}$为函数$\mathsf{BytesToBits}$的逆，将输入长度为ℓ的比特数组转换为输出长度$\ell/8$的字节数组，即对于$\alpha = \mathsf{BitsToBytes}(a)$，那么$a_i := \alpha_{8i+7} \cdot 2^7 + \cdots + \alpha_{8i+1} \cdot 2 + \alpha_{8i}$
$\mathcal{HW}(\mathbf{v})$	对于比特向量$\mathbf{v}$，$\mathcal{HW}(\mathbf{v})$表示向量$\mathbf{v}$中1的个数；对于多项式向量$\mathbf{v}$，$\mathcal{HW}(\mathbf{v})$表示向量$\mathbf{v}$中所有多项式的非零系数的总个数

2.1 基本定义

可忽略函数. 可忽略函数是一个函数$\mathsf{negl}:\mathbb{N}\to[0,1]$满足对于每一个正整数$c$，存在一个整数$K$，满足对于所有$k>K$使得$\mathsf{negl}(k)<1/k^c$成立。

统计距离. 两个概率分布X和Y之间的统计距离定义为：

$$\Delta=\frac{1}{2}\cdot\sum_{\alpha}|\Pr[X=\alpha]-\Pr[Y=\alpha]|$$

如果X与Y之间的统计距离是可忽略的，那么我们称X统计接近于Y。

2.2 基本操作

模约化. 对于一个正偶数α，定义$r'=r \bmod^{\pm}\alpha$是在$(-\frac{\alpha}{2},\frac{\alpha}{2}]$范围内的唯一元素$r'$满足$r'=r \bmod \alpha$成立。对于一个正奇数$\alpha$，定义$r'=r \bmod^{\pm}\alpha$是在$[-\frac{\alpha-1}{2},\frac{\alpha-1}{2}]$范围内的唯一元素$r'$满足$r'=r \bmod \alpha$成立。对于任意正整数$\alpha$，定义$r'=r \bmod^{+}\alpha$是在$[0,\alpha)$范围内唯一的元素$r'$满足$r'=r \bmod \alpha$成立。当精确的表示不重要的时候，简写为$r \bmod \alpha$。

元素的范数. 对于一个元素$w\in\mathbb{Z}_q$，$\|w\|_\infty$表示$|w \bmod^{\pm} q|$。下面定义关于商环R上元素$w=w_0+w_1X+\cdots+w_{n-1}X^{n-1}\in R$的$\ell_\infty$和$\ell_2$范数：

$$\|w\|_\infty=\max_i\|w_i\|_\infty,\ \|w\|=\sqrt{\|w_0\|_\infty^2+\ldots+\|w_{n-1}\|_\infty^2}$$

相应地，对于向量$\mathbf{w}=(w_1,\ldots,w_k)\in R^k$，定义

$$\|\mathbf{w}\|_\infty=\max_i\|w_i\|_\infty,\ \|\mathbf{w}\|=\sqrt{\|w_1\|^2+\ldots+\|w_k\|^2}$$

对称密码组件. 埃奎斯签名方案将使用两个扩展输出函数$\mathsf{XOF}_1:\mathcal{B}^*\to\mathcal{B}^*$，$\mathsf{XOF}_2:\mathcal{B}^*\to\mathcal{B}^*$，和一个抗碰撞杂凑函数$\mathsf{CRH}:\{0,1\}^*\to\{0,1\}^{384}$。

数论变换(NTT)定义域表示. 对于商环$R_q=\mathbb{Z}_q[X]/(X^n+1)$，我们将选择模数$q$满足存在一个$2n$次单位根$r \bmod q$。在这种情况下，分圆多项式$X^n+1$完全分解为线性因子$X-(r^i \bmod q)$对于$i=1,3,5,\ldots,2n-1$。根据中国剩余定理(CRT)，分圆环$R_q$与环$\mathbb{Z}_q[X]/(X-r^i)\cong\mathbb{Z}_q$的直积同构，即$R_q\cong\prod_i\mathbb{Z}_q[X]/(X-r^i)$。此外，我们能够利用快速傅里叶变换(FFT)快速计算该同构

$$a\mapsto(a(r),a(r^3),\ldots,a(r^{2n-1})):R_q\to\prod_i\mathbb{Z}_q[X]/(X-r^i)$$

当基域是有限域的时候，FFT也称为NTT。我们实现的快速NTT算法并没有按照顺序$a(r), a(r^3), \ldots, a(r^{2n-1})$输出，而是输出

$$\hat{a} = \mathsf{NTT}(a) = (a(r_0), a(-r_0), \ldots, a(r_{n/2-1}), a(-r_{n/2-1}))$$

其中$r_i = r^{\mathsf{brv}(n/2+i)}$和$\mathsf{brv}(k)$表示$k(\log_2 n$比特)的比特逆序。利用NTT变换及其逆变换$\mathsf{NTT}^{-1}$，我们可以快速的计算$R_q$中的元素乘法。特别地，对于$a, b \in R_q$，我们有$a \cdot b = \mathsf{NTT}^{-1}(\mathsf{NTT}(a) \circ \mathsf{NTT}(b))$。对于向量$\mathbf{v}$和矩阵$\mathbf{A}$，$\hat{\mathbf{v}} = \mathsf{NTT}(\mathbf{v})$和$\hat{\mathbf{A}} = \mathsf{NTT}(\mathbf{A})$意味着构成$\mathbf{v}$和$\mathbf{A}$的每个多项式$\mathbf{v}[i]$和$\mathbf{A}[i][j]$都是NTT域表示形式。

均匀随机采样R_q中的多项式. 我们将采用算法1中的函数$\mathsf{Parse} : \mathcal{B}^* \to R_q$来确定性地采样$R_q$中的随机均匀元素，其中$16 < \lceil \log_2 q \rceil \leq 24$。特别地。该函数以一个字节流$B = b_0, b_1, b_2, \ldots$作为输入，然后输出$R_q$中的一个元素$\hat{a} = \hat{a}_0 + \hat{a}_1 X + \cdots + \hat{a}_{n-1} X^{n-1}$。由于NTT变换能将$R_q$中均匀随机分布的元素映射到$\mathbb{Z}_q^n$中均匀随机分布的向量，我们可以假设函数Parse输出即为某个随机元素经过NTT变换后的结果。

算法 1: $\mathsf{Parse} : \mathcal{B}^* \to R_q$

输入: 字节流$b_0, b_1, b_2, \cdots \in \mathcal{B}^*$

输出: 多项式$\hat{a} \in R_q$

$i := 0$;
$j := 0$;
while $j < n$ **do**
    $b'_{i+2} := \mathsf{Trunc}(b_{i+2}, \lceil \log_2 q \rceil - 16)$;
    $d := b_i + 256 \cdot b_{i+1} + 65536 \cdot b'_{i+2}$;
    if $d < q$ **then**
        $\hat{a}_j := d$;
        $j := j + 1$;
    end
    $i := i + 3$;
end
return $\hat{a}_0 + \hat{a}_1 X + \cdots + \hat{a}_{n-1} X^{n-1}$;

抽样集合B_{60}中的元素. 令B_h表示环R中恰好有h个系数为-1或1，且其他系数为0的元素构成的集合。显然，我们有$|B_h| = 2^h \cdot \binom{n}{h}$。经简单计算可知，$B_{60}$有超过$2^{256}$个元素，即$|B_{60}| > 2^{256}$。埃奎斯签名方案将用到能够从$B_{60}$中抽样的确定性函数，其中$n = 256$。算法2给出了SampleInBall函数的伪代码描述，该抽样算法利用Fisher-Yates洗牌算法[23]将（均匀随机）字节流映射到B_{60}的元素上。

算法 2: SampleInBall : $\mathcal{B}^* \to B_{60}$

输入: 字节流$b_0, b_1, b_2, \cdots \in \mathcal{B}^*$

输出: 多项式$c \in B_{60}$

$k := 0$;
for i从0到255 **do**
    $c_i := 0$;
end
for i从0到59 **do** /* 随机选取$s_0, s_1, \ldots, s_{59} \xleftarrow{\$} \{0,1\}$ */
    if $i \bmod 8 = 0$ **then**
        $(t_0, \ldots, t_7) :=$ BytesToBits(b_k); /* 将字节b_k转换成比特数组$t_0, \ldots, t_7$ */
        $k := k + 1$;
    end
    $s_i := t_{i \bmod 8}$;
end
for i从196到255 **do**
    while $b_k > i$ **do**
        $k := k + 1$;
    end
    $j := b_k$; /* 随机选取$j \xleftarrow{\$} \{0, 1, \ldots, i\}$ */
    $c_i := c_j$;
    $c_j := (-1)^{s_{(i-196)}}$;
    if $c_j = -1$ **then**
        $c_j = q + c_j$; /* 将c_j转换成$\mathbb{Z}_q$中的元素 */
    end
end
return $c_0 + c_1 X + \cdots + c_{255} X^{255}$;

此外，我们还需要函数Pack_c将B_{60}中的元素转换成字节数组和其逆函数Unpack_c。算法3给了函数Pack_c的定义。

算法 3: $\mathsf{Pack}_c : B_{60} \to \mathcal{B}^{40}$

输入: 多项式$c \in B_{60}$

输出: 字节数组$b = (b_0, b_1, b_2, \ldots, b_{39}) \in \mathcal{B}^{40}$

```
1  k := 1;
2  s := 0;
3  for i从0到31 do
4      b_i = 0;
5      for j从0到7 do
6          if c_{8i+j} ≠ 0 then
7              b_i := b_i + 2^j;
8              if c_{8i+j} = q − 1 then
9                  s := s + k;                    /* 存储c_{8i+j} = −1的符号 */
10             end
11             k := 2k;                                  /* 左移1位 */
12         end
13     end
14 end
15 for i从32到39 do
16     b_i := s mod 256;                          /* 用8字节存储所有符号 */
17     s := ⌊s/8⌋;
18 end
19 return (b_0, b_1, b_2, ..., b_{39});
```

随机抽样S_{γ_1-1}中的元素. 对于任意正整数η，定义集合S_η为环R中每个系数都属于$\{q-\eta,\ldots,q+\eta\}$中的多项式组成的集合。埃奎斯签名方案将使用算法4中的ExpandMask函数从环R中抽样一个环元素$y \in R_q$，使得当输入字节流是均匀随机时y的每个系数都均匀随机取自于$\{q-\gamma_1+1,\ldots,q+\gamma_1-1\}$，其中$\gamma_1 = 131072 = 2^{17}$。

算法 4: ExpandMask : $\mathcal{B}^* \to S_{\gamma_1-1}$

输入: 字节流$b_0, b_1, b_2, \cdots \in \mathcal{B}^*$

输出: 多项式$y \in S_{\gamma_1-1}$

$i := 0$;
$j := 0$;
while $j < n$ **do**
  $b'_{i+2} := \mathsf{Trunc}(b_{i+2}, 2)$;
  $c := b_i + 256 \cdot b_{i+1} + 65536 \cdot b'_{i+2}$;
  if $c \le 2\gamma_1 - 2$ **then**
    $y_j := q + \gamma_1 - 1 - c$;
    $j := j + 1$;
  end
  $b''_{i+2} := \lfloor b_{i+2}/16 \rfloor$;
  $b'_{i+4} := \mathsf{Trunc}(b_{i+4}, 6)$;
  $d := b''_{i+2} + 16 \cdot b_{i+3} + 4096 \cdot b'_{i+4}$;
  if $d \le 2\gamma_1 - 2$ **then**
    $y_j := q + \gamma_1 - 1 - d$;
    $j := j + 1$;
  end
  $i := i + 5$;
end
return $y_0 + y_1 X + \cdots + y_{n-1} X^{n-1}$;

类似地，对于任意正整数$\eta \in \mathbb{Z}$，我们可以按照算法4中的拒绝采样逻辑从任意正整数η定义的集合$S_\eta = \{-\eta,\ldots,\eta\}$中抽样均匀随机的元素。

编码与解码. 埃奎斯签名方案需要将多项式（或多项式向量）编码为字节数组以及将字节数组解码为多项式（或多项式向量）。我们定义解码函数Decode_ℓ将32ℓ字节数组解码为多项式$f = f_0 + f_1 X + \cdots + f_{255} X^{255}$，即我们只考虑$n = 256$的环，且每个系数$f_i \in \{0, 1, \ldots, 2^\ell - 1\}$。我们定义编码函数$\mathsf{Encode}_\ell$为$\mathsf{Decode}_\ell$的逆。当我们将$\mathsf{Encode}_\ell$作用到多项式向量时，其意思是将编码操作作用到向量中

的每个多项式分量，并将按顺序级联的字节数组输出（对应的Decode_ℓ函数将按顺序逐一恢复向量中的每个多项式分量）。算法5给了Decode_ℓ的伪代码描述。

算法 5: $\mathsf{Decode}_\ell : \mathcal{B}^{32\ell} \times \mathbb{Z} \to R_q$

输入: 字节数组$B \in \mathcal{B}^{32\ell}$和非负整数$u \in \mathbb{Z}$

输出: 多项式$f \in R_q$

$(\beta_0, \ldots, \beta_{256\ell-1}) := \mathsf{BytesToBits}(B)$;
for i从0到255 **do**
    $f_i := \sum_{j=0}^{\ell-1} \beta_{i\ell+j} 2^j$;
    if $u \neq 0$ **then**
        $f_i := u - f_i$;
    end
end
return $f_0 + f_1 X + \cdots + f_{255} X^{255}$;

此外，我们还需要函数Pack_h和其逆函数Unpack_h，其中函数Pack_h能将R_q^k中有至多ω个系数等于1其他系数等于0的多项式向量转换成字节数组。算法6给了函数Pack_h的定义，该定义默认环R_q的维数$n = 256$。

算法 6: $\mathsf{Pack}_h : R_q^k \times \mathbb{N} \to \mathcal{B}^{\omega+k}$

输入: 多项式向量$\mathbf{h} \in R_q^k$和正整数$\omega \in \mathbb{N}$

输出: 字节数组$b = (b_0, b_1, \ldots, b_{\omega+k-1}) \in \mathcal{B}^{\omega+k}$

$\ell := 0$;
for i从0到$k-1$ **do**
    for j从0到255 **do**
        if $\mathbf{h}[i][j] = 1$ **then** /* 向量$\mathbf{h}$的第i个多项式的第j个系数 */
            $b_\ell := j$;
            $\ell := \ell + 1$;
        end
    end
    $b_{\omega+i} := \ell$;
end
while $\ell < \omega$ **do**
    $b_\ell := 0$; /* 用0字节填充未使用的字节空间 */
    $\ell := \ell + 1$;
end
return $b = (b_0, b_1, \ldots, b_{\omega+k-1})$;

2.3 高低位比特与提示

埃奎斯签名方案将利用一些简单的算法以抽取$\mathbb{Z}_q$中元素的高低位比特[21]，其目标是给定任意的元素$r \in \mathbb{Z}_q$和另一个小的元素$z \in \mathbb{Z}_q$，我们在不存储z的情况下能恢复$r+z$的高位比特。

首先，我们定义两种不同的方法来计算$\mathbb{Z}_q$中元素的高位比特和低位比特。特别地，算法7给了一个Power2Round$_q$函数，该函数以整数$r \in \mathbb{Z}_q$和正整数$d \in \mathbb{N}$作为输入，输出整数$r_0, r_1 \in \mathbb{Z}$，使得$r_0 = r \bmod^{\pm} 2^d$且有$r = r_1 \cdot 2^d + r_0$。

算法 7: Power2Round$_q : \mathbb{Z}_q \times \mathbb{N} \to \mathbb{Z}_q \times \mathbb{Z}_q$

输入: 整数$r \in \mathbb{Z}_q$与正整数d
输出: 整数r_1和r_0

$r := r \bmod^{+} q$;
$r_0 := r \bmod^{\pm} 2^d$;
$r_1 := (r - r_0)/2^d$;
return (r_1, r_0);

如果我们选择r_1为0与$\lfloor q/2^d \rfloor$之间的非负整数，那么除了在0和q的边界情况外，任意两个数$r_1 \cdot 2^d$与$r_1' \cdot 2^d$之间的在模q意义下的距离总是$\geq 2^d$。特别地，$\lfloor q/2^d \rfloor \cdot 2^d$与0在模$q$意义下的距离可能非常小。为了处理这个问题，算法8定义了一个新的函数Decompose$_q$，该函数以整数$r \in \mathbb{Z}_q$和正整数$\alpha \in \mathbb{N}$作为输入，输出整数$r_0, r_1 \in \mathbb{Z}$，使得$r = r_1 \cdot \alpha + r_0$只有在$r - r_0 \neq q-1$时成立，其中$\alpha | (q-1)$。

算法 8: Decompose$_q : \mathbb{Z}_q \times \mathbb{N} \to \mathbb{Z}_q \times \mathbb{Z}_q$

输入: 整数$r \in \mathbb{Z}_q$和正整数α，其中$\alpha | (q-1)$
输出: 整数r_1和r_0

$r := r \bmod^{+} q$;
$r_0 := r \bmod^{\pm} \alpha$;
if $r - r_0 = q - 1$ **then**
    $r_1 := 0$;
    $r_0 := r_0 - 1$;
else
    $r_1 := (r - r_0)/\alpha$;
end
return (r_1, r_0);

为了符号描述简单，我们在算法9和算法10中定义了两个新的函数$\mathsf{HighBits}_q$和$\mathsf{LowBits}_q$，其目的是分别得到$\mathsf{Decompose}_q$的输出中高位整数r_1和低位整数r_0。

算法 9: $\mathsf{HighBits}_q : \mathbb{Z}_q \times \mathbb{N} \to \mathbb{Z}_q$

输入: 整数$r \in \mathbb{Z}_q$和α，其中$\alpha|(q-1)$

输出: 整数r_1

$(r_1, r_0) := \mathsf{Decompose}_q(r, \alpha)$;

return r_1;

算法 10: $\mathsf{LowBits}_q : \mathbb{Z}_q \times \mathbb{N} \to \mathbb{Z}_q$

输入: 整数$r \in \mathbb{Z}_q$和α，其中$\alpha|(q-1)$

输出: 整数r_0

$(r_1, r_0) := \mathsf{Decompose}_q(r, \alpha)$;

return r_0;

借助于$\mathsf{Decompose}_q$函数，算法11与算法12分别定义了生成提示函数$\mathsf{MakeHint}_q$和使用提示函数$\mathsf{UseHint}_q$。

算法 11: $\mathsf{MakeHint}_q : \mathbb{Z}_q \times \mathbb{Z}_q \times \mathbb{N} \to \{0,1\}$

输入: 元素$z \in \mathbb{Z}_q$, $r \in \mathbb{Z}_q$, 和$\alpha \in \mathbb{N}$满足$\alpha|(q-1)$

输出: 提示$h \in \{0,1\}$

$r_1 := \mathsf{HighBits}_q(r, \alpha)$;

$v_1 := \mathsf{HighBits}_q(r+z, \alpha)$;

if $r_1 \neq v_1$ **then**

    $h := 1$;

else

    $h := 0$;

end

return h;

算法 12: $\mathsf{UseHint}_q : \{0,1\} \times \mathbb{Z}_q \times \mathbb{N} \to \mathbb{Z}_q$

输入: 提示$h \in \{0,1\}$，$r \in \mathbb{Z}_q$，和$\alpha \in \mathbb{N}$满足$\alpha | (q-1)$

输出: 整数r_1

$k := (q-1)/\alpha$;

$(r_1, r_0) := \mathsf{Decompose}_q(r, \alpha)$;

if $h = 1$和$r_0 > 0$ **then**

  $r_1 := (r_1 + 1) \bmod^{+} k$;

end

if $h = 1$和$r_0 \leq 0$ **then**

  $r_1 := (r_1 - 1) \bmod^{+} k$;

end

return r_1;

当上述算法被作用到多项式或多项式向量时，其意思是对应操作被分别独立地作用到多项式的每个系数上。埃奎斯签名方案需要用到上述函数的如下性质来保证正确性和安全性。

引理 1 令正整数q和偶数α满足$q > 2\alpha$且$q \bmod \alpha = 1$。令$\mathbf{r}, \mathbf{z}$是R_q中的多项式向量，且$\|\mathbf{z}\|_\infty \leq \alpha/2$ 。令$\mathbf{h}, \mathbf{h}'$为比特向量，那么我们有如下性质成立：

- $\mathsf{UseHint}_q(\mathsf{MakeHint}_q(\mathbf{z}, \mathbf{r}, \alpha), \mathbf{r}, \alpha) = \mathsf{HighBits}_q(\mathbf{r} + \mathbf{z}, \alpha)$;
- 如果$\mathbf{v}_1 = \mathsf{UseHint}_q(\mathbf{h}, \mathbf{r}, \alpha)$，那么$\|\mathbf{r} - \mathbf{v}_1 \cdot \alpha\|_\infty \leq \alpha + 1$。进一步，如果$\mathbf{h}$至多有$\omega$个1，那么向量$\mathbf{r} - \mathbf{v}_1 \cdot \alpha$中至多有$\omega$个系数在 $mod^{\pm}$ q意义下的绝对值超过$\alpha/2$。
- 对于任意$\mathbf{h}$与$\mathbf{h}'$，如果$\mathsf{UseHint}_q(\mathbf{h}, \mathbf{r}, \alpha) = \mathsf{UseHint}_q(\mathbf{h}', \mathbf{r}, \alpha)$，那么$\mathbf{h} = \mathbf{h}'$。

