基于格的密码学技术专题讲座(一)

第 2 讲　基于格的密码函数构造方法及其应用

韩敬利，杨　明，王兆丽
（解放军理工大学指挥信息系统学院，江苏 南京 210007）

摘　要：文章主要介绍了基于随机格、理想格的哈希函数和原像采样陷门函数，对构造思想、构造过程等方面进行了详细的描述，并对其在最坏情况下的安全性进行了讨论，介绍了在基于格的密码函数基础上构建签名方案和基于身份的加密方案的方法。

关键词：格；理想格；哈希函数；原像采样陷门函数
中图分类号：TP309.2　　**文献标识码**：A　　**文章编号**：CN32-1289(2014)01-0075-06

Construction and Application of Lattice-based Cryptographic Functions

HAN Jing-li, *YANG Ming*, *WANG Zhao-li*
(College of Command Information System, PLAUST, Nanjing 210007, China)

Abstract: The hash functions and the preimage sampleable trapdoor functions based on q-ary lattice or ideal lattice were introduced. In the lecture, the primary strategy and the algorithm of the construction were specifically explained. The security of the functions based on the worst-case hardness of lattice problems was also discussed, and the application of lattice-based cryptographic functions was introduced in the end.

Key words: lattice; ideal-lattice; hash functions; preimage sampleable trapdoor functions

哈希函数是一种将任意长度的消息压缩到某一固定长度的消息摘要的函数，可用于数字签名、消息的完整性测试等。当前广泛应用的哈希函数主要是基于 ad-hoc 的设计原则，类似于分组密码的方式进行构建，如此设计的哈希函数已经存在多种攻击手段。利用数学中已经得到证明的难解问题或者是公认的难题来构建可证明安全性的哈希函数是密码学研究的一种基本手段，比如利用大整数分解、RSA 等难题来构建抗碰撞的哈希函数。但是这样的构造方法还是存在一些缺点：效率比分组密码方式低，无法抵御量子计算的攻击，往往针对问题的最坏情况而言是安全的。由于这些缺点的存在，研究新型的构建方案从未停止过。随着 Ajtai[1] 里程碑式的论文，利用格上难题来构建抗碰撞哈希函数的做法引起了许多学者的兴趣。格具有运算效率高、抗量子攻击等优点，而且基于格构建的密码方案的平均情况安全性与最坏情况安全性是等价的。

当前实际应用的公钥密码系统中，两个比较重要的概念是陷门函数和选择密文安全 CCA(Chosen Ciphertext Attack Security)。陷门函数是一类有陷门的特殊单向函数，最早实现应用在 RSA 函数中。陷门函数在一个方向上易于计算而反方向却难于计算，但是如果知道秘密陷门，则也能很容易的在另一个方向计算这个函数。在密码学中最常用的陷门函数有两类，一是公钥密码系统中使用的单向陷门函数，二是消息摘要中使用的带陷门的哈希函数。Gentry，Peikert 和 Vaikuntanathan[2] 提出了一种基于最坏情况下格上难题来构建陷门哈希函数的方法，并将构造的陷门哈希函数称为原像采样陷门函数 PSFs(Preimage Sample-

收稿日期：2013-07-20；修回日期：2013-10-03
作者简介：韩敬利(1983－)，女，硕士，讲师.

able Trapdoor Functions)。因为格上计算的高效性,且 PSFs 是基于最坏情况下格上难题来构造的,所以可以用 PSFs 代替很多已知的签名方案中的陷门函数,如此可以得到更有效、更安全的签名方案。

1 哈希函数

1.1 哈希函数模型

哈希函数是一个带密钥的函数族$\{h_k:\mathbf{D}_n\rightarrow\mathbf{R}_n \mid k\in\mathbf{K}_n\}$,将定义域 $\mathbf{D}_n$ 映射一个更小的集合 $\mathbf{R}_n$ 上。抗碰撞的哈希函数是指对于一个随机选取的密钥 k 来说,攻击者没有有效的算法可以找到碰撞,虽然碰撞确实存在。一般哈希方案 Hash 主要包括两个算法:

(1) Setup(1^n)。给定安全参数 1^n,输出一个密钥 k。

(2) Eval(k, m)。给定密钥 k 和消息 $m\in\mathbf{D}_n$,输出一个消息摘要 $d\in\mathbf{R}_n$,记作 $d=\mathrm{Eval}(k,m)=h_k(m)$。

1.2 基于随机格的抗碰撞的哈希函数

随着 MD5 和 SHA-1 的相继找到碰撞,设计抗碰撞的安全哈希函数再次成为密码学研究的重点。本节将介绍由 Ajtai 最早提出的基于格的哈希函数的构造算法。

Ajtai 的方案是基于 q-元格(也称为随机格)来构造的,q-元格在实际中有着重要的应用,其中应用最多的是两个由随机矩阵 $\boldsymbol{A}\in\mathbf{Z}_q^{n\times m}$ 定义的 q-元格:$\boldsymbol{\Lambda}_q(\boldsymbol{A})=\{\boldsymbol{y}\in\mathbf{Z}^m \mid \boldsymbol{y}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{e} \bmod q,\ \boldsymbol{e}\in\mathbf{Z}^n\}$和 $\boldsymbol{\Lambda}_q^{\perp}(\boldsymbol{A})=\{\boldsymbol{y}\in\mathbf{Z}^m \mid \boldsymbol{A}\boldsymbol{y}=0 \bmod q\}$,Ajtai 提出的算法是利用 $\boldsymbol{\Lambda}_q^{\perp}(\boldsymbol{A})$来构造抗碰撞哈希函数,所构建的哈希函数的安全性讨论可规约到格上难题的求解上。

