木兰：高效、灵活、安全增强的格基签名

算法说明书

赵运磊（复旦大学）

巩博儒（复旦大学）

程蕾晓（复旦大学）

文黎明（银联商务股份有限公司）

谢群松（银联商务股份有限公司）

陈如钢（上海扈民区块链科技有限公司）

任思溟（银联商务股份有限公司）

吴洪祥（银联商务股份有限公司）

2019 年 10 月

摘要

根据目前格基密码的发展现状，普遍认为格基签名更难于设计和优化。在众多格基签名方案中，Dilithium 由于其在效率、公钥尺寸、抗侧信道攻击方面的性能优势成为 NIST 后量子密码标准征集中一个非常出众的格基签名方案，是进入 NIST 后量子密码标准征集第二轮的三个格基签名之一。Dilithium 可以视作基于 LWE 和 SIS 难题的实用格基签名的集大成者，其设计参考了前期格签名的众多经典之作，并精心挑选了参数以优化性能。能否在 Dilithium 已有的出色性能基础之上百尺竿头更进一步、能否提出更好的设计理念并在性能上有更好的权衡取舍被 Dilithum 的作者在[7]中留为一个重要的开放性问题。在本提案中，我们解决了这一开放性问题。

我们首先对 Dilithium 的设计给出新的解构和解读，并做通用和模块化的扩展。根据我们在 OKCN-KEX 提案中提出的密钥共识，来解读 Dilithium 的设计。在 OKCN-KEX 提案中，我们给出了最佳噪声密钥共识（OKCN）。通过将 OKCN 的确定版本移植到格基签名环境中，我们提出了 RKCN 机制，它可用于最大限度地压缩格基签名方案中的公钥长度。基于 RKCN，我们提出了基于通用和模块化结构 RKCN 的格基签名方案木兰（Mulan）。Mulan 的构造是通用且多功能的，并在量子随机预言机模型 QROM 下有可证明安全。相对于 Dilithium，木兰具有更一般的算法结构，更大的参数选取范围。特别地，木兰避免了 Dilithium 对参数（特别是 q）选取的限制。与 Dilithium 类似，Mulan 方案是利用 Fiat-Shamir 转换得到的，在 QROM 模型下满足 SEU-CMA 安全性。

RKCN 实际上是 OKCN 的确定版：相同的模块既用于数字签名，也用于 OKCN-KEX 密钥交换。据我们所知，这是第一个将格基数字签名和密钥交换的构建基于了同一个基础模块，这非常有利于模块化的软硬件优化实现，特别有利于硬件优化实现，并具有优异的适配性和兼容性。

通过大量的参数测试，在我们推荐的参数下木兰在安全性、公钥尺寸、私钥尺寸、签名尺寸、和计算效率等所有方面都超越 Dilithium。我们想强调的是，木兰相对于 Dilithium 的安全性和性能优势是建立在和 Dilithium 相同的评测标准上的。下表简要对比了 Mulan 签名方案 Dilithium [11]主要参数及性能指标。

	Mulan	Dilithium
q	1952257	8380417
n	256	256
(h,ℓ)	(5,4)	(5,4)
(η,η')	(2,2)	(5,5)
公钥长度（字节）	1312	1472
私钥长度（字节）	3056	3504
签名长度（字节）	2573	2701
重复次数	5.67	6.6
对抗密钥恢复攻击的量子比特开销	128	128
对抗伪造签名的量子比特开销	131	125

我们将 Mulan 方案推荐实例的参考版本的实现代码，和 NIST Round 2 Dilithium 方案推荐实例的参考版本的实现代码[21]，将相同的软件/硬件平台上进行性能测试比较。具体

地说，我们选用 Lenovo ThinkPad T480S 笔记本电脑（CPU 为 Intel(R) Core(TM) i7-8550U，内存在 16G），在 Ubuntu 18.04.3 LTS 系统下静态编译程序，使用 SHA-3 实例化方案中的 hash 函数。通过进行 100 万次的实验，实验效率对比如下表所示（单位：时钟周期）：

	Mulan	Dilithium
密钥生成函数	177707	198167
签名函数	859774	1056305
验证函数	191645	201511

由上表可见，我们的密钥生成函数、签名函数和验证函数，其运行速度均快于 Dilithium 的相应函数。特别地，与 Dilithium 相比，我们的签名函数的运行时间减少了约 19%，已超过由于循环次数降低所带来的潜在的效率优势（是除了循环次数减少、我们更小的 q 也对提升效率做了贡献）。

目录

1 引言

鉴于目前格密码的研究状况，人们普遍认为基于格的签名可能更微妙而且更难实现。例如，在 NIST PQC 标准化的第一轮筛选中有 20 多个格基 KEM 提案，但只有五个格基签名提案。其中，Dilithium[8,11] 因其简单，高效，公钥长度短和抗侧信道攻击，是最有希望的格签名候选者之一（目前入选 NIST PQC 标准化第二轮）。它的设计基于一系列前期的工作（例如，[15，16，4]等），并在实现和参数选择方面进行全面细致的优化。能否在 Dilithium 已有的出色性能基础上百尺竿头更进一步、能否提出更好的设计理念并在性能上有更好的权衡取舍被 Dilithum 的作者在[7]中留为一个重要的开放性问题。

在这项工作中，我们解决[7]中留下的问题。我们首先对 Dilithium 的设计给出新的解构和解读，并做通用和模块化的扩展。根据我们在 OKCN-KEX 提案中提出的密钥共识，来解读 Dilithium 的设计。在 OKCN-KEX 提案中，我们给出了最佳噪声密钥共识（OKCN）。通过将 OKCN 的确定版本移植到格基签名环境中，我们提出了 RKCN 机制，它可用于最大限度地压缩格基签名方案中的公钥长度。基于 RKCN，我们提出了基于 RKCN 的通用和模块化结构的格基签名方案木兰（Mulan）。Mulan 的构造是通用且多功能的，并在量子随机预言机模型 QROM 下有可证明安全。相对于 Dilithium，木兰具有更一般的算法结构，更大的参数选取范围。特别地，木兰避免了 Dilithium 对参数（特别是 q）选取的限制。与 Dilithium 类似，Mulan 方案是利用 Fiat-Shamir 转换得到的，在 QROM 模型下满足 SEU-CMA 安全性。

RKCN 实际上是 OKCN 的确定版：相同的模块既用于数字签名，也用于 OKCN-KEX 密钥交换。据我们所知，这是第一个将格基数字签名和密钥交换的构建基于了同一个基础模块，这非常有利于软硬件的模块化优化实现，特别有利于硬件优化实现，并具有优异的适配性和兼容性。

通过大量的参数测试，在我们推荐的参数下木兰在安全性、公钥尺寸、私钥尺寸、签名尺寸、和计算效率等所有方面都超越 Dilithium。我们想强调的是，木兰相对于 Dilithium 的安全性和性能优势是建立在和 Dilithium 相同的评测标准上的。

下表简要对比了 Mulan 签名方案 Dilithium [11]主要参数及性能指标。

Mulan 方案与 Dilithium 的对比

	Mulan	Dilithium
q	1952257	8380417
n	256	256
(h,ℓ)	(5,4)	(5,4)
(η,η')	(2,2)	(5,5)
公钥长度（字节）	1312	1472
私钥长度（字节）	3056	3504
签名长度（字节）	2573	2701
重复次数	5.67	6.6
对抗密钥恢复攻击的量子比特开销	128	128
对抗伪造签名的量子比特开销	131	125

我们将 Mulan 方案推荐实例的参考版本的实现代码，和 NIST Round 2 Dilithium 方案推荐实例的参考版本的实现代码[21]，将相同的软件/硬件平台上进行性能测试比较。具体地说，我们选用 Lenovo ThinkPad T480S 笔记本电脑（CPU 为 Intel(R) Core(TM) i7-8550U，内存在 16G），在 Ubuntu 18.04.3 LTS 系统下静态编译程序，使用 SHA-3 实例化方案中的 hash 函数。通过进行 100 万次的实验，实验效果对比如下表所示：

Mulan 方案实现与 Dilithium 方案实现的效率对比（单位：时钟周期）

	Mulan	Dilithium
密钥生成函数	177707	198167
签名函数	859774	1056305
验证函数	191645	201511

由上表可见，我们的密钥生成函数、签名函数和验证函数，其运行速度均快于 Dilithium 的相应函数。特别地，与 Dilithium 相比，我们的签名函数的运行时间减少了约 19%，已超过由于循环次数降低所带来的潜在的效率优势（是除了循环次数减少、我们更小的 q 也对提升效率做了贡献）。

2 预备知识

不失一般性，本文中的字符串都是二进制的。对于一个（二进制的）字符串 $s \in 0,1^*$，默认$|s|$表示该字符串的长度。对于任意实数$x \in \mathbb{R}$，令$\lfloor x \rfloor$表示不大于x的最大整数，且$\lfloor \quad x \rceil \coloneqq \lfloor x + 1/2 \rfloor$。对任意$i, j \in \mathbb{Z}$使得$i < j$，用$[i, j]$表示整数

$\{i,\ i+1,\cdots,j-1,j\}$的集合。对整数$r,\alpha>0$，令$r \bmod \alpha$表示唯一的整数$r' \in [0,\alpha-1]$使得$\alpha|(r'-r)$，$r \bmod^{\pm} \alpha$表示唯一的整数$r' \in \left[-\left\lfloor\frac{\alpha-1}{2}\right\rfloor, \left\lfloor\frac{\alpha}{2}\right\rfloor\right]$使得$\alpha|(r''-r)$。对于一个元素$x \in \mathbb{Z}_q$，我们用$|x \bmod \pm q|$表示$\|x\|_\infty$。