引理 2 如果$\|\mathbf{s}\|_\infty \leq \beta$且$\|\mathsf{LowBits}_q(\mathbf{r}, \alpha)\|_\infty < \alpha/2 - \beta$, 那么我们有等式成立：

$$\mathsf{HighBits}_q(\mathbf{r}, \alpha) = \mathsf{HighBits}_q(\mathbf{r} + \mathbf{s}, \alpha).$$

引理1和引理2的证明可参考文献[21]。此外，我们还需要引理3和引理4来优化签名算法中的拒绝策略和简化$\mathsf{MakeHint}_q$函数，最终减少复杂耗时的$\mathsf{Decompose}_q$操作次数。

引理 3 令$(r_1, r_0) = \mathsf{Decompose}_q(r, \alpha)$, $(w_1, w_0) = \mathsf{Decompose}_q(r+s, \alpha)$，其中$r, s \in \mathbb{Z}_q$，$\alpha$是偶数且$\alpha | (q-1)$。那么我们有以下等价关系，

$$\|r_0 + s\|_\infty < \alpha/2 - \beta \Leftrightarrow \|w_0\|_\infty < \alpha/2 - \beta \wedge w_1 = r_1.$$

证明. 令$t = r \bmod^{\pm} \alpha$，$t \in (\alpha/2, \alpha/2]$。我们考虑两种情况：

- 当$r-t=q-1$时，由$\mathsf{Decompose}_q$的定义可知，我们有$r_1=0, r_0=t-1$。此时，$r+s=q+r_0+s$。显然，如果$\|r_0+s\|_\infty<\alpha/2-\beta$，那么我们总有$w_1=0=r_1$且$\|w_0\|_\infty<\alpha/2-\beta$。反之，当$r+s=w_1\alpha+w_0$时，我们有$w_0=r_0+s+q$；当$r+s\neq w_1\alpha+w_0$时，我们有$r+s-(w_0+1)=q-1\Rightarrow w_0=r_0+s$。根据$\|\cdot\|_\infty$的定义，我们都有$\|r_0+s\|_\infty=\|w_0\|_\infty\leq\alpha/2-\beta$。
- 如果$r-t\neq q-1$，我们有$r=r_1\alpha+r_0$，其中$r_0\in(\alpha/2,\alpha/2]$。显然，如果$\|r_0+s\|_\infty<\alpha/2-\beta$，那么我们有$\|w_0\|_\infty=\|r_0+s\|_\infty<\alpha/2-\beta$且$w_1=r_1$。反之，当$r+s=w_1\alpha+w_0$时，由$w_1=r_1$可得$w_0=r_0+s$；当$r+s\neq w_1\alpha+w_0$时，由$r+s-(w_0+1)=q-1$可知$w_0=r_0+s-q$。根据$\|\cdot\|_\infty$的定义，我们总有$\|r_0+s\|_\infty=\|w_0\|_\infty\leq\alpha/2-\beta$。

□

引理 4 令$(r_1,r_0)=\mathsf{Decompose}_q(r,\alpha)$，$(w_1,w_0)=\mathsf{Decompose}_q(r+s,\alpha)$，其中$r,s\in\mathbb{Z}_q$，$\alpha$是偶数且$\alpha|(q-1)$。那么我们有以下等价关系，

$$\begin{cases}\|r_0+s\|_\infty<\alpha/2 \ \ or\\ \|r_0+s\|_\infty=\alpha/2\wedge r_0+s=\alpha/2 \ \ or\\ \|r_0+s\|_\infty=\alpha/2\wedge r_0+s=-\alpha/2\wedge r_1=0\end{cases}\Leftrightarrow w_1=r_1.$$

证明. 令$t=r \bmod^{\pm}\alpha$，$t\in(\alpha/2,\alpha/2]$。如引理3证明一样，我们考虑两种情况：

- 如果$r-t=q-1$，我们有$r_1=0, r_0=t-1$。此时，$r+s=q+r_0+s$。显然，如果$\|r_0+s\|_\infty\leq\alpha/2$，通过分析$r_0+s$的符号并由$\mathsf{Decompose}_q$的定义可知，我们总有$w_1=0=r_1$。反之，当$\|r_0+s\|_\infty>\alpha/2$时，我们必有$w_1\neq r_1$。
- 如果$r-t\neq q-1$，我们有$r=r_1\alpha+r_0$，其中$r_0\in(\alpha/2,\alpha/2]$。显然，如果$\|r_0+s\|_\infty<\alpha/2$，或$\|r_0+s\|_\infty=\alpha/2\wedge r_0+s=\alpha/2$，通过分析$r_0+s$的符号并由$\mathsf{Decompose}_q$的定义可知，我们总有$w_1=r_1$。反之，当$\|r_0+s\|_\infty>\alpha/2$，或$\|r_0+s\|_\infty=\alpha/2\wedge r_0+s=-\alpha/2$时，我们必有$w_1\neq r_1$。

□

注意在算法14的第22行、27行和32行我们总共需要计算4次$\mathsf{Decompose}_q$操作。利用引理3和引理4并对应修改第27行、28行，以及32行的$\mathsf{MakeHint}_q$函数，我们可以将$\mathsf{Decompose}_q$操作降低为1次。

3 埃奎斯数字签名方案

埃奎斯数字签名方案(Aigis-sig)本质上是基于文献[10,21]中的数字签名方案，其主要区别在于底层使用了不同的困难问题和不同的参数选取方式。埃奎斯签

名方案需要由12个整数参数$n, q, k, \ell, d, \omega, \eta_1, \eta_2, \beta_1, \beta_2, \gamma_1, \gamma_2$来实例化。第3.2节将给出方案的具体参数选择。令

$$d_{t_1} = \lceil \log_2 q \rceil - d, \qquad d_{\eta_1} = \lceil \log_2(2\eta_1 + 1) \rceil, \quad d_{\eta_2} = \lceil \log_2(2\eta_2 + 1) \rceil,$$
$$d_z = \lceil \log_2(2(\gamma_1 - \beta_1) - 1) \rceil, \quad d_{w_1} = \lceil \log_2(\tfrac{q-1}{2\gamma_2} - 1) \rceil.$$

我们将在算法13中给出密钥生成算法Aigis-sig.KeyGen()，在算法14中给出签名生成算法Aigis-sig.Sign()，在算法15中给出签名验证算法Aigis-sig.Verify()。为了算法描述简单，我们定义杂凑函数$\mathsf{H} : \mathcal{B}^{48} \times R_q^k \to B_{60}$如下：

$$\mathsf{H}(\mu, \mathbf{w}_1) := \mathsf{SampleInBall}\big(\mathsf{XOF}_2(\mu \| \mathsf{Encode}_{d_{w_1}}(\mathbf{w}_1, 0))\big)$$

算法 13: Aigis-sig.KeyGen()：密钥生成算法

输出: 公钥$pk \in \mathcal{B}^{32+d_{t_1} \cdot k \cdot n/8}$

输出: 私钥$sk \in \mathcal{B}^{112+(d_{\eta_1} \cdot \ell + d_{\eta_2} \cdot k + d \cdot k) \cdot n/8}$

$\rho \xleftarrow{\$} \mathcal{B}^{32}$;

$K \xleftarrow{\$} \mathcal{B}^{32}$;

$(\mathbf{s}_1, \mathbf{s}_2) \xleftarrow{\$} S_{\eta_1}^{\ell} \times S_{\eta_2}^{k}$;

for i从0到$k-1$ **do** /* 生成矩阵$\hat{\mathbf{A}} = \mathsf{NTT}(\mathbf{A}) \in R_q^{k \times \ell}$ */

  for j从0到$\ell - 1$ **do**

    $\hat{\mathbf{A}}[i][j] := \mathsf{Parse}(\mathsf{XOF}_1(\rho \| j \| i))$

  end

end

$\hat{\mathbf{s}}_1 := \mathsf{NTT}(\mathbf{s}_1)$;

$\mathbf{t} := \mathsf{NTT}^{-1}(\hat{\mathbf{A}} \circ \hat{\mathbf{s}}_1) + \mathbf{s}_2$; /* $\mathbf{t} := \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$ */

$(\mathbf{t}_1, \mathbf{t}_0) := \mathsf{Power2Round}_q(\mathbf{t}, d)$;

$pk := \big(\rho \| \mathsf{Encode}_{d_{t_1}}(\mathbf{t}_1, 0)\big)$;

$tr := \mathsf{CRH}(pk) \in \mathcal{B}^{48}$;

$sk :=$
$\big(\rho \| K \| tr \| \mathsf{Encode}_{d_{\eta_1}}(\mathbf{s}_1, q + \eta_1) \| \mathsf{Encode}_{d_{\eta_2}}(\mathbf{s}_2, q + \eta_2) \| \mathsf{Encode}_d(\mathbf{t}_0, q + 2^{d-1})\big)$;

return (pk, sk);

算法 14: Aigis-sig.Sign(sk, M)：签名生成算法

输入: 私钥$sk \in \mathcal{B}^{112+(d_{\eta_1}\cdot\ell+d_{\eta_2}\cdot k+d\cdot k)\cdot n/8}$和消息$M \in \mathcal{B}^*$

输出: 签名$\sigma \in \mathcal{B}^{(d_z\cdot\ell+1)\cdot n/8+\omega+k+8}$

$\rho := sk$;
for i从0到$k-1$ **do** /* 生成矩阵$\hat{\mathbf{A}} = \mathsf{NTT}(\mathbf{A}) \in R_q^{k\times\ell}$ */
  for j从0到$\ell-1$ **do**
    $\hat{\mathbf{A}}[i][j] := \mathsf{Parse}(\mathsf{XOF}_1(\rho\|j\|i))$
  end
end
$K := sk + 32$; $tr := sk + 64$;
$\mathbf{s}_1 := \mathsf{Decode}_{d_{\eta_1}}(sk + 112, q + \eta_1)$;
$\hat{\mathbf{s}}_1 := \mathsf{NTT}(\mathbf{s}_1)$;
$\mathbf{s}_2 := \mathsf{Decode}_{d_{\eta_2}}(sk + 112 + d_{\eta_1}\cdot\ell\cdot n/8, q + \eta_2)$;
$\hat{\mathbf{s}}_2 := \mathsf{NTT}(\mathbf{s}_2)$;
$\mathbf{t}_0 := \mathsf{Decode}_d(sk + 112 + (d_{\eta_1}\cdot\ell + d_{\eta_2}\cdot k)\cdot n/8, q + 2^{d-1})$;
$\hat{\mathbf{t}}_0 := \mathsf{NTT}(\mathbf{t}_0)$;
$\mu := \mathsf{CRH}(tr\|M) \in \mathcal{B}^{48}$;
$N := 0$; $(\mathbf{z}, \mathbf{h}) := \perp$;
while $(\mathbf{z}, \mathbf{h}) := \perp$ **do** /* 使用拒绝采样技术产生签名 */
  for i从0到$\ell-1$ **do** /* 生成随机向量$\mathbf{y} \in S^{\ell}_{\gamma_1-1}$ */
    $\mathbf{y}[i] := \mathsf{ExpandMask}(\mathsf{XOF}_2(K\|\mu\|N))$;
    $N := N + 1$;
  end
  $\mathbf{w} := \mathsf{NTT}^{-1}(\hat{\mathbf{A}} \circ \mathsf{NTT}(\mathbf{y}))$; /* $\mathbf{w} := \mathbf{Ay}$ */
  $\mathbf{w}_1 := \mathsf{HighBits}_q(\mathbf{w}, 2\gamma_2)$;
  $c := \mathsf{H}(\mu, \mathbf{w}_1)$;
  $\hat{c} := \mathsf{NTT}(c)$;
  $\mathbf{z} := \mathbf{y} + \mathsf{NTT}^{-1}(\hat{c} \circ \hat{\mathbf{s}}_1)$; /* $\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$ */
  $\mathbf{u} := \mathbf{w} - \mathsf{NTT}^{-1}(\hat{c} \circ \hat{\mathbf{s}}_2)$; /* $\mathbf{u} := \mathbf{w} - c\mathbf{s}_2$ */
  $(\mathbf{r}_1, \mathbf{r}_0) := \mathsf{Decompose}_q(\mathbf{u}, 2\gamma_2)$;
  if $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta_1$或$\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta_2$或$\mathbf{r}_1 \neq \mathbf{w}_1$ **then**
    $(\mathbf{z}, \mathbf{h}) := \perp$;
  else
    $\mathbf{v} := \mathsf{NTT}^{-1}(\hat{c} \circ \hat{\mathbf{t}}_0)$; /* $\mathbf{v} := c\mathbf{t}_0$ */
    $\mathbf{h} := \mathsf{MakeHint}_q(-\mathbf{v}, \mathbf{u} + \mathbf{v}, 2\gamma_2)$;
    if $\|\mathbf{v}\|_\infty \geq \gamma_2$或$\mathcal{HW}(\mathbf{h}) > \omega$ **then**
      $(\mathbf{z}, \mathbf{h}) := \perp$;
    end
  end
end
return $\sigma = \big(\mathsf{Encode}_{d_z}(\mathbf{z}, q + \gamma_1 - \beta_1 - 1)\|\mathsf{Pack}_h(\mathbf{h}, \omega)\|\mathsf{Pack}_c(c)\big)$; /* $\sigma = (\mathbf{z}, \mathbf{h}, c)$ */

算法 15: Aigis-sig.Verify(pk, M, σ)：签名验证算法

输入: 公钥$pk \in \mathcal{B}^{32+d_{t_1} \cdot k \cdot n/8}$和消息$M \in \mathcal{B}^*$

输入: 签名$\sigma \in \mathcal{B}^{(d_z \cdot \ell+1) \cdot n/8+\omega+k+8}$

输出: 验证结果$b \in \{0, 1\}$

$\rho := pk$;

$\mathbf{t}_1 := \mathsf{Decode}_{d_{t_1}}(pk + 32)$;

$\mathbf{z} := \mathsf{Decode}_{d_z}(\sigma, q + \gamma_1 - \beta_1 - 1)$;

$\mathbf{h} := \mathsf{Unpack}_h(\sigma + d_z \cdot \ell \cdot n/8, \omega)$;

$c := \mathsf{Unpack}_c(\sigma + (d_z \cdot \ell + 1) \cdot n/8 + 8)$;

for i从0到$k-1$ **do** /* 生成矩阵$\hat{\mathbf{A}} = \mathsf{NTT}(\mathbf{A}) \in R_q^{k \times \ell}$ */

  for j从0到$\ell - 1$ **do**

    $\hat{\mathbf{A}}[i][j] := \mathsf{Parse}(\mathsf{XOF}_1(\rho \| j \| i))$

  end

end

$\mu := \mathsf{CRH}(\mathsf{CRH}(pk) \| M) \in \mathcal{B}^{48}$;

$\mathbf{u} := \mathsf{NTT}^{-1}(\hat{\mathbf{A}} \circ \mathsf{NTT}(\mathbf{z}) - \mathsf{NTT}(c) \circ \mathsf{NTT}(\mathbf{t}_1 \cdot 2^d))$; /* $\mathbf{u} := \mathbf{Az} - c\mathbf{t}_1 2^d$ */

$\mathbf{w}'_1 := \mathsf{UseHints}_q(\mathbf{h}, \mathbf{u}, 2\gamma_2)$;

$c' := \mathsf{H}(\mu, \mathbf{w}'_1)$;

if $\|\mathbf{z}\|_\infty < \gamma_1 - \beta_1$且$c = c'$且$\mathcal{HW}(\mathbf{h}) \leq \omega$ **then**

  $b := 1$; /* 签名验证通过 */

else

  $b := 0$; /* 签名验证失败 */

end

return b;

正确性分析. 如果$\|c\mathbf{t}_0\|_\infty < \gamma_2$，那么根据引理1我们有以下成立：

$$\mathsf{UseHint}_q(\mathbf{h}, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0, 2\gamma_2) = \mathsf{HighBits}_q(\mathbf{w} - c\mathbf{s}_2, 2\gamma_2)$$

因为$\mathbf{w} = \mathbf{Ay}$和$\mathbf{t} = \mathbf{As}_1 + \mathbf{s}_2$，所以我们有以下等式成立：

$$\mathbf{w} - c\mathbf{s}_2 = \mathbf{Ay} - c\mathbf{s}_2 = \mathbf{A}(\mathbf{z} - c\mathbf{s}_1) - c\mathbf{s}_2 = \mathbf{Az} - c\mathbf{t} \tag{1}$$

$$\mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0 = \mathbf{Az} - c\mathbf{t}_1 \cdot 2^d \tag{2}$$

因此，签名验证算法计算的值满足以下等式：

$$\mathsf{UseHint}_q(\mathbf{h}, \mathbf{Az} - c\mathbf{t}_1 \cdot 2^d, 2\gamma_2) = \mathsf{HighBits}_q(\mathbf{w} - c\mathbf{s}_2, 2\gamma_2)$$

因为签名生成算法检查了$\mathbf{r}_1 = \mathbf{w}_1$成立，所以我们有以下等式成立：

$$\mathsf{HighBits}_q(\mathbf{w} - c\mathbf{s}_2, 2\gamma_2) = \mathsf{HighBits}_q(\mathbf{w}, 2\gamma_2)$$

从而，签名验证算法与签名生成算法计算的$\mathbf{w}_1$是相同的，使得签名验证算法总是接受签名生成算法计算的签名。

3.1 签名生成算法的计算复杂度

埃奎斯签名方案的签名算法Aigis-sig.Sign(sk, m)采用了拒绝采样技术[30,31]来生成$(\mathbf{z}, \mathbf{h})$，其效率主要由第16步中while循环的次数来决定。接下来，我们将计算 $(\mathbf{z}, \mathbf{h}) \neq \perp$的概率。首先，如果$\|c\mathbf{s}_1\|_\infty \leq \beta_1$成立，那么当$\|\mathbf{y}\|_\infty \leq \gamma_1 - 2\beta_1 - 1$时，我们总有$\|\mathbf{z}\|_\infty \leq \gamma_1 - \beta_1 - 1$。该范围的大小为$2(\gamma_1 - \beta_1) - 1$。由于$\mathbf{y}$中每个多项式系数都是从$2\gamma_1 - 1$个元素中均匀取值（即对于任意固定的$c\mathbf{s}_1$，向量$\mathbf{z} = \mathbf{y} + c\mathbf{s}_1$中每个多项式的系数都有$2\gamma_1 - 1$种可能值），因此$\|\mathbf{z}\|_\infty \leq \gamma_1 - \beta_1 - 1$的概率是

$$\left(\frac{2(\gamma_1 - \beta_1) - 1}{2\gamma_1 - 1}\right)^{n\cdot\ell} = \left(1 - \frac{\beta_1}{\gamma_1 - 1/2}\right)^{n\cdot\ell} \approx e^{-n\ell\beta_1/\gamma_1}, \tag{3}$$

以上近似关系利用了γ_1取值远大于$1/2$的事实。

现在，我们计算不等式$\|\mathbf{r}_0\|_\infty = \|\mathsf{LowBits}_q(\mathbf{w} - c\mathbf{s}_2, 2\gamma_2)\|_\infty < \gamma_2 - \beta_2$成立的概率。假设$\mathbf{r}_0$中多项式的每个系数都服从在模$2\gamma_2$的剩余系中的均匀分布，那么$\|\mathbf{r}_0\|_\infty < \gamma_2 - \beta_2$的概率为

$$\left(\frac{2(\gamma_2 - \beta_2) - 1}{2\gamma_2}\right)^{n\cdot k} \approx e^{-nk\beta_2/\gamma_2}$$

以上近似关系利用了β_2取值远大于$1/2$的事实。

根据引理2，如果$\|c\mathbf{s}_2\|_\infty \leq \beta_2$，那么$\|\mathbf{r}_0\|_\infty < \gamma_2 - \beta_2$蕴含了$\mathbf{r}_1 = \mathbf{w}_1$。因此，如果前两个检查通过，那么最后一个检查能以$1 - \mathsf{negl}(\kappa)$的概率成功。从而，第28步中每次检查使得$(\mathbf{z}, \mathbf{h}) \neq \perp$的概率约为

$$\approx e^{-n(\ell\beta_1/\gamma_1 + k\beta_2/\gamma_2)} \tag{4}$$

此外，在我们的参数设置下，第33步判断导致$(\mathbf{z}, \mathbf{h}) = \perp$的概率小于1%。因此，大部分循环将是由第28步引起。

3.2 参数集以及相关函数的实例化

根据当前格上困难问题求解状态和未来数年内对抗量子安全签名方案的需求，我们为埃奎斯签名方案选择了四组参数集，即PARAMS I、PARAMS II、

PARAMS II-b和PARAMS III，分别瞄准目标量子安全强度80、128、128和160（对应保守估计的经典安全强度分别约为98、141、141和178），其中PARAMS II为推荐参数。四组参数对应签名算法中while循环的期望次数分别约为5.86、7.61、11.7和6.67。

表 1. 埃奎斯签名方案参数集(公钥、私钥和签名的长度单位为字节)

参数集名称	(n,k,ℓ,q,d,ω)	(η_1,η_2)	(β_1,β_2)	(γ_1,γ_2)	公钥$\|pk\|$	私钥$\|sk\|$	签名$\|\sigma\|$
PARAMS I	$(256,4,3,2021377,13,80)$	$(2,3)$	$(120,175)$	$(131072,168448)$	1056	2448	1852
PARAMS II	$(256,5,4,3870721,14,96)$	$(2,5)$	$(120,275)$	$(131072,322560)$	1312	3376	2445
PARAMS II-b	$(256,5,4,3870721,14,96)$	$(3,5)$	$(175,275)$	$(131072,322560)$	1312	3376	2445
PARAMS III	$(256,6,5,3870721,14,120)$	$(1,5)$	$(60,275)$	$(131072,322560)$	1568	3888	3046

实例化XOF和CRH. 我们采用FIPS-202标准[37]中的函数来实例化这些对称密码组件：

– 用SHAKE-128或AES实例化XOF_1（默认使用AES）
– 用SHAKE-256或AES实例化XOF_2（默认使用AES）
– 用SHAKE-256实例化CRH，其中SHAKE-256输出的前48字节作为CRH的输出。

4 程序实现及性能

我们在Windows 10 64位操作系统（硬件配置为3.4GHz的Intel Core-i7 6700 CPU和4GB内存的ThinkCentre台式机）上实现了埃奎斯数字签名方案，并用AVX2部分优化程序实现。表2给出了埃奎斯数字签名方案各算法在Windows 10系统上运行10000次的平均CPU周期。

表 2. 埃奎斯签名Win10版本的计算效率（单位：CPU周期）

参数集名称	密钥生成(AVX2)	生成签名(AVX2)	签名验证(AVX2)
PARAMS I	75 216	296 716	78 841
PARAMS II	112 362	459 903	104 337
PARAMS II-b	102 852	624 031	104 488
PARAMS III	140 563	533 880	136 503

4.1 编译和运行程序

用VS 2017或更新的版本打开VS项目文件即可编译并运行程序。默认情况下，编译程序将会生成配置为PARAMS II中参数的数字签名方案的软件。如果

想要编译程序生成配置为PARAMS I、PARAMS II-b或PARAMS III中参数的数字签名方案的软件，只需要打开名为params.h的头文件，并对应修改PARAMS标志为1、22或者3即可。

5 可证明安全

5.1 困难假设

LWE问题. 令n, q是任意正整数，α是任意正实数，χ_α是$\mathbb{Z}$上以α为参数的离散分布$\chi_\alpha \subseteq \mathbb{Z}$。对于向量$\mathbf{s} \in \mathbb{Z}_q^n$，定义分布$A_{\mathbf{s},\alpha}$如下：

$$A_{\mathbf{s},\alpha} = \{(\mathbf{a}, b = \mathbf{a}^T\mathbf{s} + e \bmod q) : \mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n, e \xleftarrow{\$} \chi_\alpha\}.$$

当随机均匀地选取秘密向量$\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$时，计算性LWE问题的目标是在给定分布$A_{\mathbf{s},\alpha}$中任意多项式个样本的条件下计算出秘密向量$\mathbf{s} \in \mathbb{Z}_q^n$。而对应的判定性LWE问题的目标则是在给定任意多项式个样本的条件下区分分布$A_{\mathbf{s},\alpha}$和$\mathbb{Z}_q^n \times \mathbb{Z}_q$上的均匀分布。对于满足某些条件的$q$和分布，判定性LWE问题在平均情况下的困难性在多项式时间的意义下等价于计算性LWE问题在最坏情况下的困难性[41,38,7]。当分布χ_α是以α为标准差的高斯分布时，Regev证明了相应的$\mathrm{LWE}_{n,q,\alpha}$问题在平均情况下的困难性可以量子归约到格上某些问题在最坏情况下的困难性[41]。此后，对于某些特定的参数，Peikert[38]给出了$\mathrm{LWE}_{n,q,\alpha}$到格上困难问题的经典归约。特别地，结合文献[41,39,26]的结论，我们有如下命题成立：

命题 1 *令实数$\alpha = \alpha(n) \in (0,1)$和素数$q = q(n)$满足条件$\alpha q > 2\sqrt{n}$。如果存在多项式时间的（量子）算法求解$\mathrm{LWE}_{n,q,\alpha q\sqrt{2}}$问题，那么存在多项式时间的量子算法求解秩为$n$的格上近似因子为$\gamma = \tilde{O}(n/\alpha)$的最坏情况下的$\mathrm{SIVP}_\gamma$问题。*

由SIVP_γ的困难性可知，对于任意常数$\epsilon < 1/2$和实数$\alpha = 2^{-n^\epsilon}$，$\mathrm{LWE}_{n,q,\alpha q\sqrt{2}}$仍然是困难的。此外，当秘密元素$\mathbf{s}$并不是随机均匀地选自于$\mathbb{Z}_q^n$时，LWE问题也可能是非常困难的。特别地，当$\mathbf{s} \xleftarrow{\$} \chi_\alpha^n$时，相应的$\mathrm{LWE}_{n,q,\alpha}$问题至少和标准的LWE问题是一样困难的[7,32]。事实上，这类变种的问题称为LWE的正规形，在全同态加密中常常被用来控制错误项的增长[17,15]。

SIS问题. 最小整数解问题(SIS问题)最早由密码学家Ajtai[2]开始研究，但其正式的定义却是Regev和Micciancio提出的[34,35]。正式地，给定正整数$n, m, q \in \mathbb{Z}$、随机矩阵$\mathbf{A} \in \mathbb{Z}_q^{n\times m}$和正实数$\beta \in \mathbb{R}$，正规形（无穷范数）最小整数解问题$\mathrm{SIS}_{n,q,m,\beta}^\infty$的目标是寻找非零向量$\mathbf{x} \in \mathbb{Z}^{n+m}\backslash\{\mathbf{0}\}$使得$(\mathbf{I}_n\|\mathbf{A})\mathbf{x} = \mathbf{0} \bmod q$且$\|\mathbf{x}\|_\infty \le \beta$。

注意到标准的SIS问题是给定随机矩阵$\mathbf{A} \in \mathbb{Z}_q^{n\times m}$，计算$\mathbf{A}\mathbf{x} = \mathbf{0} \bmod q$。上述正规形SIS问题相当于标准SIS问题定义中的问题实例为$\mathbf{A}' = (\mathbf{I}_n\|\mathbf{A}) \in \mathbb{Z}_q^{n\times(n+m)}$。

这两种定义看似存在差异，然而当m足够大时，这两种定义在多项式时间的意义下是等价的。特别地，当m足够大时，对于随机选择的矩阵$\mathbf{A}' \in \mathbb{Z}_q^{n\times(n+m)}$，我们以很大的概率有$\mathbf{A}'$中存在$n$个线性无关的列。不妨设$\mathbf{A}'$的前$n$列线性无关（否则我们可以通过列变换将线性无关的$n$列置换到前$n$列），且前$n$列组成的矩阵为$\mathbf{A}_1 \in \mathbb{Z}_q^{n\times n}$，即$\mathbf{A} = (\mathbf{A}_1 \| \mathbf{A}_2) \in \mathbb{Z}_q^{n\times(n\times m)}$。在这种情况下，通过在等式两边同时乘以$\mathbf{A}_1^{-1}$，我们有$\mathbf{A}'\mathbf{x} = \mathbf{0} \bmod q$和$(\mathbf{I}_n \| \mathbf{A}_1^{-1}\mathbf{A}_2)\mathbf{x} = \mathbf{0} \bmod q$的解空间完全相同，而后者恰好对应着正规形SIS问题。为了便于使用，我们直接采用正规形SIS问题的定义。

对于适当选取的参数，当随机均匀地选取矩阵$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$时，上述两个问题在平均情况下的困难性与格上问题最坏情况的困难性是等价的。

命题 2 ([24]) *对于正整数n，多项式界定的$m, \beta = \mathsf{poly}(n)$和素数$q \geq \beta \cdot \omega(\log n)$，问题$\mathrm{SIS}^{\infty}_{n,m,q,\beta}$在平均情况下的困难性与秩为$n$的格上近似因子$\gamma = \beta \cdot O(\sqrt{\log n})$的$\mathrm{SIVP}_{\gamma}$问题在最坏情况下的困难性是一样的。*

RLWE问题. 令整数n是2的幂次，素数q满足$q = 1 \bmod 2n$，环$R_q = \mathbb{Z}_q[x]/(x^n+1)$。对于任意环元素$s \in R_q$和实数$\alpha$，定义分布$B_{s,\alpha}$如下：

$$B_{s,\alpha} = \{(a, b = as + e) : a \xleftarrow{\$} R_q, e \xleftarrow{\$} \chi_\alpha^n\}.$$

对于随机均匀选取的秘密元素$s \xleftarrow{\$} R_q$和任意多项式界定的正整数$\ell$，RLWE问题$\mathrm{RLWE}_{n,q,\ell,\alpha}$的目标是在给定$\ell$个样本的条件下区分分布$B_{s,\alpha}$和$R_q \times R_q$上的均匀分布。对于适当选取的参数，我们有如下结论成立：

命题 3 ([32, 定理3.6]) *令正整数n是2的幂次，$\alpha \in (0,1)$是一个实数，q是一个素数，定义$R = \mathbb{Z}[x]/(x^n+1)$。如果$q = 1 \bmod 2n$且$\beta q > \omega(\sqrt{\log n})$，那么存在从环$R$的理想格上最坏情况的$\mathrm{SIVP}_{\tilde{O}(\sqrt{n}/\beta)}$问题到平均情况的$\mathrm{RLWE}_{n,q,\ell,\alpha}$问题的多项式时间的量子归约，其中$\alpha = \beta q \cdot (n\ell/\log(n\ell))^{1/4}$。*

类似地，当秘密元素$s \xleftarrow{\$} \chi_\alpha^n$时，相应的$\mathrm{RLWE}_{n,q,\alpha,\ell}$问题至少和环上标准的RLWE问题是一样困难的[7,32]。

RSIS问题. 令整数n是2的幂次，素数q满足$q = 1 \bmod 2n$，环$R_q = \mathbb{Z}_q[x]/(x^n+1)$。给定正整数$m \in \mathbb{Z}$、随机向量$\mathbf{a} \in R_q^m$和正实数$\beta \in \mathbb{R}$，环$R_q$上正规形（无穷范数）最小整数解问题(RSIS问题)$\mathrm{RSIS}^{\infty}_{n,q,m,\beta}$的目标是寻找非零环向量$\mathbf{x} \in R_q^{1+m} \backslash \{\mathbf{0}\}$使得$(1, \mathbf{a}^T)\mathbf{x} = \mathbf{0} \bmod q$且$\|\mathbf{x}\|_\infty \leq \beta$。

MLWE问题. RLWE具有更好的结构，从而使得基于RLWE问题设计的密码方案无论是运行速度还是密钥/密文大小都具有较好的效率表现，但参数选择往往比

较受限；而LWE问题的参数选择则具有较高的灵活性。综合考虑安全性和效率，文献[15,1]提出了标准 LWE问题和RLWE问题的结合版本–模LWE问题(MLWE)。特别地，令整数n是2的幂次，素数q满足$q = 1 \bmod 2n$，环$R_q = \mathbb{Z}_q[x]/(x^n + 1)$。对于任意正整数$k, \ell \geq 1$，正实数$\alpha$，判定性MLWE问题$\mathrm{MLWE}_{n,q,k,\ell,\alpha}$的目标是将样本$(\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e})$和选自于$R_q^{k\times\ell} \times R_q^k$上均匀分布的元组区分开，其中$\mathbf{A} \xleftarrow{\$} R_q^{k\times\ell}, \mathbf{s} \xleftarrow{\$} (\chi_\alpha^n)^\ell, \mathbf{e} \xleftarrow{\$} (\chi_\alpha^n)^k$。显然，当$R_q = \mathbb{Z}_q$（即$n = 1$）时，MLWE问题就是标准的LWE问题，而当$\ell = 1$时，MLWE问题就是标准的RLWE问题。因此，研究者们倾向于相信MLWE问题的困难性介于标准LWE问题与RLWE问题之间。特别地，目前所有求解LWE和RLWE问题的算法在求解MLWE时并没有明显的优势，事实上，目前最好的求解算法都没有用到RLWE和MLWE问题的环结构。

MSIS问题. 令整数n是2的幂次，素数q满足$q = 1 \bmod 2n$，环$R_q = \mathbb{Z}_q[x]/(x^n + 1)$。给定正整数$k, \ell \in \mathbb{Z}$、随机矩阵$\mathbf{A} \in R_q^{k\times\ell}$和正实数$\beta \in \mathbb{R}$，环$R_q$上正规形（无穷范数）MSIS问题$\mathrm{MSIS}_{n,q,k,\ell,\beta}^{\infty}$的目标是寻找非零向量$\mathbf{x} \in R_q^{k+\ell}\backslash\{\mathbf{0}\}$使得其满足$(\mathbf{I}_k\|\mathbf{A})\mathbf{x} = \mathbf{0} \bmod q$且$\|\mathbf{x}\|_\infty \leq \beta$。显然，当$R_q = \mathbb{Z}_q$（即$n = 1$）时，MSIS问题就是标准的SIS问题，而当$k = 1$时，MSIS问题就是标准的RSIS问题。因此，研究者们倾向于相信MSIS问题的困难性介于标准SIS问题与RSIS问题之间。特别地，目前所有求解SIS和RSIS问题算法在求解MSIS时并没有明显的优势，事实上，目前最好的求解算法都没有用到RSIS和MSIS问题的环结构。

ALWE问题. 令n, q是任意正整数，α_1, α_2是任意正实数，$\chi_{\alpha_1}, \chi_{\alpha_2}$是$\mathbb{Z}$上以$\alpha_1, \alpha_2$为参数的离散分布。对于向量$\mathbf{s} \xleftarrow{\$} \chi_{\alpha_1}^n$，定义非对称LWE(ALWE)分布$A_{\mathbf{s},\alpha_2}$如下：

$$A_{\mathbf{s},\alpha_2} = \{(\mathbf{a}, b = \mathbf{a}^T\mathbf{s} + e \bmod q) : \mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n, e \xleftarrow{\$} \chi_{\alpha_2}\}.$$

其中非对称LWE是由正规形LWE演变而来。在正规形LWE中，向量$\mathbf{s}$的各分量和错误e取自同一分布，而在非对称LWE分布中，秘密向量$\mathbf{s}$各分量和错误e可取自参数不同的（相同）分布。判定性ALWE问题$\mathrm{ALWE}_{n,q,\ell,\alpha_1,\alpha_2}$是指在$\mathbf{s} \in \chi_{\alpha_1}^n$给定且有$\ell$个样本的条件下区分分布$A_{\mathbf{s},\alpha_2}$和 $\mathbb{Z}_q^n \times \mathbb{Z}_q$上的均匀分布。计算性ALWE问题$\mathrm{ALWE}_{n,q,\ell,\alpha_1,\alpha_2}$ 则是指在ℓ个样本的条件下计算出秘密向量$\mathbf{s} \in \chi_{\alpha_1}^n$。

ASIS问题. 给定正整数$n, m_1, m_2 \in \mathbb{Z}$、随机矩阵$\mathbf{A} \in \mathbb{Z}_q^{n\times(m_1+m_2-n)}$和正实数$\beta_1$, $\beta_2 \in \mathbb{R}$，正规形（无穷范数）ASIS问题$\mathrm{ASIS}_{n,q,m_1,m_2,\beta_1,\beta_2}^{\infty}$的目标是寻找非零向量$\mathbf{x} \in \mathbb{Z}^{m_1+m_2}\backslash\{\mathbf{0}\}$使得其满足$(\mathbf{I}_n\|\mathbf{A})\mathbf{x} = \mathbf{0} \bmod q$，$\|\mathbf{x}_1\|_\infty \leq \beta_1$且$\|\mathbf{x}_2\|_\infty \leq \beta_2$，其中$\mathbf{x}^T = (\mathbf{x}_1^T, \mathbf{x}_2^T), \mathbf{x}_1 \in \mathbb{Z}^{m_1}, \mathbf{x}_2 \in \mathbb{Z}^{m_2}$。显然，当$\beta_1 = \beta_2$时，ASIS问题就退化为SIS问题。

AMLWE问题. 结合密码方案的设计和目前针对(M/R)LWE求解算法的现状，我们提出非对称AMLWE问题（类似地还可以定义ARLWE问题）。特别地，令整数n是2的幂次，素数q满足$q = 1 \bmod 2n$，环$R_q = \mathbb{Z}_q[x]/(x^n+1)$。设$k, \ell \geq 1$为正整数，$\alpha_1, \alpha_2$为正实数。判定性AMLWE问题$\mathrm{AMLWE}_{n,q,k,\ell,\alpha_1,\alpha_2}$的目标是将样本$(\mathbf{A}, \mathbf{b} = \mathbf{As}+\mathbf{e})$和选自于$R_q^{k\times\ell} \times R_q^k$上均匀分布的元组区分开，其中$\mathbf{A} \xleftarrow{\$} R_q^{k\times\ell}, \mathbf{s} \xleftarrow{\$} (\chi_{\alpha_1}^n)^\ell, \mathbf{e} \xleftarrow{\$} (\chi_{\alpha_2}^n)^k$。计算性AMLWE问题$\mathrm{AMLWE}_{n,q,k,\ell,\alpha_1,\alpha_2}$的目标是给定样本$(\mathbf{A}, \mathbf{b} = \mathbf{As}+\mathbf{e}) \in R_q^{k\times\ell} \times R_q^k$，输出环向量$\mathbf{s} \in R_q^\ell$，其中$\mathbf{A} \xleftarrow{\$} R_q^{k\times\ell}, \mathbf{s} \xleftarrow{\$} (\chi_{\alpha_1}^n)^\ell, \mathbf{e} \xleftarrow{\$} (\chi_{\alpha_2}^n)^k$。显然，当$\alpha_1 = \alpha_2$时，AMLWE问题就退化为标准的MLWE问题。此外，当χ_α为高斯分布时，我们还可以利用高斯分布的性质证明如下困难关系在多项式时间的意义下成立：

$$\mathrm{MLWE}_{n,q,k,\ell,\min(\alpha_1,\alpha_2)} \leq \mathrm{AMLWE}_{n,q,k,\ell,\alpha_1,\alpha_2} \leq \mathrm{MLWE}_{n,q,k,\ell,\max(\alpha_1,\alpha_2)}.$$

换句话说，只要选择合适的参数，我们总能够保证AMLWE问题是难于求解的。在第6节中，我们将考虑MLWE问题的已有攻击算法及它们的变形来估计AMLWE问题的具体求解复杂度。

AMSIS问题. 结合密码方案的设计和目前针对(M/R)SIS问题求解算法的现状，我们提出非对称AMSIS问题（类似地可以定义ARSIS问题）。正式地，令整数n是2的幂次，素数q满足$q = 1 \bmod 2n$，环$R_q = \mathbb{Z}_q[x]/(x^n+1)$。给定正整数$k, \ell_1, \ell_2 \in \mathbb{Z}$、随机矩阵$\mathbf{A} \in R_q^{k\times(\ell_1+\ell_2-k)}$和正实数$\beta_1, \beta_2 \in \mathbb{R}$，环$R_q$上正规形（无穷范数）AMSIS问题$\mathrm{AMSIS}^\infty_{n,q,k,\ell_1,\ell_2,\beta_1,\beta_2}$的目标是寻找向量$\mathbf{x} \in R_q^{\ell_1+\ell_2}\backslash\{\mathbf{0}\}$使得$(\mathbf{I}_k\|\mathbf{A})\mathbf{x} = \mathbf{0} \bmod q$，$\|\mathbf{x}_1\|_\infty \leq \beta_1$且$\|\mathbf{x}_2\|_\infty \leq \beta_2$，其中$\mathbf{x} = \begin{pmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{pmatrix} \in R_q^{\ell_1+\ell_2}, \mathbf{x}_1 \in R_q^{\ell_1}, \mathbf{x}_2 \in R_q^{\ell_2}$。显然，当$\beta_1 = \beta_2$时，AMSIS问题就退化为标准的MSIS问题。特别地，我们还可以从理论上很容易证明如下困难关系在多项式时间的意义下成立：

$$\mathrm{MSIS}^\infty_{n,q,k,\ell_1+\ell_2,\max(\beta_1,\beta_2)} \leq \mathrm{AMSIS}^\infty_{n,q,k,\ell_1,\ell_2,\beta_1,\beta_2} \leq \mathrm{MSIS}^\infty_{n,q,k,\ell_1+\ell_2,\min(\beta_1,\beta_2)}.$$

换句话说，只要选择合适的参数，我们总能够保证AMSIS问题是难于求解的。在第6节中，我们将考虑MSIS问题的已有攻击算法及它们的变形来估计AMSIS问题的具体计算复杂度。

与文献[21]类似，由于压缩公钥的需要，我们还要用到AMSIS的一个变种的问题，即AMSIS-R问题。正式地，给定随机矩阵$\mathbf{A} \in R_q^{k\times(\ell_1+\ell_2-k)}$和随机向量$\mathbf{t} \in R_q^k$，$\mathrm{AMSIS\text{-}R}^\infty_{n,q,d,k,\ell_1,\ell_2,\beta_1,\beta_2}$问题的目标是寻找非零向量$\mathbf{x} \in R_q^{\ell_1+\ell_2+1}\backslash\{\mathbf{0}\}$使得$(\mathbf{I}_k\|\mathbf{A}\|\mathbf{t}_1\cdot 2^d)\mathbf{x} = \mathbf{0} \bmod q$，$\|\mathbf{x}_1\|_\infty \leq \beta_1, \|\mathbf{x}_2\|_\infty \leq \beta_2$且$\|x_3\|_\infty \leq 2$，其中$\mathbf{x} =$