对于一个有限集合S，$|S|$表示集合的势，$x \leftarrow$ S表示从集合S中随机均匀地选择一个元素。我们使用下面的标准符号和惯例来表示概率算法、实验和交互协议。对于任意概率分布$\mathcal{D}$，$x \leftarrow \mathcal{D}$表示在预定义的分布$\mathcal{D}$中选择元素的操作。如果α既不是一个算法也不是一个集合，则$x \leftarrow \alpha$表示简单的赋值语句。如果A是一个概率算法，则$\mathrm{A}(x_1,x_2,\cdots;r)$表示在输入$x_1,x_2,\cdots$和随机数$r$的情况下运行 A 的结果。令$y \leftarrow \mathrm{A}(x_1,x_2,\cdots)$表示随机选择$r$并输出$y \coloneqq A(x_1,x_2,\cdots;r)$。$\Pr[R_1;\cdots R_n\colon \mathrm{E}]$表示在顺序执行随机过程$R_1;\cdots R_n$后事件E发生的概率。

2.1 模-LWE 与模-SIS 问题

在本文中，令$\mathrm{n} \geq 8$是 2 的乘方，$\mathrm{q} > 17$表示一个使得$2n|(q-1)$正的有理素数。在我们的 Mulan 签名方案中，我们一般令$n=256, q=1952257$。此外，$\mathcal{R}$和$\mathcal{R}_q$分别表示环$\mathbb{Z}[x]/\langle x^n+1\rangle$和$\mathbb{Z}_q[x]/\langle x^n+1\rangle$。对一个元素$w=\sum_{i=0}^{n-1} w_i x^i \in \mathcal{R}$，其$\ell_\infty$-范数定义为$\|w\|_\infty \coloneqq \max_i \|w_i\|_\infty$。类似地，对于一个元素$\boldsymbol{w}=(w_1,\cdots,w_k) \in \mathcal{R}^k$，它的$\ell_\infty$-范数定义为$\|\boldsymbol{w}\|_\infty \coloneqq \max_i \|w_i\|_\infty$。特别地，令$S_\eta$表示$\mathcal{R}$中的所有满足$\|w\|_\infty \leq \eta$的元素构成的集合。

我们的签名方案的安全性基于模-LWE(MLWE)和模-SIS(MSIS)困难问题。它们在[14]中已经得到了很好的研究，可以被分别看作是环-LWE[13]和环-SIS 问题的推广[12,19]。

固定一个参数$\ell \in \mathbb{N}$，模-LWE 分布(由$\mathrm{s} \in \mathcal{R}_q^\ell$引起的)是随机对$(\boldsymbol{a_i}, b_i)$在支集$\mathcal{R}_q^\ell \times \mathcal{R}_q$上的分布，其中$\boldsymbol{a_i} \leftarrow \mathcal{R}_q^\ell$是均匀的，$b_i \coloneqq \boldsymbol{a_i}^T \boldsymbol{s} + e_i$，每一个样本都由相同的$\boldsymbol{s} \leftarrow S_\eta^\ell$和不同的$e_i \leftarrow S_\eta$决定。给定任意多的取自模-LWE 分布的由$\boldsymbol{s} \leftarrow S_\eta^\ell$产生的样本，（搜索版本的）模-LWE 问题要求恢复$\boldsymbol{s}$。相应的模-LWE 假设指的是

给定$\mathbf{A} \leftarrow \mathcal{R}_q^{k\times\ell}$和$\mathbf{b} \coloneqq \mathbf{As}+\mathbf{e}$，其中$k=\text{poly}(\lambda)$，$(\boldsymbol{s},\boldsymbol{e}) \leftarrow S_\eta^\ell \times S_\eta^k$，假设参数适当选择，则没有概率多项式时间算法能够以不可忽略的概率恢复$\boldsymbol{s}$。类似地，给定$\mathbf{A} \leftarrow \mathcal{R}_q^{h\times\ell}$，其中$h=\text{poly}(\lambda)$，参数为$\beta>0$的模-SIS 问题要求找到一个原像$\mathbf{x} \in \mathcal{R}_q^{h+\ell}$使得$[\boldsymbol{A}|\boldsymbol{I}]\cdot\mathbf{x}=\mathbf{0}$且$\|\mathbf{x}\| \le \beta$。相应的模-SIS 假设指的是指在提供适当选择的参数的前提下，没有概率多项式时间算法能够以不可忽略的概率找到满足要求的原像$\mathbf{x}$。

3 RKCN:木兰签名方案的基础构建工具

在这一部分，我们给出确定性对称密钥（DKC）共识的定义。然后我们构造并分析一个称为带噪音的四舍五入的对称密钥共识（RKCN）的 DKC（Algorithm1），该算法是[9]中提出的带噪声的最优密钥共识方案（OKCN）的变体。基于 RKCN，我们给出了许多算法并提升了他们的一些性质。由于篇幅有限，本节中提出的定理的证明列于附录 A。RKCN 机制是 Mulan 签名方案的重要基础，也是该方案的创新点所在。

需要说明的是，这些算法可以以分量方式自然地推广到向量级操作中。

定义 1 一个 DKC 方案 DKC=(params,Con,Rec)，具体如下：

- $\mathbf{params}=(q,k,g,d,aux)$表示系统参数，其中$q,k,g,d$是满足$2 \le k,g \le q, 0 \le d \le \left\lfloor \frac{q}{2} \right\rfloor$的正整数，$aux$表示一些通常由$(q,k,g,d)$决定的辅助值，$aux$可被设为特殊符号$\emptyset$表示“空”。
- $(k_1,v) \leftarrow Con(\sigma_1,\mathbf{params})$：给定输入$(\sigma_1 \in \mathbb{Z}_q,\mathbf{params})$，确定性调和算法 Con 输出$(k_1,v)$，其中$k_1 \in \mathbb{Z}_k$是共识密钥，$v \in \mathbb{Z}_g$是一个提示信号，将公开传递给通信对方，以帮助双方达成共识。
- $k_2 \leftarrow \text{Rec}(\sigma_2,v,\mathbf{params})$：给定输入$(\sigma_2 \in \mathbb{Z}_q,v,\mathbf{params})$，确定性多项式时间调和算法 Rec 输出$k_2 \in \mathbb{Z}_k$。

正确性：一个 DKC 方案是正确的，如果对任意$\sigma_1,\sigma_2 \in \mathbb{Z}_q$满足$|\sigma_1-\sigma_2|_q \le d$，$k_1=k_2$成立。

我们给出了带噪音的四舍五入对称密钥共识（RKCN）的构造和分析，其示意图在算法 1 中给出。为简单起见，我们有时忽略"params"。请注意，如果我们正确设置参数，根据定理 1，RKCN 就是一个 DKC 方案。

Algorithm 1 RKCN: Rounded Symmetric KC with Noise

params $= (q, k, g, d, aux)$, $aux = \{q' = kq, \alpha = k, \beta = q\}$, q is prime
procedure CON(σ_1, params) ▷ $\sigma_1 \in [0, q-1]$
    $v = k\sigma_1 \bmod^{\pm} q$
    if $k\sigma_1 - v = kq$ **then**
        $k_1 = 0$
    else
        $k_1 = (k\sigma_1 - v)/q$
    end if
    return (k_1, v)
end procedure
procedure REC(σ_2, v, params) ▷ $\sigma_2 \in [0, q-1]$
    $k_2 = \lfloor (k\sigma_2 - v)/q \rceil \bmod k$
    return k_2
end procedure

定理 1 假设系统参数满足$2kd < q$，其中$k \geq 2$、$g \geq 2$，则这个 RKCN 方案是正确的。定理 1 的证明参见附录 A，以及所附的英文支持文档。

基于 RKCN，我们提出如下的一些算法。$\text{HighBits}_{q,k}$和$\text{LowBits}_{q,k}$是从 RKCN 的$\mathbf{Con}(r)$的输出中抽取r_1和r_0的程序。给定$r, z \in \mathbb{Z}_q$，为了从r, q, k中导出$\text{HighBits}_{q,k}(r+z)$，我们提出一个$\text{MakeHint}_{q,k}$的程序去产生一个 1 比特的提示$h$。程序$\mathbf{UseHint}_{q,k}$则说明如何使用提示$h$并恢复$\text{HighBits}_{q,k}(r+z)$。

procedure $\text{HIGHBITS}_{q,k}(r)$
    $(r_1, r_0) \leftarrow \text{Con}(r)$
    return r_1
end procedure

procedure $\text{LOWBITS}_{q,k}(r)$
    $(r_1, r_0) \leftarrow \text{Con}(r)$
    return r_0
end procedure

procedure $\text{MAKEHINT}_{q,k}(z, r)$
    $r_1 := \text{HighBits}_{q,k}(r)$
    $v_1 := \text{HighBits}_{q,k}(r+z)$
    if $r_1 = v_1$ **then**
        return 0
    else
        return 1
    end if
end procedure

procedure $\text{USEHINT}_{q,k}(h, r)$
    $(r_1, r_0) := \text{Con}(r)$
    if $h = 0$ **then**
        return r_1
    else if $h = 1$ and $r_0 > 0$ **then**
        return $(r_1 + 1) \bmod k$
    else
        return $(r_1 - 1) \bmod k$
    end if
end procedure

命题 1 对$r,z\in\mathbb{Z}_q$满足$\|z\|_\infty<\lfloor q/(2k)\rfloor$，我们有：

$$\mathrm{UseHint}_{q,k}\big(\mathrm{MakeHint}_{q,k}(z,r),r\big)=\mathrm{HighBits}_{q,k}(r+z)$$