$\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ x_3 \end{pmatrix} \in R_q^{\ell_1+\ell_2+1}, \mathbf{x}_1 \in R_q^{\ell_1}, \mathbf{x}_2 \in R_q^{\ell_2}, x_3 \in R_q$，$(\mathbf{t}_1, \mathbf{t}_0) := \mathsf{Power2Round}_q(\mathbf{t}, d)$。由于对$\mathbf{t}_1$和$x_3$的限制，直观上，求解AMSIS-R问题不比求解AMSIS问题容易。

为了便于系统的实现和提高安全性，我们将使用均匀分布来作为AMLWE的噪音分布。在没有特别说明的情况下，本文档将使用符号$\mathrm{AMLWE}_{n,q,k,\ell,\eta_1,\eta_2}$来表示使用$S_{\eta_1}$上的均匀分布来作为秘密向量$\mathbf{s}$的分布和$S_{\eta_2}$上的均匀分布来作为噪音向量$\mathbf{e}$的分布的AMLWE问题，其中$\eta_1, \eta_2 \in \mathbb{Z}$是正整数。

定义 1 (AMLWE困难假设) *对于适当选取的正整数$n, q, k, \ell, \eta_1, \eta_2 \in \mathbb{Z}$，不存在（量子）多项式时间的敌手能够解决*AMLWE问题$\mathrm{AMLWE}_{n,q,k,\ell,\eta_1,\eta_2}$。

定义 2 (AMSIS-R困难假设) *对于适当选取的正整数$n, q, k, d, \ell_1, \ell_2, \beta_1, \beta_2 \in \mathbb{Z}$，不存在（量子）多项式时间的敌手能够求解*AMSIS-R问题$\mathrm{AMSIS\text{-}R}^{\infty}_{n,q,d,k,\ell_1,\ell_2,\beta_1,\beta_2}$。

5.2 数字签名方案定义

数字签名方案的语法. 消息空间为$\mathcal{M}$的数字签名方案$\mathsf{SIG} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$由以下算法组成：

密钥生成算法: 给定安全参数κ，概率算法KeyGen输出一对公私钥(pk, sk)。密钥生成过程表示为$(pk, sk) \leftarrow \mathsf{KeyGen}(\kappa)$。

签名生成算法: 确定性(或概率)算法Sign输入私钥sk和消息$M \in \mathcal{M}$，输出签名σ。签名过程表示为$\sigma \leftarrow \mathsf{Sign}(sk, M)$。

签名验证算法: 确定性算法Verify输入私钥pk，消息$M \in \mathcal{M}$和签名σ，输出0代表拒绝或者1代表接受。验证过程表示为$0/1 \leftarrow \mathsf{Verify}(pk, M, \sigma)$。

正确性. 我们称数字签名方案SIG是*正确的*，如果对于所有消息$M \in \mathcal{M}$以下成立：

$$\Pr\left[\ (pk, sk) \leftarrow \mathsf{KeyGen}(\kappa) :\ \mathsf{Verify}(pk, M, \mathsf{Sign}(sk, M)) = 1\ \right] = 1,$$

其中概率来自于KeyGen(和Sign)使用的随机数。

定义 3 (SUF-CMA) *我们称数字签名方案SIG满足在自适应选择消息攻击下强不可伪造性(即$SUF\text{-}CMA$安全性)，如果以下定义的敌手优势$\mathsf{Adv}^{\mathsf{SUF-CMA}}_{\mathsf{SIG}}(\mathcal{A})$是可忽略的，*

$$\mathsf{Adv}^{\mathsf{SUF-CMA}}_{\mathsf{SIG}}(\mathcal{A}) := \Pr\left[b = 1 \;\middle|\; \begin{array}{l} (pk, sk) \leftarrow \mathsf{KeyGen}(\kappa); \\ (M^*, \sigma^*) \leftarrow \mathcal{A}^{\mathrm{SIGN}(\cdot)}(pk); \\ b := (\mathsf{Verify}(pk, M^*, \sigma^*) \wedge (M^*, \sigma^*) \notin Q) \end{array} \right],$$

其中签名预言机$\text{SIGN}(\cdot)$定义为：对于消息询问M，生成签名$\sigma \leftarrow \mathsf{Sign}(sk, M)$，添加$(M, \sigma)$到集合$Q$中(即$Q := Q \cup \{(M, \sigma)\}$)，然后返回$\sigma$作为回答。

如果Q仅是询问消息的集合以及仅仅要求$M^* \notin Q$，那么上述定义将退化为更弱的在自适应选择消息攻击下存在不可伪造性(即UF-CMA安全性)。当敌手$\mathcal{A}$不允许询问签名预言机的时候，上述安全定义将变为更弱的在无消息攻击下强不可伪造性(即SUF-NMA安全性)。

5.3 埃奎斯签名方案的选择消息强存在不可伪造安全性

在这一节中，我们将分别在随机预言机模型和量子随机预言机模型下证明埃奎斯签名方案的安全性。特别地，在随机预言机模型(ROM)[12]中，敌手$\mathcal{A}$能询问一个随机预言机多项式次。在量子随机预言机模型(QROM)[13]中，敌手$\mathcal{A}$能用任意输入字符串构成的量子叠加态(Superpositions)作为输入来询问量子随机预言机，且可以执行任意多项式次这样的询问。

随机预言机模型(ROM)下的选择消息强存在不可伪造安全性(SUF-CMA). 为了证明的更加直观和简洁，我们忽略相关用于实现的编码和解码，以及NTT等相关操作，而主要关注于下列形式化方案的设计。正式地，令$n, q, k, \ell, d, \omega, \eta_1, \eta_2, \beta_1, \beta_2, \gamma_1, \gamma_2$与第3节中的含义相同，令$\mathsf{H}_1 : \mathcal{B}^{n/8} \to R_q^{k \times \ell}$是用于生成公钥中矩阵$\mathbf{A}$的杂凑函数，那么我们可以将第3节中的签名方案形式化的描述为签名方案$\mathsf{Aigis\text{-}sig}' = (\mathsf{Aigis\text{-}sig}'.\mathsf{KeyGen}, \mathsf{Aigis\text{-}sig}'.\mathsf{Sign}, \mathsf{Aigis\text{-}sig}'.\mathsf{Verify})$。特别地，如果在实现中用于生成矩阵$\mathbf{A}$的$\mathsf{Parse}$和$\mathsf{XOF}_1$函数的组合满足随机预言机的性质，且$\mathsf{XOF}_2 : \mathcal{B}^{32} \times \mathcal{B}^{48} \times \mathbb{Z} \to \mathcal{B}^*$是伪随机函数，那么签名方案$\mathsf{Aigis\text{-}sig}'$与第3节中描述的签名方案的安全性相同。正式地，

- **密钥生成算法**$\mathsf{Aigis\text{-}sig}'.\mathsf{KeyGen}(1^\kappa)$：随机选择$\rho \xleftarrow{\$} \mathcal{B}^{32}$，$\mathbf{s}_1 \xleftarrow{\$} S_{\eta_1}^{\ell}, \mathbf{s}_2 \xleftarrow{\$} S_{\eta_2}^{k}$，计算$\mathbf{A} = \mathsf{H}_1(\rho) \in R_q^{k \times \ell}, \mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 \in R_q^k, (\mathbf{t}_1, \mathbf{t}_0) := \mathsf{Power2Round}_q(\mathbf{t}, d)$和$tr = \mathsf{CRH}(pk) \in \mathcal{B}^{48}$，最后返回公钥$pk = (\rho, \mathbf{t}_1)$和私钥$sk = (\rho, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0)$。[3]
- **签名生成算法**$\mathsf{Aigis\text{-}sig}'.\mathsf{Sign}(sk, M)$：给定私钥$sk = (\rho, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0)$和消息$M \in \mathcal{B}^*$，计算$\mu = \mathsf{CRH}(tr \| M)$，并执行以下步骤来产生签名：
  1. 随机选择$\mathbf{y} \xleftarrow{\$} \mathbf{S}_{\gamma_1 - 1}^{\ell}$；
  2. 计算$\mathbf{w} = \mathbf{A}\mathbf{y} \in R_q^k$和$\mathbf{w}_1 = \mathsf{HighBits}_q(\mathbf{w}, 2\gamma_2)$；
  3. 计算$c = \mathsf{H}(\mu, \mathbf{w}_1)$，$\mathbf{z} = \mathbf{y} + c\mathbf{s}_1$和$\mathbf{u} = \mathbf{w} - c\mathbf{s}_2$；
  4. 计算$(\mathbf{r}_1, \mathbf{r}_0) := \mathsf{Decompose}_q(\mathbf{u}, 2\gamma_2)$；

[3] 第3节方案的描述私钥中还选择了一个随机串$K \xleftarrow{\$} \mathcal{B}^{32}$，该随机串的作用在于产生签名算法所有的随机数（即实现确定性的签名过程）。我们在形式化方案的描述中暂时忽略随机串K。

5. 如果$\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta_1$或$\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta_2$或$\mathbf{r}_1 \neq \mathbf{w}_1$，返回第1步重新开始；
6. 计算$\mathbf{v} = c\mathbf{t}_0$, $\mathbf{h} := \mathsf{MakeHint}_q(-\mathbf{v}, \mathbf{u} + \mathbf{v}, 2\gamma_2)$；
7. 如果$\|\mathbf{v}\|_\infty \geq \gamma_2$或$\mathcal{HW}(\mathbf{h}) > \omega$，返回第1步重新开始；
8. 输出签名$\sigma = (\mathbf{z}, \mathbf{h}, c)$。

– **签名验证算法**$\mathsf{Aigis\text{-}sig}'.\mathsf{Verify}(pk, M, \sigma)$：给定公钥$pk = (\rho, \mathbf{t}_1)$，消息$M$和签名$\sigma = (\mathbf{z}, \mathbf{h}, c)$，首先计算$\mathbf{A} = \mathsf{H}_1(\rho)$，$\mu = \mathsf{CRH}(\mathsf{CRH}(pk)\|M)$，$\mathbf{u} = \mathbf{A}\mathbf{z} - c\mathbf{t}_1 \cdot 2^d$。然后，计算$\mathbf{w}_1' = \mathsf{UseHints}_q(\mathbf{h}, \mathbf{u}, 2\gamma_2)$和$c' = \mathsf{H}(\mu, \mathbf{w}_1')$。如果$\|\mathbf{z}\|_\infty < \gamma_1 - \beta_1$且$c = c'$且$\mathcal{HW}(\mathbf{h}) \leq \omega$，返回1，否则返回0。

对于安全性，我们有如下结论：

定理 1 *如果*$\mathsf{H}_1 : \mathcal{B}^{32} \to R_q^{k\times\ell}$*和*$\mathsf{H} : \mathcal{B}^{48} \times R_q^k \to B_{60}$*是随机预言机，且*$\mathsf{CRH} : \mathcal{B}^* \to \mathcal{B}^{48}$*是抗碰撞的杂凑函数，那么基于*$\mathrm{AMLWE}_{n,q,k,\ell,\eta_1,\eta_2}$*困难假设和*$\mathrm{AMSIS\text{-}R}^\infty_{n,q,d,k,k,\ell,4\gamma_2+2,2\gamma_1}$*困难假设，签名方案*$\mathsf{Aigis\text{-}sig}'$*满足*SUF-CMA*安全性。*

在给出定理1的证明之前，我们将先证明两个引理。特别地，在引理的证明中我们将假设敌手获得$\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$作为公钥。由于在真实的埃奎斯签名方案中敌手仅获得$\mathbf{t}$的高位比特，因此从攻击的角度该假设实际上更加有利于敌手，而从安全的角度该假设也意味着敌手要在实际中攻击埃奎斯签名方案的SUF-CMA安全性可能更加困难。

引理 5 *如果*H*是随机预言机,那么存在一个*PPT*的模拟器*$\mathcal{S}$*能够在仅知道公钥*pk*的情况下成功模拟任意消息的签名。*

证明. 在给出模拟器$\mathcal{S}$的构造之前，我先分析在签名算法中计算的$(\mathbf{z} = \mathbf{y} + c\mathbf{s}_1, c = \mathsf{H}(\mu, \mathbf{w}_1))$的分布，其中概率来自于随机选择$\mathbf{y}$和随机预言机$\mathsf{H}$。我们有以下概率等式成立：

$$\Pr[(\mathbf{z}, c)] = \Pr[c] \cdot \Pr[\mathbf{y} = \mathbf{z} - c\mathbf{s}_1 | c]$$

当向量$\mathbf{z}$满足$\|\mathbf{z}\|_\infty < \gamma_1 - \beta_1$且$\|c\mathbf{s}_1\|_\infty \leq \beta_1$时，我们有$\|\mathbf{y}\|_\infty = \|\mathbf{z} - c\mathbf{s}_1\|_\infty \leq \gamma_1 - 1$，即任意选取的$\mathbf{y}$都满足该不等式。因此，当不等式$\|\mathbf{z}\|_\infty < \gamma_1 - \beta_1$成立时，所有满足该关系的元组$(\mathbf{z}, c)$有都具有相同的概率分布。如果签名算法直接输出满足不等式$\|\mathbf{z}\|_\infty < \gamma_1 - \beta_1$的向量$\mathbf{z}$（即忽略其他所有检查），那么输出的签名$\sigma$对应的$(\mathbf{z}, c)$将是在$S^\ell_{\gamma_1-\beta_1-1} \times B_{60}$中均匀随机分布的。

基于以上的概率分析，再结合上文献[8,31]中的签名模拟方法，我们容易构造一个模拟器$\mathcal{S}$来模拟签名预言机询问的回答。模拟器$\mathcal{S}$仅知道公钥$pk = (\rho, \mathbf{t})$，对任意的消息询问M它需要回答一个合法的签名$\sigma = (\mathbf{z}, \mathbf{h}, c)$。模拟器$\mathcal{S}$能利用$\rho$计算矩阵$\mathbf{A}$，然后计算$\mu := \mathsf{CRH}(\mathsf{CRH}(pk)\|M)$。接下来，$\mathcal{S}$随机选取$(\mathbf{z}, c) \xleftarrow{\$} S^\ell_{\gamma_1-\beta_1-1} \times B_{60}$满足以下关系成立：

$$\|\mathsf{LowBits}_q(\mathbf{A}\mathbf{z} - c\mathbf{t}, 2\gamma_2)\|_\infty < \gamma_2 - \beta_2$$

根据等式(1)，我们有$\mathbf{w}-c\mathbf{s}_2=\mathbf{A}\mathbf{z}-c\mathbf{t}$。从而，我们有以下不等式成立：

$$\|\mathbf{r}_0\|_\infty=\|\mathsf{LowBits}_q(\mathbf{w}-c\mathbf{s}_2,2\gamma_2)\|_\infty<\gamma_2-\beta_2$$

换句话说，当$\|c\mathbf{s}_2\|_\infty\leq\beta_2$成立时，由引理2我们有以下等式成立：

$$\mathbf{r}_1=\mathsf{HighBits}_q(\mathbf{w}-c\mathbf{s}_2,2\gamma_2)=\mathsf{HighBits}_q(\mathbf{w},2\gamma_2)=\mathbf{w}_1$$

换句话说，$\mathcal{S}$并不用真的执行$\mathbf{r}_1=\mathbf{w}_1$的检查，而可以假设该等式总是成立。然后，$\mathcal{S}$能编程随机预言机$\mathsf{H}$使得$\mathsf{H}(\mu,\mathbf{w}_1)=c$成立。如果$\mathsf{H}(\mu,\mathbf{w}_1)$之前还没有被定义，那么$\mathcal{S}$模拟的元组$(\mathbf{z},c)$与真实签名对应的元组有相同的分布。进一步，由$\mathbf{y}$的随机性可知，$\mathsf{H}(\mu,\mathbf{w}_1)$已经被定义的概率是可忽略的。此外，根据等式(2)，模拟器$\mathcal{S}$能计算提示向量

$$\mathbf{h}:=\mathsf{MakeHint}_q(-c\mathbf{t}_0,\mathbf{w}-c\mathbf{s}_2+c\mathbf{t}_0)=\mathsf{MakeHint}_q(-c\mathbf{t}_0,\mathbf{A}\mathbf{z}-c\mathbf{t}_1\cdot 2^d),$$

其中$(\mathbf{t}_1,\mathbf{t}_0)=\mathsf{Power2Round}_q(\mathbf{t})$。

综上所述，如果我们选取参数使得$\|c\mathbf{s}_1\|_\infty\leq\beta_1$和$\|c\mathbf{s}_2\|_\infty\leq\beta_2$以$1-\mathsf{negl}(\kappa)$的概率成立时，那么，模拟器$\mathcal{S}$能以$1-\mathsf{negl}(\kappa)$的概率模拟一个与真实签名相同分布的$\sigma=(\mathbf{z},\mathbf{h},c)$。 □

引理 6 *伪造签名将解决*$\text{AMSIS-R}^\infty_{n,q,d,k,k,\ell,4\gamma_2+2,2\gamma_1}$*问题。特别地，对于随机选取的矩阵*$\mathbf{A}\in R_q^{k\times\ell}$*与向量*$\mathbf{t}\in R_q^k$*，该问题要求计算*$\mathbf{u}_1$*，*$\mathbf{u}_2$*和*$u_3$*满足：*

$$\begin{aligned}&\|\mathbf{u}_1\|_\infty\leq 2\gamma_1,\|\mathbf{u}_2\|_\infty\leq 4\gamma_2+2,\|u_3\|_\infty\leq 2,\\&\mathbf{A}\mathbf{u}_1+\mathbf{u}_2=u_3\mathbf{t}_1\cdot 2^d,\\&(\mathbf{u}_1,\mathbf{u}_2,u_3)\neq\mathbf{0},\\&\mathbf{u}_2\text{有至多有}2\omega\text{个系数的绝对值大于}2\gamma_2,\end{aligned}$$

其中$(\mathbf{t}_1,\mathbf{t}_0)=\mathsf{Power2Round}_q(\mathbf{t},d)$。

由于对于$\mathbf{u}_2$中系数的限制，上述问题实际上比普通$\text{AMSIS-R}^\infty_{n,q,d,k,k,\ell,4\gamma_2+2,2\gamma_1}$问题更加困难。这也意味着我们的签名方案在实际中将更加安全。

证明. 对于任意针对埃奎斯签名方案SUF-CMA安全性的PPT敌手$\mathcal{A}$，存在一个抽取器(Extractor)$\mathcal{E}$能解决以上引理中陈述的困难问题。抽取器$\mathcal{E}$被给定$(\mathbf{A},\mathbf{t})$。抽取器$\mathcal{E}$能随机选取$\rho\xleftarrow{\$}\mathcal{B}^{n/8}$，然后编程随机预言机使得$\mathsf{H}_1(\rho)=\mathbf{A}$（由随机预言机$\mathsf{H}_1$的性质和$\rho$的随机性可知，$\rho$之前被用于询问随机预言机$\mathsf{H}_1$的概率是可忽略的），并输出$pk=(\rho,\mathbf{t})$作为公钥。抽取器$\mathcal{E}$能利用引理5中的模拟器$\mathcal{S}$回答敌手$\mathcal{A}$的签名预言机询问。

当敌手$\mathcal{A}$伪造了一个关于消息M的合法签名$(\mathbf{z},\mathbf{h},c)$，那么我们有如下关系式成立：

$$\mathsf{H}\big(\mu,\mathbf{w}_1=\mathsf{UseHint}_q(\mathbf{h},\mathbf{Az}-c\mathbf{t}_1\cdot 2^d,2\gamma_2)\big)=c \tag{5}$$

其中$\mu=\mathsf{CRH}(\mathsf{CRH}(pk),M)$。

下面，我们分别考虑两种不同的情况。

情况1: 如果c是由签名预言机产生的，那么$\mathcal{S}$已经回答了关于某个消息M'的签名$(\mathbf{z}',\mathbf{h}',c)$。由强不可伪造的成功条件，我们有$(M',(\mathbf{z}',\mathbf{h}',c))\neq(M,(\mathbf{z},\mathbf{h},c))$。令$\mu'=\mathsf{CRH}(\mathsf{CRH}(pk),M')$。那么，存在$\mathbf{w}_1'$使得等式$\mathsf{H}(\mu,\mathbf{w}_1)=c=\mathsf{H}(\mu',\mathbf{w}_1')$成立。由于H是随机预言机的性质，我们有等式$\mu=\mu'$和$\mathbf{w}_1=\mathbf{w}_1'$成立的概率至少为$1-\mathsf{negl}(\kappa)$。由CRH是抗碰撞杂凑函数的性质，我们进一步有$M=M'$。在这种情况下，我们有以下等式成立：

$$\begin{aligned}\mathsf{UseHint}_q(\mathbf{h},\mathbf{Az}-c\mathbf{t}_1\cdot 2^d,2\gamma_2)&=\mathbf{w}_1\\ \mathsf{UseHint}_q(\mathbf{h}',\mathbf{Az}'-c\mathbf{t}_1\cdot 2^d,2\gamma_2)&=\mathbf{w}_1\end{aligned}$$

如果$\mathbf{z}=\mathbf{z}'$，那么根据引理1我们有$\mathbf{h}=\mathbf{h}'$。显然，这与$(M',(\mathbf{z}',\mathbf{h}',c))\neq(M,(\mathbf{z},\mathbf{h},c))$的假设矛盾。换句话说，我们一定有$\mathbf{z}\neq\mathbf{z}'$。根据引理1，我们有以下关系成立：

$$\begin{aligned}\|\mathbf{Az}-c\mathbf{t}_1\cdot 2^d-\mathbf{w}_1\cdot 2\gamma_2\|_\infty&\leq 2\gamma_2+1\\ \|\mathbf{Az}'-c\mathbf{t}_1\cdot 2^d-\mathbf{w}_1\cdot 2\gamma_2\|_\infty&\leq 2\gamma_2+1\end{aligned}$$

由三角不等式可知$\|\mathbf{A}(\mathbf{z}-\mathbf{z}')\|_\infty\leq 4\gamma_2+2$。换句话说，存在向量$\mathbf{u}_1\in R_q^\ell,\mathbf{u}_2\in R_q^k$满足$\|\mathbf{u}_1\|_\infty\leq 2\gamma_1,\|\mathbf{u}_2\|_\infty\leq 4\gamma_2+2$和$\mathbf{Au}_1+\mathbf{u}_2=\mathbf{0}$，其中$\mathbf{u}_1=(\mathbf{z}-\mathbf{z}')\neq\mathbf{0}$。因为向量$\mathbf{h}$与$\mathbf{h}'$至多有$\omega$个非零（且等于1）的元素，由引理1可知，向量$\mathbf{Az}-c\mathbf{t}_1\cdot 2^d-\mathbf{w}_1\cdot 2\gamma_2$和向量$\mathbf{Az}'-c\mathbf{t}_1\cdot 2^d-\mathbf{w}_1'\cdot 2\gamma_2$至多有$\omega$个系数的绝对值大于$\gamma_2$。因此，向量$\mathbf{u}_2$至多有$2\omega$个系数的绝对值大于$2\gamma_2$。综上所述，$\mathcal{E}$找到了引理中AMSIS-R问题的一个解$(\mathbf{u}_1,\mathbf{u}_2,0)\neq\mathbf{0}$。

情况2: 如果c是由敌手询问随机预言机产生的，即敌手使用了某个μ'和$\mathbf{w}_1'$询问随机预言机H，并得到了回答c。换句话说，我们有$\mathsf{H}(\mu'\|\mathbf{w}_1')=c$。由随机预言机的性质可知，以$1-\mathsf{negl}(\kappa)$的概率我们有$\mu'=\mu=\mathsf{CRH}(\mathsf{CRH}(pk),M)$，且$\mathbf{w}_1'=\mathbf{w}_1=\mathsf{UseHint}_q(\mathbf{h},\mathbf{Az}-c\mathbf{t}_1\cdot 2^d,2\gamma_2)$。

由标准的分支引理(Forking Lemma)[40,11]可知，抽取器$\mathcal{E}$能抽取两个伪造的签名$(\mathbf{z},\mathbf{h},c)$和$(\mathbf{z}',\mathbf{h}',c')$满足$c\neq c'$且

$$\begin{aligned}\mathsf{UseHint}_q(\mathbf{h},\mathbf{Az}-c\mathbf{t}_1\cdot 2^d,2\gamma_2)&=\mathbf{w}_1,\\ \mathsf{UseHint}_q(\mathbf{h}',\mathbf{Az}'-c'\mathbf{t}_1\cdot 2^d,2\gamma_2)&=\mathbf{w}_1.\end{aligned}$$

根据引理1，我们有以下关系成立：

$$\|\mathbf{Az} - c\mathbf{t}_1 \cdot 2^d - \mathbf{w}_1 \cdot 2\gamma_2\|_\infty \leq 2\gamma_2 + 1$$
$$\|\mathbf{Az}' - c'\mathbf{t}_1 \cdot 2^d - \mathbf{w}_1 \cdot 2\gamma_2\|_\infty \leq 2\gamma_2 + 1$$

由三角不等式可知，$\|\mathbf{A}(\mathbf{z} - \mathbf{z}') - (c - c')\mathbf{t}_1 \cdot 2^d\|_\infty \leq 4\gamma_2 + 2$。换句话说，存在向量$\mathbf{u}_1 \in R_q^\ell, \mathbf{u}_2 \in R_q^k, u_3 \in R_q$满足$\|\mathbf{u}_1\|_\infty \leq 2\gamma_1, \|\mathbf{u}_2\|_\infty \leq 4\gamma_2 + 2, \|u_3\|_\infty \leq 2$和$\mathbf{Au}_1 + \mathbf{u}_2 = u_3\mathbf{t}_1 \cdot 2^d$，其中$\mathbf{u}_1 = (\mathbf{z} - \mathbf{z}'), u_3 = c - c' \neq 0$。因为向量$\mathbf{h}$与$\mathbf{h}'$至多有$\omega$个非零（且等于1）的元素，由引理1可知，向量$\mathbf{Az} - c\mathbf{t}_1 \cdot 2^d - \mathbf{w}_1 \cdot 2\gamma_2$和向量$\mathbf{Az}' - c\mathbf{t}_1 \cdot 2^d - \mathbf{w}_1 \cdot 2\gamma_2$至多有$\omega$个系数的绝对值大于$\gamma_2$。因此，向量$\mathbf{u}$至多有$2\omega$个系数的绝对值大于$2\gamma_2$。综上所述，$\mathcal{E}$找到了引理中AMSIS-R问题的一个解$(\mathbf{u}_1, \mathbf{u}_2, u_3) \neq \mathbf{0}$。 □

接下来，我们将基于上述两个引理来证明定理1。

证明. 我们的证明通过一系列实验(即$\mathbf{G}_0, \mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3$)来执行。

实验$\mathbf{G}_0$： 实验$\mathbf{G}_0$是真实的SUF-CMA安全实验。正式地，该实验将按如下方式为敌手模拟攻击环境：

- **随机预言机H_1的模拟：** 该实验将为随机预言机H_1维护一个询问列表$\mathcal{L}_1 := \{(\rho_i, \mathbf{A}_i)\}$。当收到敌手$\mathcal{A}$的随机预言机询问$\rho$时，$\mathcal{C}_1$首先查询列表$\mathcal{L}$中是否存在元组$(\rho, \mathbf{A})$。如果不存在，$\mathcal{C}_1$将随机选择$\mathbf{A} \xleftarrow{\$} R_q^{k\times k}$并将$(\rho, \mathbf{A})$加入到列表$\mathcal{L}_1$。此后，$\mathcal{C}_1$将矩阵$\mathbf{A}$返回给敌手$\mathcal{A}$；
- **随机预言机H的模拟：** 该实验将为随机预言机H维护一个询问列表$\mathcal{L}_2 := \{((\mu_i, \mathbf{w}_{1,i}), c_i)\}$。当收到敌手$\mathcal{A}$的随机预言机询问$(\mu, \mathbf{w}_1)$时，$\mathcal{C}_1$首先查询列表$\mathcal{L}_2$中是否存在元组$((\mu, \mathbf{w}_1), c)$。如果不存在，$\mathcal{C}_1$将随机选择$c \xleftarrow{\$} B_{60}$并将$((\mu, \mathbf{w}_1), c)$加入到列表$\mathcal{L}_2$。此后，$\mathcal{C}_1$将$c$返回给敌手$\mathcal{A}$；
- **生成公钥：** 运行密钥生成算法$(pk, sk) \leftarrow \mathsf{Aigis}'.\mathsf{KeyGen}(\kappa)$，并将公钥$pk = (\rho, \mathbf{t}_1)$交给敌手$\mathcal{A}$；
- **回答签名询问：** 收到$\mathcal{A}$的签名询问M后，计算$\sigma \leftarrow \mathsf{Aigis}'.\mathsf{Sign}(sk, M)$，并将签名$\sigma$发送给敌手$\mathcal{A}$。

最后，敌手$\mathcal{A}$将伪造一个新的消息和签名对(M^*, σ^*)。

实验$\mathbf{G}_1$： 除了在实验过程中检查是否存在消息M'和M使得等式$\mathsf{CRH}(\mathsf{CRH}(pk), M) = \mathsf{CRH}(\mathsf{CRH}(pk), M')$成立，并且在等式成立后直接中止实验之外，实验$\mathbf{G}_1$与$\mathbf{G}_0$相同。

由CRH是抗碰撞的杂凑函数，实验$\mathbf{G}_1$中途中止的概率是可略的。换句话说，实验$\mathbf{G}_1$和$\mathbf{G}_0$是计算不可区分的。

实验$\mathbf{G}_2$： 除了用引理5中的模拟器$\mathcal{S}$回答签名询问之外，实验$\mathbf{G}_2$与$\mathbf{G}_1$相同。

由引理5可知，实验$\mathbf{G}_2$和$\mathbf{G}_1$是计算不可区分的。

实验$\mathbf{G}_3$ 除了在Aigis-sig′.KeyGen算法中直接随机选择$\mathbf{A} \xleftarrow{\$} R_q^{k\times\ell}$和$\mathbf{t} \in R_q^k$来生成公钥$pk$之外，实验$\mathbf{G}_3$与$\mathbf{G}_2$相同。

如果存在敌手$\mathcal{A}$能够区分实验$\mathbf{G}_3$与实验$\mathbf{G}_2$，那么我们可以构造一个攻击AMLWE困难假设的PPT敌手$\mathcal{C}_1$。具体地，给定$\mathrm{AMLWE}_{n,q,k,\ell,\eta_1,\eta_2}$问题的一个实例$(\mathbf{A},\mathbf{t}) \in R_q^{k\times\ell} \times R_q^k$作为输入，敌手$\mathcal{C}_1$要判断$(\mathbf{A},\mathbf{t})$是否选自于$R_q^{k\times\ell} \times R_q^k$的随机分布。正式地，$\mathcal{C}_1$将按下列方式修改公钥$pk$的生成方式，除此之外，$\mathcal{C}_1$与敌手$\mathcal{A}$的交互与实验$\mathbf{G}_2$中相同：

- **生成公钥：**随机选择$\rho \xleftarrow{\$} \mathcal{B}^{32}$，查询列表$\mathcal{L}_1$中是否存在元组$(\rho,*)$。如果存在，则中止实验。否则，将元组$(\rho,\mathbf{A})$加入列表$\mathcal{L}_1$（即定义$\mathsf{H}_1(\rho)=\mathbf{A}$）。然后，计算$(\mathbf{t}_1,\mathbf{t}_0) := \mathsf{Power2Round}_q(\mathbf{t},d)$和$tr = \mathsf{CRH}(\rho\|\mathbf{t}_1) \in \mathcal{B}^{48}$，最后，将公钥$pk=(\rho,\mathbf{t}_1)$返回给敌手$\mathcal{A}$。

最后，$\mathcal{C}_1$将$\mathcal{A}$的猜测$b' \in \{0,1\}$作为自己对于$\mathrm{AMLWE}_{n,q,k,k,\eta_1,\eta_2}$问题的输出。

首先，由H_1是随机预言机的假设和$\rho \in \mathcal{B}^{32}$的随机性，列表$\mathcal{L}_1$中存在元组$(\rho,*)$的概率是可忽略的。其次，如果$(\mathbf{A},\mathbf{t})$随机选自于$R_q^{k\times\ell} \times R_q^k$，那么$\mathcal{C}_1$与敌手$\mathcal{A}$的交互与实验$\mathbf{G}_3$中相同，否则$\mathcal{C}_1$与敌手$\mathcal{A}$的交互与实验$\mathbf{G}_2$中相同。换句话说，如果敌手$\mathcal{A}$能够区分实验$\mathbf{G}_3$和$\mathbf{G}_2$，那么$\mathcal{C}_1$能够解决$\mathrm{AMLWE}_{n,q,k,\ell,\eta_1,\eta_2}$问题。因此，在$\mathrm{AMLWE}_{n,q,k,\ell,\eta_1,\eta_2}$困难假设下，实验$\mathbf{G}_3$和$\mathbf{G}_2$是计算不可区分的。

显然，如果敌手能在实验$\mathbf{G}_3$中以不可忽略的概率伪造了一个新的消息和签名(M,σ)对，那么我们能用引理6中的抽取器$\mathcal{E}$以不可忽略的概率解决AMSIS-R问题。综上所述，定理1得证。 □

量子随机预言机模型(QROM)下的SUF-CMA安全性. 类似于文献[21]中一样，为了证明埃奎斯签名方案在量子随机预言机模型下的安全性，我们还需要引入如下AMSIS问题的变种问题SelfTargetAMSIS问题。正式地，令$\mathsf{H}:\{0,1\}^* \to B_{60}$是一个密码学杂凑函数，SelfTargetAMSIS问题$\mathrm{SelfTargetAMSIS}^{\infty}_{\mathsf{H},n,q,k,\ell_1,\ell_2,q,\beta_1,\beta_2}$的目标是给定随机矩阵$\mathbf{A} \in R_q^{k\times(\ell_1+\ell_2-k)}$和随机向量$\mathbf{t} \in R_q^k$，要求能够以量子态询问杂凑函数$\mathsf{H}$的算法输出$\mathbf{y} = \begin{pmatrix}\mathbf{y}_1\\ \mathbf{y}_2\\ c\end{pmatrix}$和$\mu \in \{0,1\}^*$，使得$\|\mathbf{y}_1\|_\infty \le \beta_1, \|\mathbf{y}_2\|_\infty \le \beta_2, \|c\|_\infty \le 1$且$\mathsf{H}(\mu,(\mathbf{I}_k\|\mathbf{A}\|\mathbf{t})\mathbf{y}) = c$成立。

定义 4 (SelfTargetAMSIS困难假设) *对于适当选取的正整数$n,q,k,\ell_1,\ell_2,\beta_1,\beta_2 \in \mathbb{Z}$和杂凑函数$\mathsf{H}$，不存在（量子）多项式时间的敌手能够求解*SelfTargetAMSIS问题$\mathrm{SelfTargetAMSIS}^{\infty}_{\mathsf{H},n,q,k,\ell_1,\ell_2,\beta_1,\beta_2}$。

根据文献[28]的结论，我们有如下定理。

定理 2 基于$\mathrm{AMLWE}_{n,q,k,\ell,\eta_1,\eta_2}$困难假设、$\text{AMSIS-R}^{\infty}_{n,q,d,k,k,\ell,4\gamma_2+2,2\gamma_1}$困难假设和$\mathrm{SelfTargetAMSIS}^{\infty}_{\mathsf{H},n,q,k,\ell_1,\ell_2,4\gamma_2,\gamma_1}$困难假设，埃奎斯签名方案在量子随机预言机模型下是可证明*SUF-CMA*安全的。

直观上，AMLWE困难假设用于阻止密钥恢复攻击，AMSIS困难假设用于获得强不可伪造，而SelfTargetAMSIS困难假设则用于阻止对新消息的签名伪造（在经典随机预言机模型下，借助于分支引理，AMSIS困难假设就足够用于阻止对新消息的签名伪造。引入SelfTargetAMSIS困难假设主要是解决量子随机预言机模型下不能使用分支引理的原因）。

定理2的证明可以通过修改文献[28]中的证明而得到。接下来，我们简单说明与SelfTargetAMSIS困难假设相关的部分（与AMSIS问题相关的部分与引理6中关于情况1部分的证明类似）。首先，文献[28]证明了在量子随机预言机模型下对于具有零知识性的数字签名方案，UF-CMA安全性可归约到UF-NMA安全性，其中UF-NMA安全性没有允许敌手做签名询问（即仅获得公钥）。在AMLWE困难假设下，我们能用随机的向量$\mathbf{t} \xleftarrow{\$} R_q^k$替换公钥$pk$中的向量$\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$。根据引理5和文献[28]在量子随机预言机模型下的模拟技术，我们能证明埃奎斯签名方案具有零知识性，即只知道公钥的情况下模拟任意消息的签名。

从而，在$\mathrm{AMLWE}_{n,q,k,\ell,\eta_1,\eta_2}$困难假设下，我们仅需要分析敌手在仅获得均匀随机公钥$(\mathbf{A},\mathbf{t})$的情况下，伪造一个合法消息/签名$(M,\sigma=(\mathbf{z},\mathbf{h},c))$的困难性。注意在量子随机预言机模型下，直接设置$\mathbf{A}$作为公钥与设置随机种子$\rho$作为公钥从安全性角度来说是一样的。令$\mu := \mathsf{CRH}(\mathsf{CRH}(pk), M)$，那么签名是合法的意味着以下关系成立：

$$\begin{aligned}&\|\mathbf{z}\|_\infty < \gamma_1 - \beta_1\\ &\mathsf{H}(\mu, \mathsf{UseHint}_q(\mathbf{h}, \mathbf{A}\mathbf{z} - c\mathbf{t}_1 \cdot 2^d, 2\gamma_2)) = c\\ &\mathcal{HW}(\mathbf{h}) \le \omega\end{aligned}$$

从引理1，我们可知以下等式成立：

$$\mathsf{UseHint}_q(\mathbf{h}, \mathbf{A}\mathbf{z} - c\mathbf{t}_1 \cdot 2^d, 2\gamma_2) = \mathbf{A}\mathbf{z} - c\mathbf{t}_1 \cdot 2^d + \mathbf{y}, \tag{6}$$

其中$\|\mathbf{y}\|_\infty \le 2\gamma_2 + 1$。而且，$\mathbf{y}$中至多有$\omega$个系数$\ell_\infty$范数大于$\gamma_2$。如果我们写$\mathbf{t} = \mathbf{t}_1 \cdot 2^d + \mathbf{t}_0$，其中$\|\mathbf{t}_0\|_\infty \le 2^{d-1}$，那么我们重写等式6如下：

$$\mathbf{A}\mathbf{z} - c\mathbf{t}_1 \cdot 2^d + \mathbf{y} = \mathbf{A}\mathbf{z} - c(\mathbf{t} - \mathbf{t}_0) + \mathbf{y} = \mathbf{A}\mathbf{z} - c\mathbf{t} + (c\mathbf{t}_0 + \mathbf{y}) = \mathbf{A}\mathbf{z} - c\mathbf{t} + \mathbf{y}' \tag{7}$$

$\mathbf{y}'$的ℓ_∞范数的上界为：

$$\|\mathbf{y}'\|_\infty \le \|c\mathbf{t}_0\|_\infty + \|\mathbf{y}\|_\infty \le \|c\|_1 \cdot \|\mathbf{t}_0\|_\infty + \|\mathbf{y}\|_\infty \le 60 \cdot 2^{d-1} + 2\gamma_2 + 1 < 4\gamma_2$$

因此，敌手$\mathcal{A}$需要找$(M,(\mathbf{z},c,\mathbf{y}'))$满足$\|\mathbf{z}\|_\infty < \gamma_1-\beta_1, \|c\|_\infty = 1, \|\mathbf{y}'\|_\infty < 4\gamma_2$和$M \in \{0,1\}^*$使得以下成立：

$$\mathsf{H}\left(\mu,\ (\mathbf{I}_k\|\mathbf{A}\|\mathbf{t})\cdot\begin{bmatrix}\mathbf{y}'\\ \mathbf{z}\\ -c\end{bmatrix}\right)=c, \tag{8}$$

其中$\mu=\mathsf{CRH}(\mathsf{CRH}(pk),M)$。因为$(\mathbf{A},\mathbf{t})$是均匀随机的，所以等式(8)定义的问题就是SelfTargetAMSIS困难问题。换句话说，在量子随机预言机模型下，如果存在一个量子多项式时间的敌手$\mathcal{A}$能够伪造一个关于新消息的合法签名，那么我们能构造一个敌手$\mathcal{C}$解决SelfTargetAMSIS困难问题。

6 抵抗已知攻击的能力

求解LWE的算法主要包括原始攻击(primal attack)，对偶攻击(dual Attack)以及利用BKW、Arora-Ge方法直接求解[5]等。由于BKW、Arora-Ge攻击方法需要的样本数据量多为指数或次指数量级，这两类方法并不适用于分析只有比较少样本的具体密码方案。因此，对实际格上密码系统的分析往往只考虑原始攻击和对偶攻击这两种目前最有效的攻击方法。此外，由于这两类方法在求解RLWE和MLWE问题时并没有比求解标准LWE问题更有优势，因此文献中在分析基于MLWE和RLWE问题密码方案的具体安全强度时，往往只是将相应的RLWE或MLWE问题转换成标准LWE问题来进行分析[21,14]。特别地，我们首先会将$\mathrm{AMLWE}_{n,q,k,\ell,\alpha_1,\alpha_2}$问题转换成 $\mathrm{ALWE}_{nk,q,k\ell,\alpha_1,\alpha_2}$问题，然后再将针对LWE问题的求解方法推广到求解ALWE问题。由于任意其他有界中心对称分布都可以看成服从一定参数的亚高斯分布，为了便于分析且不失一般性，我们将只考虑秘密向量$\mathbf{s}\xleftarrow{\$}\chi_{\alpha_1}^n$和噪音向量$\mathbf{e}\xleftarrow{\$}\chi_{\alpha_2}^m$的各分量分别选自于以$\alpha_1$和$\alpha_2$为标准差的亚高斯分布的$\mathrm{ALWE}_{n,q,m,\alpha_1,\alpha_2}$问题。具体地，我们的目标是在给定问题样本

$$(\mathbf{A},\mathbf{b}=\mathbf{A}\mathbf{s}+\mathbf{e})\in\mathbb{Z}_q^{m\times n}\times\mathbb{Z}_q^m$$