证：对于$(r_1,r_0)\leftarrow Con(r)$和$(r_1',r_0')\leftarrow Con(r+z)$，有$0\le r_1,r_1'<k$，以及$\|r_0\|_\infty,\|r_0'\|_\infty\le q/2$。

由于$\|z\|_\infty<\lfloor q/(2k)\rfloor$，由定理 1 可知，$\mathrm{Rec}(r,r_0')=r_1'=\mathrm{HighBits}_{q,k}(r+z)$。

记$h\overset{def}{=}\mathrm{MakeHint}_{q,k}(z,r)$，由于

$$r_1'=\mathrm{Rec}(r,r_0')=\lfloor(kr-r_0')/q\rceil \bmod k=\lfloor r+(r_0-r_0')/q\rceil \bmod k\in\{r_1-1,r_1,r_1+1\}$$

当$r_0>0$时，则$\mathrm{Rec}(r,r_0')\in\{r_1,r_1+1\}$；当$r_0<0$时，则$\mathrm{Rec}(r,r_0')\in\{r_1-1,r_1\}$。由定义可知，$h=0$当且仅当$r_1=r_1'$。

所以，$\mathrm{HighBits}_{q,k}(r+z)=r_1'=\mathrm{Rec}(r,r_0')=\mathrm{UseHint}_{q,k}(h,r)$的正确性成立□

命题 2 对$r_1'\in\mathbb{Z}_k,r\in\mathbb{Z}_q,h\in\{0,1\}$，如果$r_1'=\mathrm{UseHint}_{q,k}(h,r)$，则$\|r-\lfloor q\cdot r_1'/k\rceil\|_\infty\le\frac{q}{k}+1/2$。

证：易知对于$(r_1,r_0)\leftarrow Con(r)$，有$r_1\in Z_k,r_0\in(-q/2,q/2)$，且存在$\theta\in\{0,1\}$使得$k\cdot r=(r_1+k\theta)\cdot q+r_0$。

若$h=0$，则$r_1'=r_1$，故$\left\|r-\lfloor q\cdot r_1'/k\rceil\right\|_\infty\le q/(2k)+1/2$。

若$h=1$且$r_0>0$，则$r_1'=r_1+1\bmod k$，故$\left\|r-\lfloor q\cdot r_1'/k\rceil\right\|_\infty\le q/k+1/2$。

若$h=1$且$r_0<0$，则$r_1'=r_1-1\bmod k$，故$\left\|r-\lfloor q\cdot r_1'/k\rceil\right\|_\infty\le q/k+1/2$。□

命题 3 对$r,z\in\mathbb{Z}$使得$\|z\|_\infty<U$。如果$\|r_0'\|_\infty<\frac{q}{2}-kU$，其中$(r_1,r_0)\leftarrow\mathrm{Con}(r)$，$(r_1',r_0')\leftarrow\mathrm{Con}(r+z)$，则$r_1=r_1'$。

证：

对于某些整数θ,θ'，由$k\cdot r=q\cdot(r_1+k\theta)+r_0$和$k\cdot(r+z)=q\cdot(r_1'+k\theta')+r_0'$，

易 知

$r_1 = \lfloor kr/q \rceil \bmod k = \lfloor k(r+z-z)/q \rceil \bmod k = \lfloor r_1' + (r_0' - kz)/q \rceil \bmod k = r_1'$。

4 Mulan 签名方案设计与分析

4.1 Mulan 签名方案设计

在本节中，我们将介绍 Mulan 签名方案的构造。简单地说，Mulan 使用 RKCN 对 Dilithium 进行了通用化和模块化扩展，它继承了 Dilithium 的正确性证明和安全性分析。

为了描述的简化，我们默认$n = 256$，$q = 1952257$，$\mathcal{R} = \mathbb{Z}[x]/\langle x^n + 1\rangle$以及$\mathcal{R}_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$。$S_\eta = \{a \in \mathcal{R}_q \mid \|a\|_\infty \le \eta\}$，$B_{60} \subseteq \mathcal{R}_q$是$\mathcal{R}_q$中元素的集合，并且恰好有 60 个系数取值为-1 或 1，其余取值为 0。令H:$\{0,1\}^* \to B_{60}$为一个哈希函数，在我们的签名方案中模型化为随机预言机。Sam(·)为一个可扩展输出函数（详见附录 C）。最后，对每个$\mathrm{a} = \sum_{i=0}^{n-1} a_i \cdot x^i \in \mathcal{R}_q$，$a_i \in \mathbb{Z}_q$，定义$\mathrm{Power2Round}_{q,d}(a) \stackrel{\text{def}}{=} (\sum a_i' \cdot x^i, \sum a_i'' \cdot x^i)$，其中$a_i' \stackrel{\text{def}}{=} \frac{(a_i - a_i'')}{2^d}$，$a_i'' = (a_i mod^{\pm} 2^d)$。该定义可以自然的按照分量应用的方式进行推广。

Mulan 签名方案由$q, k, n, h, \ell, d, \omega, \eta, \eta', U$参数确定。它的密钥生成、签名和验签算法分别描述如下。

Algorithm 2 Key Generation Algorithm

1: **Input**: 1^λ
2: **Output**: $(\mathrm{pk} = (\rho, \mathbf{t}_1), \mathrm{sk} = (\rho, \mathbf{s}, \mathbf{e}, \mathbf{t}))$
3: $\rho, \rho' \leftarrow \{0,1\}^{256}$
4: $\mathbf{A} \in \mathcal{R}_q^{h \times \ell} := \mathsf{Sam}(\rho)$
5: $(\mathbf{s}, \mathbf{e}) \in S_\eta^\ell \times S_\eta^h := \mathsf{Sam}(\rho')$
6: $\mathbf{t} := \mathbf{A}\mathbf{s} + \mathbf{e}$
7: $\mathbf{t}_1 := \mathsf{Power2Round}_{q,d}(\mathbf{t})$
8: **return** $(\mathrm{pk} = (\rho, \mathbf{t}_1), \mathrm{sk} = (\rho, \mathbf{s}, \mathbf{e}, \mathbf{t}))$

Algorithm 3 The Signing Algorithm

Input: $\mu \in \{0,1\}^*$, $\mathrm{sk} = (\rho, \mathbf{s}, \mathbf{e}, \mathbf{t})$
Output: $\sigma = (\mathbf{z}, c, \mathbf{h})$
$\mathbf{A} \in \mathcal{R}_q^{h \times \ell} := \mathsf{Sam}(\rho)$
$\mathbf{t}_1 := \mathsf{Power2Round}_{q,d}(\mathbf{t})$
$\mathbf{t}_0 := \mathbf{t} - \mathbf{t}_1 \cdot 2^d$
$r \leftarrow \{0,1\}^{256}$
$\mathbf{y} \in S^{\ell}_{\lfloor q/k \rfloor - 1} := \mathsf{Sam}(r)$
$\mathbf{w} := \mathbf{A}\mathbf{y}$
$\mathbf{w}_1 := \mathsf{HighBits}_{q,k}(\mathbf{w})$
$c \leftarrow H(\rho, \mathbf{t}_1, \lfloor q \cdot \mathbf{w}_1 / k \rceil, \mu)$
$\mathbf{z} := \mathbf{y} + c\mathbf{s}$
$(\mathbf{r}_1, \mathbf{r}_0) := \mathsf{Con}(\mathbf{w} - c\mathbf{e})$
Restart if $\|\mathbf{z}\|_\infty \geq \lfloor q/k \rfloor - U$ or $\|\mathbf{r}_0\|_\infty \geq q/2 - kU$ or $\mathbf{r}_1 \neq \mathbf{w}_1$
$\mathbf{h} := \mathsf{MakeHint}_{q,k}(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{e} + c\mathbf{t}_0)$
Restart if $\|c\mathbf{t}_0\|_\infty \geq \lfloor q/2k \rfloor$ or the number of 1's in $\mathbf{h}$ is greater than ω
return $(\mathbf{z}, c, \mathbf{h})$

Algorithm 4 Verification Algorithm

Input: $\mathrm{pk} = (\rho, \mathbf{t}_1), \mu \in \{0,1\}^*, (\mathbf{z}, c, \mathbf{h})$
Output: $b \in \{0,1\}$
$\mathbf{A} \in \mathcal{R}_q^{h \times \ell} := \mathsf{Sam}(\rho)$
$\mathbf{w}'_1 := \mathsf{UseHint}_{q,k}(\mathbf{h}, \mathbf{A}\mathbf{z} - c\mathbf{t}_1 \cdot 2^d)$
$c' \leftarrow H(\rho, \mathbf{t}_1, \lfloor q\mathbf{w}'_1 / k \rceil, \mu)$
if $c = c'$ and $\|\mathbf{z}\|_\infty < \lfloor q/k \rfloor - U$ and the number of 1's in $\mathbf{h}$ is $\leq \omega$ **then**
    return 1
else
    return 0
end if

需要说明的是，为了简化方案的描述，我们遵循 HNF-LWE 问题及其变体的定义要求，要求其中的秘密 $\mathbf{s}$ 和噪音 $\mathbf{e}$ 服从相同的分布。但从方案具体参数设置的角度出发，我们可以取秘密 $\mathbf{s}$ 和噪音 $\mathbf{e}$ 服从不同的分布，例如：我们可以取秘密 $\mathbf{s}$ 的每个系数独立地服从 $\{-\eta, 1-\eta, \cdots, \eta-1, \eta\}$ 上的均匀分布，而让噪音 $\mathbf{e}$ 的每个系数独立地服从 $\{-\eta', 1-\eta', \cdots, \eta'-1, \eta'\}$ 上的均匀分布。这有利于扩大参数的选择范围，从而使得我们可以在效率和安全性方面取得更好的折衷。