的情况下，计算并输出$\mathbf{s}\in\mathbb{Z}_q^n$，其中$\mathbf{s}\xleftarrow{\$}\chi_{\alpha_1}^n, \mathbf{e}\xleftarrow{\$}\chi_{\alpha_2}^m$。

目前，求解SIS主要方法是BKZ及其变形的格基约化算法。由于这类算法在求解RSIS和MSIS问题时并没有比求解标准SIS问题更有优势，因此文献中分析基于MSIS和RSIS问题密码方案的具体安全强度时，往往只是将对应的RSIS或MSIS问题转换成标准SIS问题来进行分析[21]。特别地，我们首先将$\mathrm{AMSIS}^\infty_{n,q,k,\ell_1,\ell_2,\beta_1,\beta_2}$问题转换成$\mathrm{ASIS}^\infty_{nk,q,n\ell_1,n\ell_2,\beta_1,\beta_2}$问题，然后将针对SIS问题的求解方法推广到求解ASIS问题。正式地，本节考虑的ASIS问题的目标是在给定$\mathrm{ASIS}^\infty_{n,q,m_1,m_2,\beta_1,\beta_2}$问题实例$\mathbf{A}=(\mathbf{A}_1\|\mathbf{A}_2)\in\mathbb{Z}_q^{n\times(m_1+m_2-n)}$的情况下，计算并输出$\mathbf{x}^T=(\mathbf{x}_1^T,\mathbf{x}_2^T)$使其满足$\|\mathbf{x}_1\|\le\beta_1, \|\mathbf{x}_2\|\le\beta_2$且$\mathbf{A}_1\mathbf{x}_1+\mathbf{A}_2\mathbf{x}_2=\mathbf{0} \bmod q$，其中$\mathbf{x}_1\in\mathbb{Z}^{m_1}, \mathbf{x}_2\in\mathbb{Z}^{m_2}$。

6.1 针对ALWE问题的原始攻击及其变形

原始攻击的基本思路是通过嵌入的方式将ALWE问题转化为求解适当的格上有界译码问题(BDD)或短向量问题，不同原始攻击的主要区别在于嵌入的方式不同。我们将考虑以下针对ALWE问题$\mathrm{ALWE}_{n,q,m,\alpha_1,\alpha_2}$的原始攻击及其变形。

传统原始攻击. 传统原始攻击利用了Kannan嵌入[27,4]将LWE的求解问题转换成求解格中的唯一最短向量问题(uSVP)。首先，定义格

$$\Lambda = \{\mathbf{y} \in \mathbb{Z}^m | \mathbf{y} = \mathbf{A}\mathbf{x} \bmod q, \mathbf{x} \in \mathbb{Z}_q^n\},$$

则格Λ的维数$d = m$。当m足够大于n时，矩阵$\mathbf{A}$以很大概率存在n个线性无关的行。不妨设$\mathbf{A}$的前n行线性无关（否则我们可以通过行变换将线性无关的行置换到前n行），且设前n行对应的子矩阵为$\mathbf{A}_1 \in \mathbb{Z}_q^{n\times n}$，即 $\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{pmatrix}$。记矩阵$\mathbf{A}_1^{-1} \in \mathbb{Z}_q^{n\times n}$为$\mathbf{A}_1$的逆，令 $\mathbf{A}' = \begin{pmatrix} \mathbf{I}_n \\ \mathbf{A}_2\mathbf{A}_1^{-1} \end{pmatrix}$，则我们有

$$\Lambda = \{\mathbf{y}|\mathbf{y} = \mathbf{A}\mathbf{x} \bmod q, \mathbf{x} \in \mathbb{Z}_q^n\} = \{\mathbf{y}|\mathbf{y} = \mathbf{A}'\mathbf{x} \bmod q, \mathbf{x} \in \mathbb{Z}_q^n\} \subset \mathbb{Z}^m,$$

且以下矩阵

$$\mathbf{B} = \begin{pmatrix} \mathbf{I}_n & 0 \\ \mathbf{A}_2\mathbf{A}_1^{-1} & q\mathbf{I}_{m-n} \end{pmatrix} \in \mathbb{Z}^{m\times m}$$

的列向量构成格Λ的一组基，容易有$\det(\Lambda) = q^{m-n}$。由$\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$可知向量$\mathbf{b}$距离格$\Lambda$的距离为$\|\mathbf{e}\|$。如果能找到一个离$\mathbf{b}$最近的格点$\mathbf{u} \in \Lambda$，则我们将以很大概率有$\mathbf{e} = \mathbf{b} - \mathbf{u}$. 换句话说，求解向量$\mathbf{e}$的问题可转化为格$\Lambda$上的有界译码问题(BDD)，从而可利用文献[29]中的多最近平面算法进行求解。进一步，利用Kannan嵌入[27,4]可以将BDD问题转化为唯一最短向量问题(uSVP)。具体地，考虑由以下矩阵$\mathbf{B}'$的列向量生成的格Λ',

$$\mathbf{B}' = \begin{pmatrix} \mathbf{I}_n & 0 & \\ \mathbf{A}_2\mathbf{A}_1^{-1} & q\mathbf{I}_{m-n} & \mathbf{b} \\ 0 & 0 & t \end{pmatrix} \in \mathbb{Z}^{(m+1)\times(m+1)},$$

其中$t \in \mathbb{Z}$是一个可调参数。理论上，取$t = \|\mathbf{e}\|$时能够取得较好的效果，但实际实验则显示$t = 1$时效果较好，所以我们通常选取$t = 1$。在这种情况下，我们以极大的概率有$\mathbf{v} = (\mathbf{e}^T, 1)^T \in \mathbb{Z}^{m+1}$为格$\Lambda'$中唯一最短向量。因此，我们可以通过求解格$\Lambda'$中uSVP问题来求解噪音向量$\mathbf{e} \in \mathbb{Z}^m$，进而通过求解线性方程组来恢复私钥$\mathbf{s} \in \mathbb{Z}_q^n$，即计算$\mathbf{A}\mathbf{s} = \mathbf{b} - \mathbf{e}$。此外，由于向量$\mathbf{e}$的每个分量都选自于亚高斯分布$\chi_{\alpha_2}$，我们有$\|\mathbf{v}\| \approx \|\mathbf{e}\| \approx \alpha_2\sqrt{m}$，即$\mathbf{v}$的长度较短。

原始攻击：变形1. 当$\alpha_1 = \alpha_2$时，这种攻击算法即为目前最有效的原始攻击算法[9,6]。定义格

$$\Lambda = \{\mathbf{v} = (\mathbf{x}^T, \mathbf{y}^T, z)^T \in \mathbb{Z}^{n+m+1} | (\mathbf{A}\|\mathbf{I}_m\| - \mathbf{b})\mathbf{v} = 0 \bmod q\},$$

容易验证矩阵

$$\mathbf{B} = \begin{pmatrix} \mathbf{I}_n & 0 & 0 \\ -\mathbf{A} & q\mathbf{I}_m & \mathbf{b} \\ 0 & 0 & 1 \end{pmatrix} \in \mathbb{Z}^{(m+n+1)\times(m+n+1)}$$

的列向量构成格Λ的一组基。显然，格Λ维数为$d = m + n + 1$。进一步，我们有$\det(\Lambda) = q^m$，且$\mathbf{v} = (\mathbf{s}^T, \mathbf{e}^T, 1)^T \in \mathbb{Z}^{n+m+1}$是格$\Lambda$的一个短向量。因此，我们可以通过求解格$\Lambda$中的(u)SVP问题来恢复向量$\mathbf{s} \in \mathbb{Z}_q^m$。此外，由于向量$\mathbf{s}$和$\mathbf{e}$的每个分量都分别选自于亚高斯分布$\chi_{\alpha_1}$和$\chi_{\alpha_2}$，我们有$\|\mathbf{v}\| \approx \sqrt{\alpha_1^2 n + \alpha_2^2 m}$。

原始攻击：变形2. 这种攻击算法由文献[9,6]中原始攻击算法进一步变形而来。当$\alpha_1 \neq \alpha_2$且α_1与α_2的值相差不大时，该变形的原始攻击算法将更加有效。正式地，令$c = \alpha_2/\alpha_1$。定义格

$$\Lambda = \left\{ \mathbf{v} = \begin{pmatrix} c\mathbf{x} \\ \mathbf{y} \\ \alpha_2 z \end{pmatrix} \in \mathbb{R}^{n+m+1} \middle| (\mathbf{A}\|\mathbf{I}_m\| - \mathbf{b})\mathbf{u} = 0 \bmod q, \mathbf{u} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \\ z \end{pmatrix} \in \mathbb{Z}^{n+m+1} \right\},$$

则以下矩阵

$$\mathbf{B} = \begin{pmatrix} c\mathbf{I}_n & 0 & 0 \\ -\mathbf{A} & q\mathbf{I}_m & \mathbf{b} \\ 0 & 0 & \alpha_2 \end{pmatrix} \in \mathbb{R}^{(n+m+1)\times(n+m+1)},$$

的列向量构成格Λ的一组基。显然，格Λ维数为$d = m + n + 1$。进一步，我们有$\det(\Lambda) = \alpha_2 c^n q^m$，且$\mathbf{v} = (c\mathbf{s}^T, \mathbf{e}^T, \alpha_2)^T \in \mathbb{R}^{m+n+1}$为格$\Lambda$中的一个短向量。因此，我们可以通过求解格$\Lambda$的(u)SVP问题来恢复私钥$\mathbf{s}$。此外，由于向量$\mathbf{s}$和$\mathbf{e}$的每个分量都分别选自于亚高斯分布$\chi_{\alpha_1}$和$\chi_{\alpha_2}$，我们有$\|\mathbf{v}\| \approx \alpha_2\sqrt{n + m + 1}$。

直观上，该变形攻击算法的思想是将目标短向量的每个分量都放缩成同等的规模，从而达到更好的攻击效果。事实上，实际实验表明格上(u)SVP问题求解算法的确更加倾向于输出每个分量值都比较平衡的短向量。

原始攻击的计算代价评估模型. 目前求解最短向量最有效的办法为BKZ-b格基约化算法及其变形算法。给定d维格的基作为输入，这类算法会将格基约化问题转变成$b < d$维子格中的最短向量问题，并不断通过多次在不同b维子格中求解最短

向量问题来达到提高输入d维格基质量的目的（即降低格基中向量的范数）。根据算法的不同，实际的BKZ-b算法通常会调用$O(d)$次b维格的最短向量求解算法。因此，BKZ-b格基约化算法的计算代价主要由b维格的最短向量求解算法的计算代价来决定。

通常，我们假设运行BKZ-b算法对d维格的基$\mathbf{B} \in \mathbb{Z}^{d\times d}$进行约化得到的约化基满足几何级数假设，即GSA假设（该情况从攻击者角度为最优）。设得到的一组约化基$\hat{\mathbf{B}} = (\hat{\mathbf{b}}_1, \ldots, \hat{\mathbf{b}}_d)$，则有

$$\|\hat{\mathbf{b}}_1\| = \delta^d \det(\Lambda)^{1/d}, \qquad \|\hat{\mathbf{b}}_i^*\| = \delta^{-2d(i-1)/(d-1)}\|\hat{\mathbf{b}}_1\|,$$

其中 $\delta = ((\pi b)^{1/b} \cdot b/2\pi e)^{\frac{1}{2(b-1)}}$[18], $\hat{\mathbf{B}}^* = (\hat{\mathbf{b}}_1^*, \ldots, \hat{\mathbf{b}}_d^*)$是矩阵$\hat{\mathbf{B}}$的正交化矩阵（即$\hat{\mathbf{b}}_1 = \hat{\mathbf{b}}_1^*$）。特别地，研究[6,4]表明当目标唯一最短向量$\mathbf{v}$在最后$b$个正交向量$(\hat{\mathbf{b}}_{d-b+1}^*, \ldots, \hat{\mathbf{b}}_d^*)$构成空间中的投影向量的范数小于$\|\hat{\mathbf{b}}_{d-b+1}^*\|$时，那么运行BKZ-$b$格基约化算法将能够恢复向量$\mathbf{v}$。

对于范数为$\ell = \|\mathbf{v}\|$的唯一最短向量$\mathbf{v}$，其在最后b个正交向量$(\hat{\mathbf{b}}_{d-b+1}^*, \ldots, \hat{\mathbf{b}}_d^*)$构成空间中的投影向量的范数约为$\ell\sqrt{b/d}$，即假设向量$\mathbf{v}$在每个投影分量上的值近似相等。在这种情况下，原始攻击算法及其变形的计算代价主要由满足不等式

$$\ell\sqrt{b/d} \leq \delta^{(-d^2+2db-d)/(d-1)} \det(\Lambda)^{1/d} \tag{9}$$

的最小b值来决定。与文献[6]中一样，我们将直接考虑用满足不等式(9)的b维子格SVP求解算法的复杂度来非常保守地估计求解d维格中唯一最短向量的复杂度（因为后者需要将b维子格SVP求解算法作为子算法运行非常多次）。进一步，与许多文献(例如[6,21,14])中一样，我们将分别使用$\text{cost}_b = 2^{0.292b}$和$\text{cost}_b = 2^{0.265b}$来分别刻画求解$b$维格最短向量的经典算法和量子算法的复杂度。在这种保守模型下，原始攻击及其变形算法的复杂度主要由满足不等式(9)的b值来确定。

6.2 针对ALWE问题的对偶攻击及其变形

对偶攻击首先将LWE问题转化为SIS问题，然后利用SIS问题的解来将求解LWE问题转化成区分一定参数下的亚高斯分布和$\mathbb{Z}_q^n$上的均匀分布的问题。我们将考虑以下针对ALWE问题$\text{ALWE}_{n,q,m,\alpha_1,\alpha_2}$的对偶攻击及其变形算法。

传统对偶攻击. 文献[36]首先给出了对偶攻击的方法。由于传统对偶攻击只考虑LWE问题的噪音分布，该类攻击方法实际上并不区分ALWE问题和LWE问题，即直接将ALWE问题看成LWE问题。特别地，当$\alpha_1 \gg \alpha_2$时，传统对偶攻击比较有效。正式地，令

$$\Lambda = \{\mathbf{x} \in \mathbb{Z}^m | \mathbf{A}^T\mathbf{x} = \mathbf{0} \bmod q\},$$

则Λ的维数$d = m$。当m足够大于n时, 矩阵$\mathbf{A}$以很大概率存在n个线性无关的列。不妨设$\mathbf{A}^T$的前n列线性无关（否则我们可以通过列变换将线性无关的列置换到前n列），且设前n列对应的子矩阵为$\mathbf{A}_1 \in \mathbb{Z}_q^{n\times n}$，即$\mathbf{A}^T = (\mathbf{A}_1 \| \mathbf{A}_2) \in \mathbb{Z}_q^{n\times m}$。记矩阵$\mathbf{A}_1^{-1} \in \mathbb{Z}_q^{n\times n}$为$\mathbf{A}_1$的逆，令$\mathbf{A}' = (\mathbf{I}_n \| \mathbf{A}_1^{-1}\mathbf{A}_2)$，那么我们有

$$\Lambda = \{\mathbf{x} \in \mathbb{Z}^m | \mathbf{A}^T\mathbf{x} = \mathbf{0} \bmod q\} = \{\mathbf{x} \in \mathbb{Z}^m | \mathbf{A}'\mathbf{x} = \mathbf{0} \bmod q\},$$

且矩阵

$$\mathbf{B} = \begin{pmatrix} q\mathbf{I}_n & -\mathbf{A}_1^{-1}\mathbf{A}_2 \\ 0 & \mathbf{I}_{m-n} \end{pmatrix} \in \mathbb{Z}^{m\times m}$$

的列向量构成格Λ的一组基，显然$\det(\Lambda) = q^n$。首先，传统对偶攻击将从以$\mathbf{B}$为基的m维格Λ中寻找一个短向量$\mathbf{v} \in \mathbb{Z}^m$满足$\mathbf{A}^T\mathbf{v} = \mathbf{0} \bmod q$（即求解SIS问题）。此后，攻击算法将计算$u = \langle \mathbf{v}, \mathbf{b}\rangle = \langle \mathbf{v}, \mathbf{e}\rangle \bmod q$，并使用已知的算法来区分$u$和选自于$\mathbb{Z}_q$上均匀分布的元素。特别地，如果$\ell = \|\mathbf{v}\|$比较小，那么$\langle \mathbf{v}, \mathbf{e}\rangle$的值比较小，且元素$u$可大致看作服从于以$\ell\alpha_2$为标准差的亚高斯分布。在这种情况下，存在有效算法能够以$4\exp(-2\pi^2\tau^2)$的概率区分u和$\mathbb{Z}_q$中均匀分布的元素，其中$\tau = \ell\alpha_2/q$。

显然，对于ALWE问题而言，α_2取值越大，$\ell\alpha_2$的值也就越大，从而将u和选自于$\mathbb{Z}_q$中均匀分布的元素区分开来的概率就越小。因此，直观上，α_2的值越大，传统对偶攻击算法将更难对ALWE进行攻击。

对偶攻击：变形1. 该变形的对偶攻击由[6]中的对偶攻击算法扩展而来。事实上，当$\alpha_1 = \alpha_2$，该变形的算法就是[6]中的对偶攻击算法。正式地，定义格

$$\Lambda = \{(\mathbf{x}^T, \mathbf{y}^T)^T \in \mathbb{Z}^{m+n} | \mathbf{A}^T\mathbf{x} = \mathbf{y} \bmod q\},$$

则格Λ的维数$d = m+n$。当m足够大于n时, 矩阵$\mathbf{A}^T$以很大概率存在n个线性无关的列。不妨设$\mathbf{A}^T$的前n列线性无关（否则我们可以通过列变换将线性无关的列置换到前n列），且设前n列对应的子矩阵为$\mathbf{A}_1 \in \mathbb{Z}_q^{n\times n}$，即 $\mathbf{A}^T = (\mathbf{A}_1 \| \mathbf{A}_2) \in \mathbb{Z}_q^{n\times m}$。记矩阵$\mathbf{A}_1^{-1} \in \mathbb{Z}_q^{n\times n}$为$\mathbf{A}_1$的逆，令$\mathbf{A}' = (\mathbf{I}_n \| \mathbf{A}_1^{-1}\mathbf{A}_2)$，那么我们有

$$\Lambda = \{(\mathbf{x}^T, \mathbf{y}^T)^T | \mathbf{A}^T\mathbf{x} = \mathbf{y} \bmod q\} = \{(\mathbf{x}^T, \mathbf{y}^T)^T | \mathbf{A}'\mathbf{x} = \mathbf{A}_1^{-1}\mathbf{y} \bmod q\},$$

且矩阵

$$\mathbf{B} = \begin{pmatrix} q\mathbf{I}_n & -\mathbf{A}_1^{-1}\mathbf{A}_2 & \mathbf{A}_1^{-1} \\ 0 & \mathbf{I}_{m-n} & 0 \\ 0 & 0 & \mathbf{I}_n \end{pmatrix} \in \mathbb{Z}^{(m+n)\times(m+n)}$$

的列向量构成格Λ的一组基，显然$\det(\Lambda) = q^n$。首先，该变形的对偶攻击将从以$\mathbf{B}$为基的$m+n$维格Λ中寻找一个短向量$\mathbf{v} = (\mathbf{x}^T, \mathbf{y}^T)^T \in \mathbb{Z}^{m+n}$满足$\mathbf{A}^T\mathbf{x} =$

$\mathbf{y} \bmod q$。此后，攻击算法将计算

$$u = \langle \mathbf{x}, \mathbf{b} \rangle = \langle \mathbf{y}, \mathbf{s} \rangle + \langle \mathbf{x}, \mathbf{e} \rangle \bmod q,$$

并使用已知的算法来区分u和选自于$\mathbb{Z}_q$上均匀分布的元素。特别地，如果$\ell = \|\mathbf{v}\|$比较小，那么$\langle \mathbf{y}, \mathbf{s} \rangle$与$\langle \mathbf{x}, \mathbf{e} \rangle$的值也都比较小。进一步，若假设$\mathbf{v}$中每个分量的值大致具有相同规模，那么元素$u = \langle \mathbf{y}, \mathbf{s} \rangle + \langle \mathbf{x}, \mathbf{e} \rangle$可看作服从于以$\ell\sqrt{\frac{\alpha_1^2 m + \alpha_2^2 n}{m+n}}$为标准差的亚高斯分布。在这种情况下，存在有效算法能够以$4\exp(-2\pi^2\tau^2)$的概率区分u 和$\mathbb{Z}_q$中均匀分布的元素，其中$\tau = \ell\sqrt{\frac{\alpha_1^2 m + \alpha_2^2 n}{m+n}}/q$。

对偶攻击：变形2. 当$\alpha_1 = \alpha_2$，对应ALWE问题退化为正规形LWE问题，且该变形的对偶攻击代表了目前针对正规形LWE问题的最有效的对偶攻击算法[6]。而$\alpha_1 \neq \alpha_2$但α_1与α_2相差不多时，该变形的对偶攻击在求解ALWE问题时则更加有效。正式地，令$c = \frac{\alpha_2}{\alpha_1}$，定义格

$$\Lambda = \{(\mathbf{x}^T, \mathbf{y}^T/c)^T \in \mathbb{R}^{m+n} | \mathbf{A}^T\mathbf{x} = \mathbf{y} \bmod q, \mathbf{x} \in \mathbb{Z}^m, \mathbf{y} \in \mathbb{Z}^n\},$$