在 Mulan 签名方案中，密钥生成算法首先随机选择一个 256 比特的种子 ρ，并通过模型化为随机预言机的可扩展函数 $\mathsf{Sam}(\cdot)$ 将其扩展成矩阵 $\mathbf{A} \leftarrow \mathcal{R}_q^{h \times l}$。相反地，私钥最重要的组成部分为 $(s, e) \in \mathcal{R}_q^h \times \mathcal{R}_q^{\ell}$，并且 s（分别地，e）的每个系数都从集合 $[-\eta, \eta]$（分别地，$[-\eta', \eta']$）中随机均匀选取。最后，我们计算 $\mathbf{t} := \mathbf{A}\mathbf{s} + \mathbf{e} \in \mathcal{R}_q^h$。公钥为 $\mathrm{pk} = \left(\rho, \boldsymbol{t}_1 = \mathit{P}\mathrm{ower2Round}_{q,d}(\boldsymbol{t})\right)$，其对应的私钥为 $\mathrm{sk} = (\rho, \mathbf{s}, \mathbf{e}, \mathbf{t})$。

给定私钥 $\mathrm{sk} = (\rho, s, e, t)$ 和需要签名的消息 $\mu \in \{0,1\}^*$，签名算法首先通过私钥中的随机种子 ρ 恢复矩阵 $\mathbf{A} \in \mathcal{R}_q^{h \times \ell}$。随后签名算法从集合 $S^{\ell}_{\lfloor q/k \rfloor} \subseteq \mathcal{R}_q^{\ell}$ 中均匀随机地选择一个短的 y，并计算 $\mathbf{w}_1 := \mathrm{HighBits}_{q,k}(\mathrm{w})$，其中 $\mathrm{w} := \mathrm{Ay}$。给定输入 $(\rho, t_1, \lfloor q \cdot \mathrm{w}_1 / k \rceil, \mu)$ 后，随机预言机 $\mathrm{H}(\cdot)$ 返回集合 $\mathrm{B}_{60} \subseteq \mathcal{R}_q$ 中的一个随机值 c（见

附录 D）。获得c之后，签名算法通过拒绝抽样（见附录 E）来确定是否$\mathbf{z} \coloneqq \mathbf{y} + c\mathbf{s} \in \mathcal{R}_q^\ell$的每一个系数都足够“小”，是否$\boldsymbol{r}_0$的每一个系数都足够小且是否$\mathbf{r}_1 = \mathbf{w}_1$，其中$(\mathbf{r}_1, \mathbf{r}_0) \leftarrow \mathrm{Con}(\mathbf{w} - c\boldsymbol{e})$；如果不是，算法重新运行直至所有的前提条件都满足。我们应该指出如果$\|c\boldsymbol{e}\|_\infty < U$，则根据命题 3，$\|\mathbf{r}_0\|_\infty < \frac{q}{2} - kU$意味着$\mathbf{r}_1 = \boldsymbol{w}_1$。我们希望$\|c\boldsymbol{e}\|_\infty < U$发生的概率是可忽略的，使得使用最后一个验证条件（$\mathbf{r}_1 = \boldsymbol{w}_1$）的概率也是可忽略的。此外，算法输入$(-c\boldsymbol{t}_0, \mathrm{w} - c\mathbf{e} + c\boldsymbol{t}_0)$调用$\mathrm{MakeHint}_{q,k}(\cdot)$生成提示$\mathbf{h}$，即$\{0,1\}^{\boldsymbol{n}\cdot\boldsymbol{h}}$中的一个二进制向量。最后，签名进行第二次拒绝采样，即是否$\|c\boldsymbol{t}_0\|_\infty \geq \lfloor q/2k \rfloor$且$\mathbf{h} \in \{0,1\}^{\boldsymbol{n}\cdot\boldsymbol{h}}$中的非零元素的个数没有超过预先设置的阈值ω；如果不是，则算法再次重新重启。这里，提示$\mathbf{h}$对应的是公钥中的$\boldsymbol{t}_1$，而不是整个的$\mathbf{t} = \boldsymbol{t}_1 \cdot 2^d + \boldsymbol{t}_0$。故即使没有$\boldsymbol{t}_0$，我们仍可以用提示$\mathbf{h}$来进行验签。

给定公钥$\mathrm{pk} = (\rho, \boldsymbol{t}_1)$、消息$\mu \in \{0,1\}^*$和签名$(\boldsymbol{z}, c, \mathbf{h})$，验证算法首先通过随机种子恢复公钥矩阵$\mathbf{A} \in \mathcal{R}_q^{h\times\ell}$。然后计算$w_1' = \mathrm{UseHint}_{q,k}\left(\mathbf{h}, \mathbf{A}\boldsymbol{z} - c\mathbf{t}_1 \cdot \mathbf{2^d}\right)$。如果给定的$(\boldsymbol{z}, c, \mathbf{h})$确实是关于消息μ的一个有效签名，则$\mathbf{z}$的每一个系数都足够小，且$\mathbf{h}$中 1 的个数小于等于ω；更重要的是，$\mathrm{HighBits}_{q,k}(\mathbf{A}\mathbf{y}) = \mathrm{HighBits}_{q,k}(\mathbf{A}\mathbf{y} - c\boldsymbol{e}) = w_1'$，因此$c = c'$，其中$c' \leftarrow H(\rho, \boldsymbol{t}_1, \lfloor qw_1'/k \rceil, \mu)$。当且仅当上述条件全部满足时验签算法接受给定输入是一个有效签名。

接下来，我们将证明只要适当地设置参数，我们的签名方案就总是正确的。粗略地说，正确性很大程度上依赖于性质 1-3。

当公/私钥对（pk,sk）固定时，对于一个有效的信息/签名对$\left(\mu, (\boldsymbol{z}, c, \mathbf{h})\right)$，首先可以看出$\|\boldsymbol{z}\|_\infty < \lfloor q/k \rceil - U$，且$\mathbf{h}$中非零的数目小于等于ω。此外，因为$\|c\boldsymbol{t}_0\|_\infty < q/2k$、$\mathbf{A}\mathbf{z} - \mathrm{c}\boldsymbol{t}_1 \cdot 2^d = \mathbf{A}\mathbf{y} - c\boldsymbol{e} + c\boldsymbol{t}_0$，根据命题 1 有：

$$\mathrm{UseHint}_{q,k}(\mathbf{h}, \mathbf{A}\mathbf{z} - ct_1 \cdot 2^d) = \mathrm{HighBits}_{q,k}(\mathbf{A}\mathbf{y} - c\mathbf{e})$$

考虑到签名算法通过拒绝抽样强制要求$\mathrm{HighBits}_{q,k}(\mathbf{A}\mathbf{y} - c\mathbf{e}) = \mathrm{HighBits}_{q,k}(\mathbf{A}\mathbf{y})$，所以有：

$$\mathrm{UseHint}_{q,k}(\mathbf{h}, \mathbf{A}\mathbf{z} - c\boldsymbol{t}_1 \cdot 2^d) = \mathrm{HighBits}_{q,k}(\mathbf{A}\mathbf{y} - c\mathbf{e}) = \mathrm{HighBits}_{q,k}(\mathbf{A}\mathbf{y})$$

因此$c = c'$。以上就完成了木兰签名方案的正确性分析。

4.2 Mulan 安全性分析，和 QROM 模型下紧规约可证安全

首先，Mulan 方案继承了 Dilithium[8,11]的安全性证明：在 module-LWE 和 module-SIS$^{\infty}$假设下，当参数适当取值的时候，通过使用 forking lemma，我们可以证明在经典的 ROM 模型（classical random oracle model）模型中，Mulan 签名方案满足 SEU-CMA 安全性（strong existential unforgeability under chosen message attack）。

对本文的签名方案来说，应考虑两种重要类型的攻击：给定相关公钥试图恢复密钥的密钥恢复攻击(key-recovery attack)；在 SEU-CMA（见定义 2）安全游戏中试图伪造签名的伪造攻击(forgery attack)。当考虑抗密钥恢复攻击的 Mulan 安全时，如[8, 11]我们假设公钥包含 t(而不是 t1)。我们使用和[8, 11]提出的相同方法衡量具体的抗伪造攻击安全，具体细节在附录 F 和附录 G 给出。关于该方案的详细安全性分析/证明，请参见附录 G 和本提案的“支持文档”。

上述 ROM 模型下的可证明安全由于使用了 forking lemma，安全规约过程是不紧（non-tight）的；同时，由于使用了 forking lemma，该经典规约证明并不能直接推广到量子计算模型下。尽管如此，到目前为止，所有通过使用 forking lemma 得到安全性证明的签名方案，都没有因为其规约过程是不紧的而降低其安全性。其原因在于：证明过程中存在着一个中间状态的问题，即使在量子规约下，从签名方案的 UF-CMA 安全性到该中间状态问题的规约证明也是紧的（tight）。在某种程度上，该中间状态问题是结合底层数学问题和 hash 函数的混合问题。到目前为止，所有的研究表明，只要底层数学问题的代数结构和 hash 函数没有任何直接关系，那么解决这个中间状态问题就不会比解决底层的数学问题更加容易。

对于木兰签名方案而言，它所对应的这个中间状态的问题是 SelfTargetSIS 问题。具体地说，对于 hash 函数 $H:\{0,1\}^{*}\to B_{60}$ 而言，相应的 SelfTargetSIS 问题是：给定矩阵 $\mathbf{A}\leftarrow \mathrm{R}_q^{m\times k}$ 以及量子访问 $H(\cdot)$ 的权限，要求输出 $\left(\mathbf{y}:=[\mathbf{r},c]^{t},\mu\right)$，使得 $0\le \|\mathbf{y}\|_{\infty}\le\gamma$ 并且 $H(\mu,[\mathbf{I},\mathbf{A}]\cdot\mathbf{y})=c$。

类似于[10]的结论，在量子随机预言机（QROM）模型下，木兰签名方案的

安全性可以被量子规约到 SelfTargetSIS 问题，并且该量子规约是紧的。

5 Mulan 签名方案的实现

5.1 Mulan 签名方案的推荐参数

在为 Mulan 方案选择推荐参数时，我们同事考虑下列要求和目标：

- 应选择能确保签名体制正确的适当参数。
- 应根据 128 比特量子安全的目标来选择相关参数。
- 参数的选择应该能够使签名算法中预期的重复次数尽可能少，保证签名算法的效率。
- 应选择使得参数，使得公钥长度和签名长度之和尽可能地小。

基于以上考虑，我们为 Mulan 签名方案选择的推荐参数集如下表 1 所示。

表 1　Mulan 签名方案的推荐参数集

	q	n	(h,ℓ)	(η,η')	d	k	(U,U)	ω
建议参数	1952257	256	(5,4)	(2,2)	13	8	118	96

需要注意的是，在我们的签名体制中，公钥的长度是$32+h\cdot 32\cdot(\lceil\log(q)-d\rceil)$字节，签名的长度是$\ell\cdot 32\cdot\lceil\log(2\cdot\lfloor q/k\rfloor)\rceil+(32+8)+(\omega+h-1)$字节。预期的重复次数取决于两个拒绝采样步骤发生的概率。重启的概率大致为

$$\left(\frac{2(\lfloor q/k\rfloor-U)-1}{2\lfloor q/k\rfloor-1}\right)^{\ell\cdot n}\cdot\left(\frac{2(\lfloor q/2\rfloor-kU)-1}{q}\right)^{h\cdot n}$$

其中，参数 U 需谨慎选择，从而使得$\Pr[\|c\mathbf{e}\|_\infty\geq U]\leq 2^{-128}$。

对于第二次重启，我们用实验估计预期的重复次数和应选择的参数，使得在实验中第二次重新开始以不超过 1%的概率进行。

对本文的签名方案来说，应考虑两种重要类型的攻击：给定相关公钥试图恢复密钥的密钥恢复攻击(key-recovery attack)；在 SEU-CMA（见定义 2）安全游戏中试图伪造签名的伪造攻击(forgery attack)。当考虑抗密钥恢复攻击的 Mulan 安全时，如[8, 11]我们假设公钥包含 t(而不是 t1)。我们使用和[8, 11]提出的相同方法衡量具体的抗伪造攻击安全，具体细节在附录 F 和附录 G 给出。

表 2 简要对比了 Mulan 签名方案 Dilithium [11]主要参数及性能指标。

表 2　Mulan 方案与 Dilithium 的对比

	Mulan	Dilithium
q	1952257	8380417
n	256	256
(h,ℓ)	(5,4)	(5,4)
(η,η')	(2,2)	(5,5)
公钥长度（字节）	1312	1472
私钥长度（字节）	3056	3504
签名长度（字节）	2573	2701
重复次数	5.67	6.6
对抗密钥恢复攻击的量子比特开销	128	128
对抗伪造签名的量子比特开销	131	125

关于表 1 的说明：类似于 Dilithium 方案，针对 Mulan 签名方案的伪造签名攻击的问题，可以被规约为无穷范数下的 Module-SIS 问题，该问题的难度被等价为 Mulan 签名方案在推荐参数下的抗伪造签名量子安全强度。如果我们采用非对称的形式去分析，该无穷范数 Module-SIS 问题的难度估值为 131；另一方面，如果我们采用对称的形式去分析(Dilithium 方案正式采用该形式进行分析)，则得到该无穷范数 Module-SIS 问题的难度估值为 128。

通过表 1 中的性能参数对比，不难看出，与 Dilithium 方案相比，Mulan 方案无论是在空间效率（即公钥长度和签名长度之和），还是在安全性方面，都具有一定的优势。

5.2 Mulan 签名的具体实现

在表 1 下的推荐参数下，我们实现了 Mulan 签名方案的具体实例。和 Dilithium 的实现类似，在不降低 Mulan 签名实例的安全强度的情况下，我们在实现过程中对方案的具体结构做了一定的调整，以提高程序的执行效率。为了便于评审，我们详细描述我们的具体实现。

Mulan 签名实例的具体结构如下的算法 5-7 所示：

Algorithm 5 Key Generation: $(\mathsf{pk} = (\rho, \mathbf{t}_1), \mathsf{sk} = (\rho, \mathsf{key}, \mathsf{tr}, \mathbf{s}, \mathbf{e}, \mathbf{t}_0)) \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\rho \leftarrow \{0,1\}^{256}$
$\mathsf{key} \leftarrow \{0,1\}^{256}$
$(\mathbf{s}, \mathbf{e}) \in S_\eta^\ell \times S_{\eta'}^h$
$\mathbf{A} \in \mathcal{R}_q^{h \times \ell} := \mathsf{ExpandA}(\rho)$ ▷ $\mathbf{A}$ is generated and stored in its NTT representation $\hat{\mathbf{A}}$.
$\mathbf{t} := \mathbf{A}\mathbf{s} + \mathbf{e}$ ▷ $\mathbf{t} := \mathsf{NTT}^{-1}(\hat{\mathbf{A}} \cdot \mathsf{NTT}(\mathbf{s})) + \mathbf{e}$
$(\mathbf{t}_1, \mathbf{t}_0) := \mathsf{Power2Round}_{q,d}(\mathbf{t})$
$\mathsf{tr} \in \{0,1\}^{384} := \mathsf{CRH}(\rho, \mathbf{t}_1)$
return $(\mathrm{pk} = (\rho, \mathbf{t}_1), \mathrm{sk} = (\rho, \mathsf{key}, \mathsf{tr}, \mathbf{s}, \mathbf{e}, \mathbf{t}_0))$

Algorithm 6 Sign: $(\mathbf{z}, c, \mathbf{h}) := \mathsf{Sign}(\mathsf{sk}, \mathsf{msg})$

$\mathbf{A} \in \mathcal{R}_q^{h \times \ell} := \mathsf{ExpandA}(\rho)$ ▷ $\mathbf{A}$ is generated and stored in its NTT representation $\hat{\mathbf{A}}$.
$\mu \in \{0,1\}^{384} := \mathsf{CRH}(\mathsf{tr}, \mathsf{msg})$
$\kappa := 0$
while true do
  $\mathbf{y} \in S_{\lfloor q/k \rfloor - 1}^\ell := \mathsf{ExpandMask}(\mathsf{key}, \mu, \kappa)$
  $\kappa := \kappa + 1$
  $\mathbf{w} := \mathbf{A}\mathbf{y}$ ▷ $\mathbf{w} := \mathsf{NTT}^{-1}(\hat{\mathbf{A}} \cdot \mathsf{NTT}(\mathbf{y}))$
  $\mathbf{w}_1 := \mathsf{HighBits}_{q,k}(\mathbf{w})$
  $c \in B_{60} := \mathsf{H}(\mu, \mathbf{w}_1)$
  $\mathbf{z} := \mathbf{y} + c\mathbf{s}$ ▷ $\mathbf{z} := \mathbf{y} + \mathsf{NTT}^{-1}(\mathsf{NTT}(c) \cdot \mathsf{NTT}(\mathbf{s}))$
  $\mathbf{u} := \mathbf{w} - c\mathbf{e}$
  $(\mathbf{r}_1, \mathbf{r}_0) := \mathsf{Con}(\mathbf{u})$
  if $\|\mathbf{z}\|_\infty \geq \lfloor q/k \rfloor - U$ **then** ▷ Check #1
    continue
  end if
  if $\|\mathbf{r}_0\|_\infty \geq \lfloor q/2 \rfloor - k \cdot U'$ **then** ▷ Check #2
    continue
  end if
  if $\mathbf{r}_1 \neq \mathbf{w}_1$ **then** ▷ Check #3
    continue
  end if
  $\mathbf{v} := c \cdot \mathbf{t}_0$ ▷ $\mathbf{v} := \mathsf{NTT}^{-1}(\mathsf{NTT}(c) \cdot \mathsf{NTT}(\mathbf{t}_0))$
  if $\|\mathbf{v}\|_\infty \geq \lfloor q/(2k) \rfloor$ **then** ▷ Check #4
    continue
  end if
  $\mathbf{h} := \mathsf{MakeHint}_{q,k}(-\mathbf{v}, \mathbf{u} + \mathbf{v})$
  if $\|\mathbf{h}\|_1 \geq \omega$ **then** ▷ Check #5
    continue
  end if
  break
end while
return $(\mathbf{z}, c, \mathbf{h})$

Algorithm 7 Verification: $b := \mathsf{Verify}(\mathsf{pk}, \mathsf{msg}, (\mathbf{z}, c, \mathbf{h})$