则格Λ的维数$d = m+n$。当m足够大于n时, 矩阵$\mathbf{A}^T$以很大概率存在n个线性无关的列。不妨设$\mathbf{A}^T$的前n列线性无关（否则我们可以通过列变换将线性无关的列置换到前n列），且设前n列对应的子矩阵为$\mathbf{A}_1 \in \mathbb{Z}_q^{n\times n}$，即 $\mathbf{A}^T = (\mathbf{A}_1 \| \mathbf{A}_2) \in \mathbb{Z}_q^{n\times m}$。记矩阵$\mathbf{A}_1^{-1} \in \mathbb{Z}_q^{n\times n}$为$\mathbf{A}_1$的逆，令$\mathbf{A}' = (\mathbf{I}_n \| \mathbf{A}_1^{-1}\mathbf{A}_2)$，那么我们有

$$\Lambda = \{(\mathbf{x}^T, \mathbf{y}^T/c)^T | \mathbf{A}^T\mathbf{x} = \mathbf{y} \bmod q\} = \{(\mathbf{x}^T, \mathbf{y}^T/c)^T | \mathbf{A}'\mathbf{x} = \mathbf{A}_1^{-1}\mathbf{y} \bmod q\},$$

且矩阵

$$\mathbf{B} = \begin{pmatrix} q\mathbf{I}_n & -\mathbf{A}_1^{-1}\mathbf{A}_2 & \mathbf{A}_1^{-1} \\ 0 & \mathbf{I}_{m-n} & 0 \\ 0 & 0 & \frac{1}{c}\mathbf{I}_n \end{pmatrix} \in \mathbb{R}^{(m+n)\times(m+n)}$$

的列向量构成格Λ的一组基，显然有$\det(\Lambda) = (q/c)^n$。首先，该变形的对偶攻击将从以$\mathbf{B}$为基的$m+n$维格Λ中寻找一个短向量$\mathbf{v} = (\mathbf{x}^T, \hat{\mathbf{y}}^T)^T \in \mathbb{R}^{m+n}$满足$\mathbf{A}^T\mathbf{x} = c\hat{\mathbf{y}} \bmod q$。此后，攻击算法将计算

$$u = \langle \mathbf{x}, \mathbf{b} \rangle = c \cdot \langle \hat{\mathbf{y}}, \mathbf{s} \rangle + \langle \mathbf{x}, \mathbf{e} \rangle \bmod q,$$

并使用已知的算法来区分u和选自于$\mathbb{Z}_q$上均匀分布的元素。特别地，如果$\ell = \|\mathbf{v}\|$比较小，那么$c \cdot \langle \hat{\mathbf{y}}, \mathbf{s} \rangle$与$\langle \mathbf{x}, \mathbf{e} \rangle$的值也都比较小，且元素$u = c \cdot \langle \hat{\mathbf{y}}, \mathbf{s} \rangle + \langle \mathbf{x}, \mathbf{e} \rangle$可大致看作服从于以$\ell\alpha_2$为标准差的亚高斯分布。在这种情况下，存在有效算法能够以$4\exp(-2\pi^2\tau^2)$的概率区分u 和$\mathbb{Z}_q$中均匀分布的元素，其中$\tau = \ell\alpha_2/q$。

最后，需要指出的是该变形的对偶攻击并不关心α_1和α_2的大小关系。事实上，我们也可以用$c' = \alpha_1/\alpha_2$来进行相应的对偶攻击，但这种对偶攻击在我们的计算代价评估模型下与用系数$c = \alpha_2/\alpha_1$是等价的。

对偶攻击：变形3. 该变形的对偶攻击算法由文献[3]中的对偶攻击算法推广而来。该算法与变形2中的对偶攻击算法类似，其主要区别在于c的取值不同。正式地，令$c = \frac{\alpha_2\sqrt{m}}{\alpha_1\sqrt{n}}$。该变形的对偶攻击将首先计算一个短向量$\mathbf{v} = (\mathbf{x}^T, \hat{\mathbf{y}}^T)^T \in \mathbb{R}^{m+n}$满足$\mathbf{A}^T\mathbf{x} = c\hat{\mathbf{y}} \bmod q$。此后，攻击算法将计算

$$u = \langle \mathbf{x}, \mathbf{b} \rangle = c \cdot \langle \hat{\mathbf{y}}, \mathbf{s} \rangle + \langle \mathbf{x}, \mathbf{e} \rangle \bmod q,$$

并使用已知的算法来区分u和选自于$\mathbb{Z}_q$上均匀分布的元素。特别地，如果$\ell = \|\mathbf{v}\|$比较小，那么$c \cdot \langle \hat{\mathbf{y}}, \mathbf{s} \rangle$与$\langle \mathbf{x}, \mathbf{e} \rangle$的值也都比较小，且元素$u = c \cdot \langle \hat{\mathbf{y}}, \mathbf{s} \rangle + \langle \mathbf{x}, \mathbf{e} \rangle$可大致看作服从于以$\ell\alpha_2\sqrt{\frac{2m}{m+n}}$为标准差的亚高斯分布。此时，存在有效算法能够以$4\exp(-2\pi^2\tau^2)$的概率区分$u$和$\mathbb{Z}_q$中均匀分布的元素，其中$\tau = \ell\alpha_2\sqrt{\frac{2m}{m+n}}/q$。

类似地，我们也可以用系数$c' = \frac{\alpha_1\sqrt{n}}{\alpha_2\sqrt{m}}$来进行相应的对偶攻击，但这种情况在我们的计算代价评估模型下与用系数$c = \frac{\alpha_2\sqrt{m}}{\alpha_1\sqrt{n}}$是等价的。

对偶攻击的计算代价评估模型. 如上所述，对偶攻击及其变形算法的计算代价主要由求解短向量问题的计算代价和将求ALWE问题的区分优势转变为计算优势的代价组成（其中后者要求区分优势至少应该大于1/2）。对于求解短向量问题的代价，我们将使用原始攻击中的模型，即运行BKZ-b约化d维格的基，我们将得到范数为$\ell = \|\mathbf{v}\| = \delta^d \det(\Lambda)^{1/d}$的短向量$\mathbf{v}$。将此代入此前的分析，我们将得到区分ALWE问题的优势为$\epsilon = 4\exp(-2\pi^2\tau^2)$，其中$\tau$由$\ell$和具体的对偶攻击算法来唯一确定。为了使得最终的区分优势大于1/2，我们将需要$1/\epsilon^2$个短向量。考虑到每次运行筛法大约能够产生$2^{0.2075b}$个短向量，那么整个攻击需要重复大约$R = 1/\max(1, 1/(2^{0.2075b}\epsilon^2))$次。

如在原始攻击的计算代价模型一样，我们将使用b维子格SVP求解算法的复杂度来非常保守地估计求解d维格中最短向量的复杂度（因为后者需要将 b维子格SVP求解算法作为子算法运行非常多次）。进一步，我们将使用$\mathrm{cost}_b = 2^{0.292b}$和$\mathrm{cost}_b = 2^{0.265b}$来分别刻画求解$b$维格最短向量的经典算法和量子算法的复杂度。在这种保守模型下，对偶攻击最终复杂度为

$$1/\max(1, 1/(2^{0.2075b} \cdot 16\exp(-4\pi^2\tau^2))) \cdot \mathrm{cost}_b, \tag{10}$$

其中τ由b和具体的对偶攻击算法来唯一确定。

6.3 针对ASIS问题的攻击及其变形

相对求解无穷范数(A)SIS问题，BKZ及其变形格的基约化算法通常更适合用来求解二范数(A)SIS问题。特别地，对于Aigis签名方案选择的四组参数所对应的ASIS问题$\mathrm{ASIS}^{\infty}_{n,q,m_1,m_2,\beta_1,\beta_2}$问题实例$\mathbf{A} \in \mathbb{Z}_q^{n\times(m_1+m_2-n)}$，虽然目标向量的$\mathbf{x} =$

$\begin{pmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{pmatrix} \in \mathbb{Z}^{m_1+m_2}$满足$\|\mathbf{x}\|_\infty \leq \max(\beta_1,\beta_2) < q$（且$(\mathbf{I}_n\|\mathbf{A})\mathbf{x} = \mathbf{0} \bmod q$），但其二范数却可能为$\|x\| > q$。换句话说，虽然平凡解$\mathbf{u} = (q,0,\ldots,0)^T$的二范数可能小于目标向量的范数，即$\|\mathbf{u}\| < \|\mathbf{x}\|$，但其无穷范数却大于我们的要求，即$\|\mathbf{u}\|_\infty > \max(\beta_1,\beta_2)$。这就意味着我们直接用BKZ格基约化算法来求解无穷范数(A)SIS问题时将存在一定的困难。为此，我们将采用文献[21]的无穷范数SIS问题求解模型。此外，根据不等式$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|$，我们还可以将无穷范数SIS问题转变为二范数SIS问题，然后进而通过求解二范数SVP问题来求解无穷范数SIS问题。接下来，我们将上述两种攻击算法分别简称为无穷范数攻击算法和二范数攻击算法，并通过考虑它们以及它们的变形来评估求解$\mathrm{ASIS}^{\infty}_{n,q,m_1,m_2,\beta_1,\beta_2}$的复杂度。

传统二范数攻击. 该类攻击算法即为普通的SVP求解方法。正式地，定义

$$\Lambda = \{\mathbf{x} \in \mathbb{Z}^{m_1+m_2} | (\mathbf{I}_n\|\mathbf{A})\mathbf{x} = 0 \bmod q\}.$$

则矩阵

$$\mathbf{B} = \begin{pmatrix} q\mathbf{I}_n & -\mathbf{A} \\ 0 & \mathbf{I}_{m_1+m_2-n}\end{pmatrix} \in \mathbb{Z}^{(m_1+m_2)\times(m_1+m_2)}$$

的列向量构成格Λ的一组基。格Λ维数为$d = m_1+m_2$且$\det(\Lambda) = q^n$。为了使得二范数ASIS问题的解一定是无穷范数ASIS问题的解，目标向量$\mathbf{x} \in \mathbb{Z}^d$的范数必须满足$\|\mathbf{x}\| \leq \beta = \min(\beta_1,\beta_2)$。因此，该攻击算法的复杂度即为运行BKZ-$b$格基约化算法来寻找满足$\|\mathbf{x}\| \leq \beta = \min(\beta_1,\beta_2)$的向量$\mathbf{x}$的复杂度。

二范数攻击：变形1. 该类攻击算法希望通过平衡目标向量中各分量的值来达到较好的攻击效果。正式地，令$c = \beta_2/\beta_1$，定义格

$$\Lambda = \left\{ \mathbf{x} = \begin{pmatrix} c\mathbf{x}_1 \\ \mathbf{x}_2\end{pmatrix} \in \mathbb{R}^{m_1+m_2} \;\middle|\; (\mathbf{I}_n\|\mathbf{A})\mathbf{v} = 0 \bmod q, \mathbf{v} = \begin{pmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{pmatrix} \in \mathbb{Z}^{m_1+m_2} \right\},$$

则矩阵

$$\mathbf{B} = \begin{pmatrix} c\mathbf{I}_{m_1} & 0 \\ 0 & \mathbf{I}_{m_2}\end{pmatrix} \begin{pmatrix} q\mathbf{I}_n & -\mathbf{A} \\ 0 & \mathbf{I}_{m_1+m_2-n}\end{pmatrix} \in \mathbb{R}^{(m_1+m_2)\times(m_1+m_2)}$$

的列向量构成格Λ的一组基。格Λ维数为$d = m_1+m_2$且$\det(\Lambda) = c^{m_1}q^n$。显然，通过求解格$\Lambda$中满足二范数小于$\beta_2$的向量$\mathbf{x} = (c\mathbf{x}_1^T, \mathbf{x}_2^T)^T \in \mathbb{R}^{m_1+m_2}$，我们就能够得到ASIS问题的解$(\mathbf{x}_1^T, \mathbf{x}_2^T)^T \in \mathbb{Z}^{m_1+m_2}$。因此，该方法的复杂度即为运行BKZ-$b$格基约化算法来寻找满足$\|\mathbf{x}\| \leq \beta = \beta_2$的向量$\mathbf{x}$的复杂度。

最后，需要指出的是该变形二范数攻击并不关心β_1和β_2的大小关系。事实上，我们也可以用$c' = \beta_1/\beta_2$来进行相应的二范数攻击，但这种变形的攻击在我们的计算代价评估模型下与用系数$c = \beta_2/\beta_1$是等价的。

二范数攻击的计算代价评估模型. 令$\delta = ((\pi b)^{1/b} \cdot b/2\pi e)^{\frac{1}{2(b-1)}}$。运行BKZ-$b$算法来约化$d$维格的基$\mathbf{B} \in \mathbb{Z}^{d\times d}$将会得到一个满足$\|\mathbf{x}\| \approx \delta^d \det(\Lambda)^{1/d}$的格向量。显然，当不等式

$$\delta^d \det(\Lambda)^{1/d} \leq \beta \tag{11}$$

成立时，我们能成功用二范数攻击算法得到ASIS问题的解。与针对ALWE问题的原始攻击计算代价模型一样，我们将使用b维子格SVP求解算法的复杂度来非常保守地估计求解d维格中最短向量的复杂度（因为后者需要将 b维子格SVP求解算法作为子算法运行非常多次）。进一步，我们将使用$\mathrm{cost}_b = 2^{0.292b}$和$\mathrm{cost}_b = 2^{0.265b}$来分别刻画求解$b$维格最短向量的经典算法和量子算法的复杂度。在这种保守模型下，二范数攻击算法的最终复杂度主要由满足不等式(11)的b值来确定。

传统无穷范数攻击. 该类攻击算法由文献[21]中的确定性方法扩展而来。正式地，定义

$$\Lambda = \{\mathbf{x} \in \mathbb{Z}^{m_1+m_2} | (\mathbf{I}_n \| \mathbf{A})\mathbf{x} = 0 \bmod q\}.$$

则矩阵

$$\mathbf{B} = \begin{pmatrix} q\mathbf{I}_n & -\mathbf{A} \\ 0 & \mathbf{I}_{m_1+m_2} \end{pmatrix} \in \mathbb{Z}^{(m_1+m_2)\times(m_1+m_2)}$$

的列向量构成格Λ的一组基。显然，格Λ维数为$d = m_1 + m_2$且$\det(\Lambda) = q^n$。以矩阵$\mathbf{B}$为输入运行BKZ-b基约化算法，我们会得到一组约化基$\hat{\mathbf{B}} \in \mathbb{Z}^{d\times d}$。令矩阵$\hat{\mathbf{B}}^* = (\hat{\mathbf{b}}_1^*, \ldots, \mathbf{b}_d^*)$是矩阵$\hat{\mathbf{B}}$的格拉姆-施密特（Gram-Schmidt）正交化矩阵，且$\ell_i = \log_2(\|\hat{\mathbf{b}}_i^*\|)$，那么存在整数$1 \leq i \leq j \leq d$使得：

- 对于前i个向量，我们有$\ell_1 = \ell_2 = \cdots = \ell_i = \log_2 q$；
- 对于中间$j-i$个向量，我们有$\ell'_j = \log_2 q + s \cdot (j' - i)$，其中$j' \in \{i+1, \ldots, j\}$，$s = -\frac{1}{b-1}\log_2(\frac{b}{2\pi e}(\pi b)^{1/b}) < 0$，且$(j-i)(i+j+1) = -\frac{2(n-j)\log_2 q}{s}$
- 对于后$d-j$个向量，我们有$\ell_{j+1} = \ell_{j+2} = \cdots = \ell_d = 0$。

虽然BKZ-b格基约化算法不能够直接保证其寻找到的短向量满足ASIS问题的解。但以约化基$\hat{\mathbf{B}} \in \mathbb{Z}^{d\times d}$为输入，我们可以用相当于求解一次$b$维格SVP问题的复杂度得到$(\sqrt{4/3})^b$个向量[21]，使得这些向量在前$i$个基向量张成的垂直空间中投影向量的二范数约为$2^{\ell_{i+1}}$。通过对这$(\sqrt{4/3})^b$个向量的分布进行建模（见计算代价评估模型部分），我们就能够估计出该算法能这在$(\sqrt{4/3})^b$个向量中找到一个ASIS解的概率。用BKZ-b格基约化算法计算代价除以该概率即可得到该攻击算法的总体计算复杂度。

无穷范数攻击：变形1. 该类攻击由文献[21]中的随机化基的方法扩展而来。正式地，定义

$$\Lambda = \{\mathbf{x} \in \mathbb{Z}^{m_1+m_2} | (\mathbf{I}_n \| \mathbf{A})\mathbf{x} = 0 \bmod q\}.$$

则矩阵

$$\mathbf{B} = \begin{pmatrix} q\mathbf{I}_n & -\mathbf{A} \\ 0 & \mathbf{I}_{m_1+m_2} \end{pmatrix} \in \mathbb{Z}^{(m_1+m_2)\times(m_1+m_2)}$$

的列向量构成格Λ的一组基。显然，格Λ维数为$d = m_1 + m_2$且$\det(\Lambda) = q^n$。随机化矩阵$\mathbf{B}$得到$\mathbf{B}'$，使得$\mathbf{B}'$中不存在平凡向量$q\mathbf{e}_i$，其中$\mathbf{e}_i$为第i个单位向量。然后以矩阵$\mathbf{B}'$为输入运行BKZ-b格基约化算法，我们会得到一组约化基$\hat{\mathbf{B}} \in \mathbb{Z}^{d\times d}$。令矩阵$\hat{\mathbf{B}}^* = (\hat{\mathbf{b}}_1^*, \ldots, \mathbf{b}_d^*)$是矩阵$\hat{\mathbf{B}}$的格拉姆-施密特（Gram-Schmidt）正交化矩阵，且$\ell_i = \log_2(\|\hat{\mathbf{b}}_i^*\|)$，那么存在整数$1 \le j \le d$使得：

- 对于前j个向量，我们有$\ell_{j'} = -s \cdot (j - j' + 1)$，其中$j' \in \{1, \ldots, j\}$，$s = -\frac{1}{b-1}\log_2(\frac{b}{2\pi e}(\pi b)^{1/b}) < 0$，且$j(j+1) = -\frac{2n\log_2 q}{s}$；
- 对于后$d-j$个向量，我们有$\ell_{j+1} = \ell_{j+2} = \cdots = \ell_d = 0$。

进一步，以约化基$\hat{\mathbf{B}} \in \mathbb{Z}^{d\times d}$为输入，我们可以用相当于求解一次$b$维格SVP问题的复杂度得到$(\sqrt{4/3})^b$个向量[21]，使得这些向量的二范数约为$2^{\ell_1}$。通过对这$(\sqrt{4/3})^b$个向量的分布进行建模（见计算代价评估模型部分），我们就能够估计出该算法能这在$(\sqrt{4/3})^b$个向量中找到一个ASIS解的概率。用BKZ-b格基约化算法计算代价除以该概率即可得到该攻击算法的总体计算复杂度。

无穷范数攻击：变形2. 该类攻击算法希望通过平衡目标向量中各分量的值来达到较好的攻击效果。正式地，令$c = \beta_2/\beta_1$，定义格

$$\Lambda = \left\{ \mathbf{x} = \begin{pmatrix} c\mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \in \mathbb{R}^{m_1+m_2} \;\middle|\; (\mathbf{I}_n \| \mathbf{A})\mathbf{v} = 0 \bmod q, \mathbf{v} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \in \mathbb{Z}^{m_1+m_2} \right\}.$$

则矩阵

$$\mathbf{B} = \begin{pmatrix} c\mathbf{I}_{m_1} & 0 \\ 0 & \mathbf{I}_{m_2} \end{pmatrix} \begin{pmatrix} q\mathbf{I}_n & -\mathbf{A} \\ 0 & \mathbf{I}_{m_1+m_2-n} \end{pmatrix} \in \mathbb{R}^{(m_1+m_2)\times(m_1+m_2)}$$

的列向量构成格Λ的一组基。格Λ维数为$d = m_1 + m_2$且$\det(\Lambda) = c^{m_1}q^n$。显然，通过求解格$\Lambda$中满足无穷范数小于$\beta_2$的向量$\mathbf{x} = (c\mathbf{x}_1^T, \mathbf{x}_2^T)^T \in \mathbb{R}^{m_1+m_2}$，我们就能够得到ASIS问题的解$(\mathbf{x}_1^T, \mathbf{x}_2^T)^T \in \mathbb{Z}^{m_1+m_2}$。

进一步，随机化矩阵$\mathbf{B}$得到$\mathbf{B}'$，使得$\mathbf{B}'$中不存在平凡向量$q\mathbf{e}_i$，其中$\mathbf{e}_i$为第i个单位向量。然后以矩阵$\mathbf{B}'$为输入运行BKZ-b基约化算法，我们会得到一组约化基$\hat{\mathbf{B}} \in \mathbb{Z}^{d\times d}$。令矩阵$\hat{\mathbf{B}}^* = (\hat{\mathbf{b}}_1^*, \ldots, \mathbf{b}_d^*)$是矩阵$\hat{\mathbf{B}}$的格拉姆-施密特（Gram-Schmidt）正交化矩阵，令$\ell_i = \log_2(\|\hat{\mathbf{b}}_i^*\|)$，那么存在整数$1 \le j \le d$使得：