$\mathbf{A} \in \mathcal{R}_q^{h \times \ell} := \mathsf{ExpandA}(\rho)$ ▷ $\mathbf{A}$ is generated and stored in its NTT representation $\hat{\mathbf{A}}$.
$\mathsf{tr} \in \{0,1\}^{384} := \mathsf{CRH}(\rho, \mathbf{t}_1)$
$\mu \in \{0,1\}^{384} := \mathsf{CRH}(\mathsf{tr}, \mathsf{msg})$
$\mathbf{w}_1' := \mathsf{UseHint}_{q,k}(\mathbf{h}, \mathbf{Az} - c\mathbf{t}_1 \cdot 2^d)$
$c' \leftarrow H(\rho, \mathbf{t}_1, \mathbf{w}_1', \mu)$
if $c = c'$ and $\|\mathbf{z}\|_\infty < \lfloor q/k \rfloor - U$ and $\|\mathbf{h}\|_1 \leq \omega$ **then**
    $b := 1$
else
    $b := 0$
end if
return b

具体而言，与第 4 章中 Mulan 方案的表述相比，我们在实现细节做了如下调整和优化。

首先，与 Dilithium 类似，我们将签名算法 Sign()修改成了确定化的算法。具体地说，对于 Mulan 方案签名算法中（唯一的）随机向量$\mathbf{y}$，我们根据伪随机生成器来确定性地生成$\mathbf{y}$。其中，伪随机生成器的随机种子包括三部分：一部分来自私钥，一部分来自待签名信息的 hash 值，最后一部分是计数器。这一调整在现实中并未降低方案实例的安全强度，但却大大加速了签名函数的运行速度。这也是 Dilithium 的实现技巧（Dilithium 的学术论文描述的随机版本，而 NIST 提交和实现的确定版本）。

其次，与 Dilithium 类似，在签名算法 Sign()中，在生成c时，我们并未将c设置为$c \leftarrow H(\mathsf{msg}, q/k \cdot \mathbf{w}_1)$，而是设置为$c \leftarrow H(\mathsf{msg}, \mathbf{w}_1)$。同样，该调整在未降低方案实例安全强度的同时，有利于提高签名函数的运行速度。

再次，类似于 Dilithium，我们采用 Fisher-Yates shuffle 的一个变体，来实现在B_{60}集合的均匀分布中抽取c元素。

最后，由于 Mulan 方案是构建在模格的基础上的，为了提高程序的整体运行速度，我们采用了 NTT（number-theoretic transform）技术，来加速实现模格上元素之间的乘法运算。具体地说，在我们在推荐参数下，模格是构建在商环$\mathrm{R}_q = \mathbb{Z}_q[x]/(x^n+1)$上的，其中有限域$\mathbb{Z}_q$上恰好存在$2n$阶的原根$\omega$，使得主理想$(x^n+1)$的生成元$x^n+1$满足$x^n+1 = (x-\omega)(x-\omega^3)\cdots(x-\omega^{2n-1}) \pmod{q}$。由此，根

据中国剩余定理（Chinese Remainder Theorem, CRT）我们首先通过插值计算每个多项式 $f \in \mathrm{R}_q$ 在 n 个不同元素下函数值所组成的向量 $\hat{f} = (f(\omega), f(\omega^3), \cdots, f(\omega^{2n-1}))$，它被称为 $f \in \mathrm{R}_q$ 的 NTT 表示。利用 $f, g \in \mathrm{R}_q$ 的 NTT 表示，我们可以轻松计算 $\hat{h} = \hat{f} \cdot \hat{g}$，并进一步通过利用中国剩余定理来计算得到 $h = f \cdot g$。

5.3 Mulan/Dilithium 推荐实例的参考实现的性能比较

我们将 Mulan 方案推荐实例的参考版本的实现代码，和 NIST Round 2 Dilithium 方案推荐实例的参考版本的实现代码[21]，将相同的软件/硬件平台上进行性能测试比较。具体地说，我们选用 Lenovo ThinkPad T480S 笔记本电脑（CPU 为 Intel(R) Core(TM) i7-8550U，内存在 16G），在 Ubuntu 18.04.3 LTS 系统下通过静态编译方式，使用 SHA-3 实例化方案中的 hash 函数。通过进行 100 万次的实验，实验效果对比如下：

表 3　Mulan 方案实现与 Dilithium 方案实现的效率对比（单位：时钟周期）

	Mulan	Dilithium
密钥生成函数	177707	198167
签名函数	859774	1056305
验证函数	191645	201511

由此可见，我们的密钥生成函数、签名函数和验证函数，其运行速度均快于 Dilithium 的相应函数。特别地，与 Dilithium 相比，我们的签名函数的运行时间减少了约 19%，已超过由于循环次数降低所带来的潜在的效率优势（主要是除了循环次数减少、我们更小的 q 也对提升效率做了贡献）。

5.4 Mulan 签名方案推荐参数的具体安全强度分析

对本文的签名方案来说，应考虑两种重要类型的攻击：给定相关公钥试图恢复密钥的密钥恢复攻击(key-recovery attack)；在 SEU-CMA（见定义 2）安全游戏中试图伪造签名的伪造攻击(forgery attack)。当考虑抗密钥恢复攻击的 Mulan 安全时，如[8, 11]我们假设公钥包含 t(而不是 t_1)。我们使用和[8, 11]提出的相同方法衡量具体的抗伪造攻击安全，具体细节在附录 F 和附录 G，以及本提案的支持

文档。

6 实现代码、及性能测试

Mulan 方案的算法实现代码在“算法实现源代码”下，其中我们的算法实现代码包括两部分：

1. “木兰 动态库参考实现”，包括 Mulan 格签名方案在推荐参数下的 Windows 版参考实现（C 语言）；
2. “木兰 动态库优化实现”：包括 Mulan 格签名方案在推荐参数下的 Windows 版优化实现（C 语言+AVX2 优化实现）。

另外，根据算法竞赛第二轮提交要求提供的测试环境说明，我们还在提供了“木兰动态库优化实现”的测试报告，其中详细给出了程序的测试环境、详细的测试用例、测试结果和基于测试结果的性能分析。

6.1 空间性能

在推荐参数（见表 1）下，木兰格签名实现的主要输入输出参数所耗空间如表 4 所示：

表 4 主要参数所耗空间

名称	公钥 pk	私钥 sk	签名 sm
长度(byte)	1312	3056	2573

6.2 时间性能

按照算法竞赛要求的编译方式，我们通过调用算法动态链接库的编程接口进行性能测试。详细的硬件测试环境及性能测试结果如表 5 所示：

表 5 木兰签名算法效率自测试结果记录

SIG 方案	平均运行效率(ms/次)	测试平台
密钥生成函数	0.165	设备类型：微型计算机 型号：HP Pavilion Gaming Desktop PC CPU：Intel® Core™ i7-8700 CPU @ 3.20GHz 内存：RAM 8GB(1*8GB) DDR4 2666 NECC 硬盘：HDD 1TB 7200RPM SATA 3.5 cDT 2nd SSD 128G 2280 PCIe NVMe Value 软件环境：Windows 10。 Microsoft Visual Studio Community2017
签名函数	0.398	
验证函数	0.106	

7 创新性、特色、和优缺点声明

根据目前格基密码的发展现状，普遍认为格基签名更难于设计和优化。在众多格基签名方案中，Dilithium 由于其在效率、公钥尺寸、抗侧信道攻击方面的性能优势成为 NIST 后量子密码标准征集中一个非常出众的格基签名方案，是进入 NIST 后量子密码标准征集第二轮的三个格基签名之一。Dilithium 可以视作基于 LWE 和 SIS 难题的实用格基签名的集大成者，其设计参考了前期格签名的众多经典之作，并精心挑选了参数以优化性能。能否在 Dilithium 已有的出色性能基础之上百尺竿头更进一步、能否提出更好的设计理念并在性能上有更好的权衡取舍被 Dilithum 的作者在[7]中留为一个重要的开放性问题。在本提案中，我们解决了这个开放性问题。

7.1 创新性

木兰签名方案的创新性体现在：

（1）我们首先对 Dilithium 的设计给出新的解构和解读，并做通用和模块化

的扩展。根据我们在 OKCN-KEX 提案中提出的密钥共识，来解读 Dilithium 的设计。在 OKCN-KEX 提案中，我们给出了最佳噪声密钥共识（OKCN）。通过将 OKCN 机制移植到格基签名环境中，我们提出了 RKCN 机制，它可用于最大限度地压缩格基签名方案中的公钥长度。

（2）基于 RKCN，我们提出了基于通用和模块化结构 RKCN 的格基签名方案木兰（Mulan）。Mulan 的构造是通用且多功能的，并在量子随机预言机模型 QROM 下有可证明安全。特别地，它允许我们可以在更广泛的范围内选择参数。

（3）RKCN 实际上是 OKCN 的确定版：相同的模块可以用于 KEX 和签名，有助于简化格密码的系统复杂性，并具有优异的适配性和兼容性。

（4）通过大量的参数测试，我们确定的参数在安全性、公私钥和签名尺寸、计算效率等所有方面都超越 Dilithium。我们想强调的是，木兰相对于 Dilithium 的安全性和性能优势是建立在和 Dilithium 相同的评测标准上的。

7.2 特色

木兰签名方案的特色体现在：

（1）同一个基础模块 RKCN，既是木兰签名的核心模块，也是 OKCN-KEX 密钥交换的核心模块。据我们所知，这是第一个将格基数字签名和密钥交换的构建基于了同一个基础模块，这非常有利于软硬件优化实现，特别有利于硬件优化实现。

（2）更一般的算法结构，更大的参数选取范围。避免了 Dilithium 对参数（特别是 q）选取的限制。

（3）基于和 Dilithium 相同的安全性评测标准，木兰在安全性、公钥尺寸、私钥尺寸、签名长度、计算效率等所有方面都超越了 Dilithium。

7.3 优缺点说明

木兰签名基于有代数结构的模格上的数学困难问题，其安全性不如定义在标准格上的数学困难难题。但是，目前关于格数学难题的分析还没有发现能够充分利用模格的代数结构来实质提升分析复杂性的方法。