– 对于前j个向量，我们有$\ell_{j'} = -s \cdot (j - j' + 1)$，其中$j' \in \{1, \ldots, j\}$，$s = -\frac{1}{b-1}\log_2(\frac{b}{2\pi e}(\pi b)^{1/b}) < 0$，且$j(j+1) = -\frac{2n\log_2 q}{s}$；
– 对于后$d-j$个向量，我们有$\ell_{j+1} = \ell_{j+2} = \cdots = \ell_d = 0$。

与此前一样，以约化基$\hat{\mathbf{B}} \in \mathbb{Z}^{d\times d}$为输入，我们可以用相当于求解一次$b$维格SVP问题的复杂度得到$(\sqrt{4/3})^b$个向量[21]，使得这些向量二范数约为$2^{\ell_1}$。通过对这$(\sqrt{4/3})^b$个向量的分布进行建模（见计算代价评估模型部分），我们就能够估计出该算法能这在$(\sqrt{4/3})^b$个向量中找到一个ASIS解的概率。用BKZ-b格基约化算法计算代价除以该概率即可得到该攻击算法的总体计算复杂度。

最后，需要指出的是该变形二范数攻击并不关心β_1和β_2的大小关系。事实上，我们也可以用$c' = \beta_1/\beta_2$来进行相应的二范数攻击，但这种变形的攻击在我们的计算代价评估模型下与用系数$c = \beta_2/\beta_1$是等价的。

无穷范数攻击的计算代价评估模型. 令整数$i < j \leq m_1 + m_2$是由前述无穷范数方法及其变形确定的参数（对于变形1和变形2，令$i = 0$）。如文献[21]中一样，我们假设以约化基$\hat{\mathbf{B}} \in \mathbb{Z}^{d\times d}$为输入而得到$(\sqrt{4/3})^b$个向量满足如下分布：其前$i$维系数服从$\mathbb{Z}_q$上的均匀分布，其$i+1$到$j$维系数服从以$2^{\ell_{i+1}}/\sqrt{j-i}$为标准差的高斯分布，其最后$d-j$为系数为0。显然，服从此分布的向量的前$i$个系数小于$\beta_1$的概率（约为$\beta_1/q$）相对较小。为了增大无穷范数的传统方法及其变形1的成功率，我们进一步假设$\beta_1 > \beta_2$（即假设总是存在高效的算法能把无穷范数限制为$\max(\beta_1, \beta_2)$的向量系数调整到最前面，虽然我们并不知道是否真的有可以完成这样功能的算法，但从安全的角度这种假设只能使我们的评估更加保守）。在这种模型下，我们能够计算出上述三类算法能够在$(\sqrt{4/3})^b$个向量中找到一个向量$\mathbf{x} = (\mathbf{x}_1^T, \mathbf{x}_2^T)^T$满足$\|\mathbf{x}_1\|_\infty \leq \beta_1$且$\|\mathbf{x}_2\|_\infty \leq \beta_2$的概率$p$。

进一步，我们使用b维子格SVP求解算法的复杂度来非常保守的估计求解d维格中最短向量的复杂度（因为后者需要将 b维子格SVP求解算法作为子算法运行非常多次）。在这种保守模型下，无穷范数方法最终复杂度为cost_b/p。

6.4 埃奎斯签名方案的安全强度

由于攻击算法可能选择不同的样本数量和不同参数$b \in \mathbb{Z}$来运行BKZ-b算法，为了全面评估埃奎斯签名方案在四组参数下抵抗密钥恢复攻击的安全强度，我们选择穷举所有可能ALWE样本数量m和格基约化算法BKZ-b的b值来估计在不同LWE原始攻击和对偶攻击下的最优计算复杂度，设其为2^{sec}。特别地，表3给出了Aigis-sig 签名方案四组参数在不同ALWE问题原始攻击下达到最小攻击复杂度的(m, b, sec)值。表4给出了埃奎斯签名方案四组参数在不同ALWE问题对偶攻击下达到最小攻击复杂度的(m, b, sec)值。

表 3. 埃奎斯签名方案抵抗ALWE问题原始攻击的安全强度

参数集名称	攻击模型	传统原始攻击 (m, b, sec)	原始攻击变形1 (m, b, sec)	原始攻击变形2 (m, b, sec)
PARAMS I	经典	$(1021, 555, 162)$	$(671, 345, 100)$	$(\mathbf{741}, \mathbf{340}, \mathbf{99})$
	量子	$(1021, 555, 147)$	$(671, 345, 91)$	$(\mathbf{741}, \mathbf{340}, \mathbf{90})$
PARAMS II	经典	$(1276, 1060, 310)$	$(996, 500, 146)$	$(\mathbf{896}, \mathbf{490}, \mathbf{143})$
	量子	$(1276, 1060, 281)$	$(996, 500, 132)$	$(\mathbf{896}, \mathbf{490}, \mathbf{129})$
PARAMS II-b	经典	$(1276, 1060, 310)$	$(926, 510, 149)$	$(\mathbf{926}, \mathbf{505}, \mathbf{147})$
	量子	$(1276, 1060, 281)$	$(926, 510, 135)$	$(\mathbf{926}, \mathbf{505}, \mathbf{133})$
PARAMS III	经典	-	$(1101, 660, 193)$	$(\mathbf{1106}, \mathbf{615}, \mathbf{179})$
	量子	-	$(1101, 660, 175)$	$(\mathbf{1106}, \mathbf{615}, \mathbf{163})$

表 4. 埃奎斯签名方案抵抗ALWE问题对偶攻击的安全性强度

参数集名称	攻击模型	传统对偶攻击 (m, b, sec)	对偶攻击变形1 (m, b, sec)	对偶攻击变形2 (m, b, sec)	对偶攻击变形 3 (m, b, sec)
PARAMS I	经典	$(1021, 550, 160)$	$(\mathbf{786}, \mathbf{340}, \mathbf{99})$	$(\mathbf{706}, \mathbf{340}, \mathbf{99})$	$(\mathbf{706}, \mathbf{340}, \mathbf{99})$
	量子	$(1021, 550, 145)$	$(\mathbf{786}, \mathbf{340}, \mathbf{90})$	$(\mathbf{706}, \mathbf{340}, \mathbf{90})$	$(\mathbf{706}, \mathbf{340}, \mathbf{90})$
PARAMS II	经典	$(1276, 1050, 307)$	$(1121, 495, 144)$	$(\mathbf{966}, \mathbf{485}, \mathbf{141})$	$(\mathbf{966}, \mathbf{485}, \mathbf{141})$
	量子	$(1276, 1050, 278)$	$(1121, 495, 131)$	$(\mathbf{966}, \mathbf{485}, \mathbf{128})$	$(\mathbf{966}, \mathbf{485}, \mathbf{128})$
PARAMS II-b	经典	$(1276, 1050, 307)$	$(1006, 505, 147)$	$(\mathbf{1001}, \mathbf{500}, \mathbf{146})$	$(\mathbf{1011}, \mathbf{500}, \mathbf{146})$
	量子	$(1276, 1050, 278)$	$(1006, 505, 133)$	$(\mathbf{1001}, \mathbf{500}, \mathbf{132})$	$(\mathbf{1011}, \mathbf{500}, \mathbf{132})$
PARAMS III	经典	$(1535, 1535, 464)$	$(1381, 650, 190)$	$(\mathbf{1031}, \mathbf{615}, \mathbf{179})$	$(\mathbf{1036}, \mathbf{615}, \mathbf{179})$
	量子	$(1235, 1535, 422)$	$(1381, 650, 172)$	$(\mathbf{1031}, \mathbf{615}, \mathbf{163})$	$(\mathbf{1036}, \mathbf{615}, \mathbf{163})$

表 5. 埃奎斯签名方案抵抗ASIS问题二范数攻击的安全性强度

参数集名称	攻击模型	传统二范数攻击 (m, b, sec)	二范数攻击变形1 (m, b, sec)
PARAMS I	经典	$(2031, 750, 219)$	$(\mathbf{2031}, \mathbf{665}, \mathbf{194})$
	量子	$(2031, 750, 198)$	$(\mathbf{2031}, \mathbf{665}, \mathbf{176})$
PARAMS II	经典	$(2537, 1100, 321)$	$(\mathbf{2537}, \mathbf{900}, \mathbf{263})$
	量子	$(2537, 1100, 291)$	$(\mathbf{2537}, \mathbf{900}, \mathbf{238})$
PARAMS II-b	经典	$(2537, 1100, 321)$	$(\mathbf{2537}, \mathbf{900}, \mathbf{263})$
	量子	$(2537, 1100, 291)$	$(\mathbf{2537}, \mathbf{900}, \mathbf{238})$
PARAMS III	经典	$(3043, 1395, 408)$	$(\mathbf{3043}, \mathbf{1140}, \mathbf{333})$
	量子	$(3043, 1395, 370)$	$(\mathbf{3043}, \mathbf{1140}, \mathbf{302})$

类似地，由于攻击算法可能选择不同参数$b \in \mathbb{Z}$来运行BKZ-b算法并选择问题实例的部分列，为了全面评估埃奎斯签名方案在四组参数下抵抗签名伪造攻击的安全强度，我们选择穷举所有可能ASIS问题实例的列数m和格基约化算法BKZ-b的b值来估计在不同ASIS问题攻击算法下的最优计算复杂度，设其为2^{sec}。特别地，表5给出了埃奎斯签名方案四组参数在ASIS问题二范数攻击下的达到最小攻击复杂度的(m, b, sec)值。表6给出了埃奎斯签名方案四组参数在ASIS问题无穷范数攻击下的达到最小攻击复杂度的(m, b, sec)值。表7给出了埃奎斯签名方案四组参数达到的整体安全强度。注意到PARAMS III旨在提高PARAMS II参数集抵抗密钥恢复攻击的冗余，并不影响ASIS攻击的安全性。

表 6. 埃奎斯签名方案抵抗ASIS问题无穷范数攻击的安全性强度

参数集名称	攻击模型	传统无穷范数攻击 (m, b, sec)	无穷范数攻击变形1 (m, b, sec)	无穷范数攻击变形2 (m, b, sec)
PARAMS I	经典	$(1831, 385, 112)$	$(1781, 385, 112)$	$(\mathbf{1731}, \mathbf{360}, \mathbf{105})$
	量子	$(1831, 385, 102)$	$(1781, 385, 102)$	$(\mathbf{1731}, \mathbf{360}, \mathbf{95})$
PARAMS II	经典	$(2387, 495, 144)$	$(2387, 545, 159)$	$(\mathbf{2187}, \mathbf{485}, \mathbf{141})$
	量子	$(2387, 495, 131)$	$(2387, 545, 144)$	$(\mathbf{2187}, \mathbf{485}, \mathbf{128})$
PARAMS II-b	经典	$(2387, 495, 144)$	$(2387, 545, 159)$	$(\mathbf{2187}, \mathbf{485}, \mathbf{141})$
	量子	$(2387, 495, 131)$	$(2387, 545, 144)$	$(\mathbf{2187}, \mathbf{485}, \mathbf{128})$
PARAMS III	经典	$(2743, 630, 184)$	$(2793, 690, 201)$	$(\mathbf{2543}, \mathbf{615}, \mathbf{179})$
	量子	$(2743, 630, 167)$	$(2793, 690, 183)$	$(\mathbf{2543}, \mathbf{615}, \mathbf{163})$

表 7. 埃奎斯签名方案的整体安全强度

参数集名称	经典安全性	量子安全性
PARAMS I	99	90
PARAMS II	141	128
PARAMS II-b	141	128
PARAMS III	179	163

7 优缺点

总的来说，我们基于一个变种的(M)SIS困难问题和(M)LWE困难问题（即非对称(M)SIS问题和非对称(M)LWE问题）构造了一类高效的、强不可伪造的数字签名方案。在理论复杂性上，该类变种问题与相关标准问题在渐进意义下是等价的。在实际安全强度上，我们分析了已有针对(M)SIS问题和(M)LWE问题的最优攻击算法，提出了针对非对称(M)SIS问题和非对称(M)LWE问题的改进算法，给

出了具体安全参数的评估方法。具体分析和实验结果显示，与同类格上签名方案比较，我们利用非对称(M)SIS困难问题和非对称(M)LWE困难问题设计的埃奎斯签名方案能够实现更好的综合效率，达到更短的公钥和签名长度，提供更加灵活细粒度的参数选取。

优点. 我们提出的埃奎斯签名方案具有以下优点:

- **安全性高:** 埃奎斯签名方案在经典随机预言机模型和量子随机预言机模型下都是可证明强不可伪造安全的。此外，在参数选取和安全评估方面，我们使用了非常保守的安全评估模型（见第6节），从而能够在一定程度上抵抗未来改进的经典和量子攻击方法。
- **公私钥和签名长度短:** 与格上同类方案比较，埃奎斯签名方案具有更短的公私钥和签名长度。例如，对于保守评估安全强度达到128比特量子安全强度的参数集，埃奎斯签名方案的公钥、私钥和密文的长度分别为1312、3376和2445字节，比只达到125比特量子安全强度的Dilithium[21]签名方案（NIST第二轮候选签名方案之一）的公钥、私钥和签名长度分别短了160、128和256字节。
- **速度快:** 埃奎斯签名方案具有非常高效的密钥生成、签名和验证算法。例如，对于保守评估安全强度达到161比特量子安全强度的参数集，在安装Windows 10操作系统的普通笔记本（2.5 GHz CPU）上，标准C语言实现的密钥生成、签名生成和签名验证算法的运行时间分别只需大概0.27ms、0.98ms和0.26ms，而用AVX2指令部分优化实现的算法则分别降低到0.14ms、0.38ms和0.13ms。
- **参数选取灵活:** 与基于标准（M）SIS和(M)LWE困难问题的格上签名方案（例如[21]）比较，埃奎斯签名方案支持更加灵活细粒度的参数选取，从而更容易实现安全和性能的平衡。
- **抗多目标攻击:** 对于不同用户，埃奎斯签名方案采用不同的随机矩阵$\mathbf{A}$，从而阻止了攻击者以恢复一个用户私钥的代价来恢复多个用户的私钥。
- **抗密钥替换攻击:** 与我国SM2签名方案类似，通过在签名算法中绑定用户的公钥和消息来产生消息的签名，埃奎斯签名方案具有抵抗密钥替换攻击的能力。
- **易于安全实现:** 与许多格上签名方案（例如[20]）不同，埃奎斯签名方案没有使用高斯分布，从而能避免相关针对高斯分布采样算法的侧信道攻击。

缺点. 与许多格上高效的密码系统一样，埃奎斯签名方案采用了环结构。虽然我们使用的AMSIS问题和AMLWE问题有望能够提供比普通环上SIS问题和LWE问题更强的安全保证，但与标准LWE问题相比，环结构的使用仍有可能留给量子敌手更多攻击的空间。

8 Aigis-sig算法的适配性

与已有标准密码算法的适配性：

- Aigis-sig算法使用标准接口的杂凑函数和伪随机函数等对称密码组件来扩展内部随机数，实现或部署过程中可替换为满足相应安全强度的标准算法（例如，SM3杂凑函数和SM4分组密码等），共享已有对称密码组件的软硬件实现；
- 除了参数长度不同外，Aigis-sig算法与标准数字签名算法具有相同的接口，多数应用可以不做或稍微修改即可将已有数字签名算法替换成Aigis-sig算法；
- 与传统基于ECC或者RSA的传统公钥密码算法比较，Aigis-sig算法无需使用大整数操作，且只需要做小模数（向量）的加法和乘法运算，很容易使用SIMD指令来加速运算，计算效率高。

与抗量子候选密码算法的适配性：

- Aigis-sig签名算法与Aigis-enc密钥封装算法均使用了AMLWE问题，具有类似的割圆环结构和操作，能够共享AMLWE问题的安全评估方法，共享如快速傅里叶变换（NTT）等基本操作的软硬件代码；
- Aigis-sig签名算法与使用拒绝采样技术的签名算法相似的算法结构，相关可证明安全的结论可以共享；与NIST第二轮的NewHope、Kyber、Dilithium 等算法具有类似的数学结构，相关数学运算的优化技术能够共享。

参考文献

1. Damien Stehlé Adeline Langlois. Worst-case toaverage-casereductions for modulelattices. *Designs, Codes and Cryptography*, 75(3):565–599, 2015.
2. Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, STOC '96, pages 99–108, New York, NY, USA, 1996. ACM.
3. Martin R. Albrecht. On dual lattice attacks against small-secret lwe and parameter choices in helib and seal. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 103–129, Cham, 2017. Springer International Publishing.
4. Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. Revisiting the expected cost of solving usvp and applications to lwe. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, pages 297–322, Cham, 2017. Springer International Publishing.
5. Matin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *In Journal of Mathematical Cryptology*, 9:169–203, oct 2015.
6. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange—a new hope. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 327–343, Austin, TX, 2016. USENIX Association.

7. Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer Berlin Heidelberg, 2009.
8. Shi Bai and Steven D. Galbraith. An improved compression technique for signatures based on learning with errors. In Josh Benaloh, editor, *Topics in Cryptology – CT-RSA 2014*, volume 8366 of *LNCS*, pages 28–47. Springer International Publishing, 2014.
9. Shi Bai and Steven D. Galbraith. Lattice decoding attacks on binary lwe. In Willy Susilo and Yi Mu, editors, *Information Security and Privacy*, pages 322–337, Cham, 2014. Springer International Publishing.
10. Shi Bai and StevenD. Galbraith. An improved compression technique for signatures based on learning with errors. In Josh Benaloh, editor, *Topics in Cryptology – CT-RSA 2014*, volume 8366 of *Lecture Notes in Computer Science*, pages 28–47. Springer International Publishing, 2014.
11. Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In *Proceedings of the 13th ACM conference on Computer and Communications Security*, CCS '06, pages 390–399, New York, NY, USA, 2006. ACM.
12. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy Denning, Ray Pyle, Ravi Ganesan, Ravi Sandhu, and Victoria Ashby, editors, *First ACM Conference on Computer and Communication Security*, pages 62–73. ACM, November 3–5 1993.
13. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In DongHoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69. Springer Berlin Heidelberg, 2011.
14. J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle. Crystals - kyber: A cca-secure module-lattice-based kem. In *2018 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 353–367, April 2018.
15. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. Fully homomorphic encryption without bootstrapping. *Innovations in Theoretical Computer Science, ITCS*, pages 309–325, 2012.
16. Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 575–584, New York, NY, USA, 2013. ACM.
17. Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 505–524. Springer Berlin Heidelberg, 2011.
18. Yuanmi Chen. Lattice reduction and concrete security of fully homomorphic encryption. *Dept. Informatique, ENS, Paris, France, PhD thesis*, 2013.
19. Jung Hee Cheon, Duhyeong Kim, Joohee Lee, and Yongsoo Song. Lizard: Cut off the tail! a practical post-quantum public-key encryption from lwe and lwr. In Dario Catalano and Roberto De Prisco, editors, *Security and Cryptography for Networks*, pages 160–177, Cham, 2018. Springer International Publishing.
20. Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In Ran Canetti and JuanA. Garay, editors, *Advances in Cryptology – CRYPTO 2013*, volume 8042 of *Lecture Notes in Computer Science*, pages 40–56. Springer Berlin Heidelberg, 2013.

21. Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(1):238–268, Feb. 2018.
22. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In AndrewM. Odlyzko, editor, *Advances in Cryptology – CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer Berlin Heidelberg, 1987.
23. Ronald Aylmer Fisher and Frank Yates. Statistical tables for biological, agricultural and medical research. *journal*, (3rd ed.):26–27, 1938.
24. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC '08, pages 197–206, New York, NY, USA, 2008. ACM.
25. Shafi Goldwasser, Yael Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. In *Proceedings of the Innovations in Computer Science 2010*. Tsinghua University Press, 2010.
26. S.Dov Gordon, Jonathan Katz, and Vinod Vaikuntanathan. A group signature scheme from lattice assumptions. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 395–412. Springer Berlin / Heidelberg, 2010.
27. Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987.
28. Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of fiat-shamir signatures in the quantum random-oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, volume 10822 of *LNCS*, pages 552–586. Springer International Publishing, 2018.
29. Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In Aggelos Kiayias, editor, *Topics in Cryptology – CT-RSA 2011*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339. Springer Berlin Heidelberg, 2011.
30. Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 598–616. Springer Berlin / Heidelberg, 2009.
31. Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 738–755. Springer Berlin Heidelberg, 2012.
32. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23. Springer Berlin / Heidelberg, 2010.
33. Daniele Micciancio. On the hardness of learning with errors with binary secrets. *Theory of Computing*, 14(13):1–17, 2018.
34. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 372 – 381, 2004.
35. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37:267–302, April 2007.
36. Daniele Micciancio and Oded Regev. Lattice-based cryptography. In DanielJ. Bernstein, Johannes Buchmann, and Erik Dahmen, editors, *Post-Quantum Cryptography*, pages 147–191. Springer Berlin Heidelberg, 2009.

37. National Institute of Standards and Technology. Sha-3 standard: Permutation-based hash and extendable-output functions. FIPS PUB 202, 2015. http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf.
38. Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 333–342, New York, NY, USA, 2009. ACM.
39. Chris Peikert. An efficient and parallel gaussian sampler for lattices. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 80–97. Springer Berlin Heidelberg, 2010.
40. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
41. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, STOC '05, pages 84–93, New York, NY, USA, 2005. ACM.
42. C.P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.