8 适配性说明

我们的木兰签名方案具有优异的兼容性和适配性：

（1）木兰签名和 NIST 的 Dilithium 签名方案具有良好的兼容性，并和我国

算法竞赛的基于模格的密钥交换（OKCN-KEX）、基于模格的密钥封装（AKCN-MLWE）具有优良的兼容性：具有相同的数学基础和兼容性强的计算操作。

（2）木兰在兼容性上有一个独特的优势：RKCN 不仅是既是木兰签名的核心模块，也是 OKCN-KEX 密钥交换的核心模块。据我们所知，这是第一个将格基数字签名和密钥交换的构建基于了同一个基础模块，这非常有利于软硬件优化实现，特别有利于硬件优化实现。

（3）更一般的算法结构，更大的参数选取范围。避免了 Dilithium 对参数（特别是 q）选取的限制。这些性质都有助于木兰签名方案的适配性。

9 第二轮修改说明

相较于第一轮，我们没有对算法结构做修改。主要是通过测试更多的参数，寻找了比第一轮更优的参数。目前找到的参数，在安全性、效率和实现的简洁性上取得了更好的平衡。

利用我们第一轮中给出的多证据机制，签名算法的循环次数还可以进一步降低，但我们也注意到会显著影响硬件实现复杂性。从更有利于硬件实现的角度，也从在有限时间内聚焦精力优化实现的角度，我们在第二轮中重点聚焦木兰基础版。

10 支持文档

[2018-1180] Lattice-Based Signature from Key Consensus. Cryptology ePrint Archive: Report 2018/1180.

11 参考文献

1. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - a new hope. InProceedings of the 25th USENIX Security Symposium. USENIX Association, 327-343.
2. Sanjeev Arora and Rong Ge. New Algorithms for Learning in Presence of Errors. In ICALP 2011, Part I (LNCS), LucaAceto, Monika Henzinger, and Jiri Sgall (Eds.), Vol. 6755. Springer, Heidelberg, 403 - 415.
3. Divesh Aggarwal and Priyanka Mukhopadhyay. Improved algorithms for the

Shortest Vector Problem and the ClosestVector Problem in the infinity norm. Available at http://arxiv.org/abs/1801.02358v2.

4．Shi Bai and Steven D. Galbraith. An Improved Compression Technique for Signatures Based on Learning with Errors.CT-RSA 2014, LNCS Vol. 8366, pages 28-47. Springer, 2014.

5．Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical querymodel. J. ACM 50(4): 506-519 (2003).

6．Mihir Bellare and Gregory Neven Multi-Signatures in the Plain Public-Key Model and a General Forking Lemma.Proceedings of the 13th Association for Computing Machinery (ACM) Conference on Computer and CommunicationsSecurity (CCS), Alexandria, Virginia, 2006, pp. 390-399.

7．Presentation of CRYSTALS (Kyber and Dilithium) at 2018 NIST PQC standardizaiton conference. https://csrc.nist.gov/CSRC/media/Presentations/Crystals-Dilithium

8．Léo Ducas, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS -Dilithium: Digital Signatures from Module Lattices. IACR Cryptology ePrint Archive, 2017/633, 2017.

9．Zhengzhong Jin and Yunlei Zhao. Optimal Key Consensus in Presence of Noise. In CoRR, Vol. abs/1611.06150, 2016.Available at http://arxiv.org/abs/1611.06150

10．Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantumrandom-oracle model. Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference onthe Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, PartIII, pages 552-586.

11．Vadim Lyubashevsky, Leo Ducas, Eike Kiltz, Tancrede Lepoint, Peter Schwabe, Gregor Seiler, and DamienStehle. Crystals-dilithium. Technical report, National Institute of Standards and Technology, 2017. Available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions

12．Vadim Lyubashevsky and Daniele Micciancio. Generalized Compact Knapsacks Are Collision Resistant. In ICALP2006, Part II (LNCS), Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and IngoWegener (Eds.), Vol. 4052. Springer,Heidelberg, 144-155.

13．Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On Ideal Lattices and Learning with Errors over Rings. InEUROCRYPT 2010 (LNCS), Henri Gilbert (Ed.), Vol. 6110. Springer, Heidelberg, 1-23.

14．Adeline Langlois and Damien Stehlé Worst-case to average-case reductions for module lattices. Des. Codes Cryptography75, 3 (2015), 565-599.

15．Vadim Lyubashevsky. Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. ASIACRYPT2009: 598-616.

16．Vadim Lyubashevsky. Lattice Signatures without Trapdoors. In EUROCRYPT 2012 (LNCS), David Pointcheval andThomas Johansson (Eds.), Vol. 7237. Springer, Heidelberg, 738-755.

17．Phong Q Nguyen and Thomas Vidick. Sieve algorithms for the shortest vector problem are practical. Journal ofMathematical Cryptology, 2(2):181õ207, 2008.

18．NIST. Post-Quantum Cryptography Standardization. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization

19．Chris Peikert and Alon Rosen. Efficient Collision-Resistant Hashing fromWorst-Case Assumptions on Cyclic Lattices.In TCC 2006 (LNCS), Shai Halevi and Tal Rabin (Eds.), Vol. 3876. Springer, Heidelberg, 145-166.

20．KCL. Yunlei Zhao, Zhengzhong Jin, Boru Gong, and Guangye Sui. KCL. Technical report, National Institute of Standardsand Technology, 2017. Available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions

21．Vadim Lyubashevsky, Leo Ducas, Eike Kiltz, Tancrede Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehle. CRYSTAL-DILITHIUM. Technical report, National Institute of Standardsand Technology, 2019. Available at

https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions. Last access 20 Oct, 2019.

附录

A 定理 1 证明

Lemma 2 ([JZ16]). *For any $x, y, t, l \in \mathbb{Z}$ where $t \geq 1$ and $l \geq 0$, if $|x - y|_t \leq l$, then there exists $\theta \in \mathbb{Z}$ and $\delta \in [-l, l]$ such that $x = y + \theta t + \delta$.*

Proof. Suppose $|\sigma_1 - \sigma_2|_q \leq d$, by Lemma 2, there exist $\theta \in \mathbb{Z}$ and $\delta \in [-d, d]$ such that $\sigma_2 = \sigma_1 + \theta q + \delta$. From Line 3 to 7 in Algorithm 1, we know that there exists $\theta' \in \mathbb{Z}$ such that $k\sigma_1 = (k_1 + k\theta') \cdot q + v$. Taking these into the formula of k_2 in Rec (Line 12), we have

$$k_2 = \lfloor (k\sigma_2 - v)/q \rceil \bmod k \tag{1}$$
$$= \lfloor k(\sigma_1 + \theta q + \delta)/q - v/q \rceil \bmod k \tag{2}$$
$$= \lfloor k_1 + k\delta/q \rceil \bmod k \tag{3}$$

From the assumed condition $2kd < q$, we get that $|k\delta/q| \leq kd/q < 1/2$, thus $k_2 = k_1$. □

B 数字签名机制及其安全性

定义 2.一个数字签名机制$\prod$由三个概率多项式算法(KeyGen,Sign,Verify)所组成:

- KeyGen 是一个密钥生成算法，输入安全参数1^{λ}，输出(pk,sk)。
- Sign 是一个签名算法，输入待签名的消息$\mu \in \{0,1\}^*$和私钥 sk, 输出签名σ。
- Verify 是一个确定性的验证算法，输入公钥 pk 和消息/签名对(μ, σ),输出$b \in \{0,1\}$,表征输入的消息/签名对(μ, σ)是否有效。

我们说如果对于每一个足够大的 λ, 每一个$(pk, sk) \leftarrow (KeyGen(1^{\lambda})$和每一个$\mu \in \{0,1\}^*$，都有

$$Pr\left[Verify\left(pk, \mu, Sign(sk, \mu)\right) = 1\right] = 1$$

的话，那么这个签名机制$\prod = (KeyGen, Sign, Verify)$是正确的。

定义 3. (自适应消息选择攻击下的(强)存在的不可伪造性(S)EU-CMA)一个签名机制$\prod = (KeyGen, Sign, Verify)$的安全性是通过如下介于一个挑战者和一个敌手 A 之间的游戏来定义的。

- 建立:输入安全参数λ, 挑战者运行$(pk, sk) \leftarrow (KeyGen(1^{\lambda})$。并将公钥 pk 给予敌手 A(同时保持私钥 sk 的机密性)。

- 挑战:设想 A 执行至多s次签名查询。每一个签名问询包括如下步骤: (1) A 基于它所看到的全部信息自适应地选择消息$\mu_i \in \{0,1\}^*, 1 \le i \le s$,并将$\mu_i$发送给签名者；(2) 在收到私钥 sk 和待签名的消息μ_i, 签名者生成并返回相关的签名σ_i给 A。
- 输出: 最终 A 输出一个对(μ, σ), 如果(1) $\text{Verify}(\text{pk}, \mu, \sigma) = 1$ 并且 (2) $(\mu, \sigma) \notin \{(\mu_1, \sigma_1), \dots, (\mu_{q_s}, \sigma_{q_s})\}$，那么 A 获胜。

我们定义$Adv Sig_{\prod, A}^{seu-cma}(1^\lambda)$为 A 在上述游戏中获胜的概率，其中概率来自密钥生成，敌手 A，挑战者(同样也有随机预言机)的随机数。如果对于任意概率多项式时间的敌手 A 都有$Adv Sig_{\prod, A}^{seu-cma}(1^\lambda) = negl(\lambda)$成立的话，我们说签名机制$\prod$是自适应消息选择攻击下是强存在不可伪造的。

一个稍弱一点的定义可以通过相似的方式来定义，其中的安全游戏除了限制$\mu \notin \{\mu_1, \dots, \mu_{q_s}\}$之外与上述的游戏几乎完全一样。如果没有概率多项式时间的敌手能够在这种经过修改的游戏中以不可忽略的概率获胜的话，这样的签名机制称为是自适应消息选择攻击下是(标准)存在不可伪造的。

C 可扩展输出函数

可扩展输出函数的概念来源于[8,11]。一个可扩展输出函数 Sam 是一个输入比特字符串的函数，输出能扩展为任意长度字符串。符号$y \in S := Sam(x)$表示函数 Sam 接受输入 x 并根据S(或是在集合 S 上的均匀分布)分布输出 y。如果对任一个给定的 x 都输出一个相同的 y 的话，在这个意义上，整个过程就被称为是确定性的。简单起见我们假定 Sam 的输出分布是完美的，而实际上它是通过能产生一个在统计上接近完美分布的输出的密码散列函数(建模为随机预言机)实现的。

D 散列函数

我们将要介绍的签名机制可以在随机预言机模型下证明安全性。其中$B_\omega \subsetneq \mathcal{R}_q$表征$\mathcal{R}_q$中元素的集合，集合中恰好$\omega$个系数的值是 1 或者-1，而其它系数的值都是 0。在本论文中一直有$\omega = 60$，因为集合$B_{60} \subseteq \mathcal{R}_q$的大小是$2^{60} \cdot \binom{n}{60} \approx 2^{256}$。

在该项工作中我们以一个散列函数$H:\{0,1\}^* \to B_{60}$来建模为一个随机预言机。实际上，我们可以使用一个由内而外的 Fisher-Yates shuffle 版本从B_{60}中随机选择一个元素。在实际应用中，我们也可以一个特殊的散列函数 H 使得其输出正好包含 30 个位置的 1 和 30 个位置的-1。

E 拒绝采样

引理 3（拒绝采样）令$\mathrm{f}: \mathrm{Z}^{\mathrm{n}} \to R$是一个概率分布。给定一个子集$\mathrm{V} \subseteq \mathrm{Z}^{\mathrm{n}}$，令$\mathrm{h}: \mathrm{V} \to \mathrm{R}$是一个定义在V上的概率分布。令$\mathrm{g_v}: Z^n \to R$是一个以$\mathrm{v} \in \mathrm{V}$来索引的概率分布族，并满足对于几乎所有v，始终存在一个通用上界$\mathrm{M} \in \mathrm{R}$，满足

$$\Pr_{\mathrm{z} \leftarrow \mathrm{f}}[M \cdot g_v(z) < f(z)] = negl(\lambda)$$

之后，对于下列两个算法的输出分布，统计区分的概率是可忽略：

$v \leftarrow h;$
$z \leftarrow g_v;$
Output (z, v) *with probability* $\min\left(1, \frac{f(z)}{M \cdot g_v(z)}\right)$

$v \leftarrow h;$
$z \leftarrow f;$
Output (z, v) *with probability* $\frac{1}{M}$

F 关于参数估计的备注

首先，正如之前提及的，我们通过一般方法[8,11]估计了我们推荐的 SIS 实例的困难性。因为上述文献中的相关估计脚本是不公开的，我们通过[8,11]的方法论来开发了一个简单的估计脚本。

然后，计算实验显示，对于文献[8,11]中建议的 4 个参数设置，我们估计脚本给出的困难性评估总是稍小于文献[8,11]中给出的困难性，详细的对比见下面表 4。我们认为稍小的不同是合理的。可能的原因是，我们用来计算累计分布函数的高斯分布底层子程序与 Dilithium 的不同。因此，反过来没有初等不定积分与数值方法，来计算涉及的积分。

然而，表 4 中的简单对比展示了我们的估计脚本总是比文献[8,11]中的稍保守，特别是，对于本工作中的这些无穷范数下的 SIS 实例，我们在困难性上的估计结果应该都稍小于文献[8,11]中估计脚本的输出结果。这让我们对我们的参数选择更加自信。

表 4 关于在 Dilithium 中 4 组不同建议参数下 SIS 困难性的比较。特别的，对于该四组参数，有$q = 8380417$，$d = 14$，$k = 16$.

	$(h,\ell) = (3,2)$	$(h,\ell) = (4,3)$	$(h,\ell) = (5,4)$	$(h,\ell) = (6,5)$
Dilithium 声称的量子安全性	62	94	125	160
我们的量子安全性	61.1	92.2	124.5	158.2

G 具体安全分析

对于我们的签名来说，需要考虑两种重要的攻击：在给定相应公钥的情况下，致力于恢复密钥的密钥恢复攻击；在 SEU-CMA（见定义 2）安全游戏中尝试伪造一个签名的伪造攻击。据我们所知，实现该两种攻击的最著名的算法都涉及到格基规约和 Core-SVP 问题。

G.1 格基规约，BKZ 算法和 Core-SVP 问题

给定我们推荐的参数，在实际中最著名的找到一个欧几里得格中最短非零向量的算法为 BKZ 算法以及它的变体，该方法要优于组合攻击（例如，BKW 攻击[5]）和代数攻击（例如，Arora-Ge 算法[2]）。

为了解决ℓ_2范数下的 SVP 问题，BKZ 算法解决了一个相关又更一般化的问题，例如，格基规约问题。通常来说，BKZ 通过向区块大小为b的 SVP 预言机询问多项式次来解决格基规约问题。格基规约问题的困难性由下面两个方面推出：为了获取一个“好”的基，BKZ 算法需要配备一个大的区块大小为b的 SVP 预言机；实现该 SVP 预言机的开销为区块大小b的指数级别（事实上，最著名的量子 SVP 解决器[1]运行时间$\approx 2^{c_Q \cdot b}$，其中$c_Q = \log_2 \sqrt{13}/9 \approx 0.265$）。

由于分析 SVP 调用数量的上界相对比较困难，因此 Core-SVP 模型被提出[1]用来确认根据单次调用区块大小为b的 SVP 预言机的开销情况下的 BKZ 算法的开销。这一悲观的估计表明了与[8,11]类似，我们的安全分析非常保守。

G.2 伪造攻击和 Module-SIS 问题

如 5.3.2 小节所述，伪造攻击可以归约到解决 SelfTargetMSIS 问题。而要解决 SelfTargetMSIS 问题，我们要么攻破$H(\cdot)$的安全性，要么解决输入为$(\mathbf{A}, \mathrm{t}', \beta)$时的 MSIS 问题。攻破$H(\cdot)$的困难性在于在我们的安全证明中$H(\cdot)$被模型化为一个随机预言机，并且$H(\cdot)$的值域$\mathrm{B}_{60}$的大小为$2^{256}$左右。因此，我们将重点放在分析 MSIS 的困难性上。

MSIS 问题在本质上可以看作是 SIS 问题在l_∞范数下的变体，和标准的l_2范数下的 SIS 问题中的形成鲜明对比。简而言之，l_∞范数下的一个“短”向量同样也是l_2范数下的一个“短”解，但是反之则不成立。（实际上，对于使用我们推荐的参数下的我们的签名方案而言，我们的 SelfTargetMSIS 问题的解的欧几里得长度大于q，鉴于向量$(q, 0, \ldots, 0)^t$的欧几里得长度为q，但是它的无穷范数并不满足条件。）这说明了我们面对的问题比l_2范数下的 SIS 问题更难。

因为 BKZ 算法只能在欧几里得格上运行，我们无法直接的将 MSIS 实例转化为 Core-SVP 实例。尽管如此，我们可以使用文献[8,11]中提出的通用的方法，继续使用 BKZ 算法来确定无穷范数下的解。

具体来说，我们首先可以通过选择一个w列的子集来缩小分析的范围，并将目标向量的其他维数全部赋值为零。然后我们通过调用 BKZ 算法来寻找一个合适的“短”向量，但是不再是输出格基中的第一个，例如，欧几里得范数下最短的一个。相反，我们要找到一个格向量，该向量在每个方向上的投影都不是非常大，这与我们希望找小的l_∞范数下的非零向量的目的是一致的。

当 BKZ 的区块大小确定时，可以进行启发式假设以估计存在理想的格向量的概率：我们假设那些不受 BKZ 算法影响的维度，相关坐标遵循模 q 意义下的均匀分布，而对于那些受 BKZ 算法影响的维度，相关坐标遵循具有适当标准差的高斯分布。成本估算是该概率的倒数乘以 b-维 SVP-求解器的运行时间[8,11]。

G.3 无穷范数下的 SVP 问题

回想一下，BKZ 算法在欧几里得范数下的 b-维 SVP 求解器的帮助下解决了格基归约问题，这使得 BKZ 可以解决欧几里得范数下的 SIS 问题。在 Dilithium[8,11]中，该方法被推广用于估算无穷范数中 SIS 问题的困难度。

最近，在[3]中提出了一种新的筛分程序，他的运行时间与在底层格中随机选

择的样品数量 N 存在线性关系; 因此，结合其他结果，获得了用于求解无穷范数下的 SVP 问题的改进算法。该新算法运行时间至少为$(4/3)^n$，其中 n 表示输入的格维度。此外，在[3]中暗示，该下界在无穷范数下的 SVP 问题的空间复杂度而言几乎是最优的，因为为了进一步改进该下界，需要一些理论上的突破。