算法 1:Ajtai 的基于随机格的哈希函数

(1) 参数。整数 n、m、q、$r\geqslant 1$,其中 $n\log q<m\leqslant\frac{q}{2n^4}$。

(2) 密钥。一个随机选取的矩阵 $\boldsymbol{A}\in\mathbf{Z}_q^{n\times m}$。

(3) 哈希函数。$h_{\boldsymbol{A}}:\{0,\cdots,r-1\}^m\rightarrow\mathbf{Z}_q^n$,$h_{\boldsymbol{A}}(\boldsymbol{y})=\boldsymbol{A}\boldsymbol{y} \bmod q$。

如果对于 $\boldsymbol{y}\neq\boldsymbol{y}'$有一个碰撞 $h_{\boldsymbol{A}}(\boldsymbol{y})=h_{\boldsymbol{A}}(\boldsymbol{y}')$存在,那么立即会产生一个非零的短向量 $\boldsymbol{y}-\boldsymbol{y}'\in\boldsymbol{\Lambda}_q^{\perp}(\boldsymbol{A})$,Goldreich[3] 等证明了此问题可在多项式时间内规约到最坏情况下格上近似最短向量问题(app-SVP)[4,5],所以此算法构建的哈希函数具有抗碰撞的安全属性。

因为只涉及到模 q 加法和乘法,所以 Ajtai 的算法相较于其他密码函数来说运算效率更高。如果参数选择得当,比如令 q 为 2 的幂、r 为 2,还可以消除掉算法当中的乘法运算。但是算法的主要缺点是密钥过长,密钥 $\boldsymbol{A}\in\mathbf{Z}_q^{n\times m}$ 长度至少是 n 的平方级,所以算法还不是实际有效的,还有待进一步的优化。

1.3 基于理想格的抗碰撞的有效哈希函数

为改善格在密码学应用中的效率,需要考虑一些具有特殊结构的格,循环格就是这样的格,它在公钥密码方案 NTRU(Number Theory Research Unit)中有很好的应用。循环格是对格上点的分量进行循环移位后形成的点依然是这个格上的点,即循环格 $\mathbf{L}\subseteq\mathbf{Z}^n$,对于 $\forall(u_1,u_2,\cdots,u_n)\in\mathbf{L}$,必然有$(u_n,u_1,\cdots,u_{n-1})\in\mathbf{L}$。循环格与环的理想有密切的联系,将格矢量 $\boldsymbol{u}\in\mathbf{Z}^n$ 进行多项式的系数嵌入,即 $\boldsymbol{u}=(u_1,u_2,\cdots,u_n)$对应多项式 $u(x)=u_1+u_2x+\cdots+u_nx^{n-1}$,这样的循环格对应多项式商环 $\mathbf{Z}[x]/\langle x^n-1\rangle$,其中$\langle x^n-1\rangle$为多项式环 $\mathbf{Z}[x]$的理想。

利用循环格的特殊结构,不仅可以进行更紧凑的表示,减少密钥量,还可利用快速算法如 FFT 实现高效的密码运算。Goldreich 等将 Ajtai 算法中使用到的一般格替换为具有特殊结构的理想格(ideal lattice),降低密钥长度来提高哈希函数的有效性[6,7]。

Goldreich 优化的具体思路为将算法 1 中的随机矩阵 $\boldsymbol{A}\in\mathbf{Z}_q^{n\times m}$,替换为一个分块矩阵 $\boldsymbol{A}=[\boldsymbol{A}^{(1)} \mid \cdots \mid \boldsymbol{A}^{(m/n)}]$,其中每一个分块 $\boldsymbol{A}^i\in\mathbf{Z}_q^{n\times n}$ 均是对应于一个 n 维向量的循环矩阵。

理想格是对应环 $\mathbf{R}_f=\mathbf{Z}[x]/\langle f(x)\rangle$中理想的格,其中 $f(x)=f_0+f_1x+\cdots+f_{n-1}x^{n-1}+x^n\in\mathbf{Z}[x]$为首

一的不可约整系数多项式。多项式环 $\mathbf{R}_f$ 中的元素 $a=a_0+a_1x+\cdots+a_{n-1}x^{n-1}$ 均为次数不超过 $n-1$ 的整系数多项式，可直接用 n 维列向量 $\boldsymbol{a}=(a_0,\cdots,a_{n-1})$ 来表示。由 $\boldsymbol{a}$ 可构造循环矩阵：

$$\boldsymbol{A}^{(i)}=\begin{bmatrix} a_0^{(i)} & a_{n-1}^{(i)} & \cdots & a_1^{(i)} \\ a_1^{(i)} & a_0^{(i)} & \cdots & a_2^{(i)} \\ & & \cdots & \\ a_{n-1}^{(i)} & a_{(n-2)}^{(i)} & \cdots & a_0^{(i)} \end{bmatrix}=\boldsymbol{F}\cdot\boldsymbol{a}^{(i)}=[\boldsymbol{a}^{(i)},\boldsymbol{F}\boldsymbol{a}^{(i)},\cdots,\boldsymbol{F}^{n-1}\boldsymbol{a}^{(i)}]$$

其中，$\boldsymbol{F}=\left[\begin{array}{c|c} \begin{matrix}0^{\mathrm{T}}\\ \cdots\end{matrix} & -\boldsymbol{f} \\ \boldsymbol{I} & \\ \cdots & \end{array}\right]$，$\boldsymbol{f}=(f_0,\cdots,f_{n-1},0)$。

算法 2：基于理想格的哈希函数

(1) **参数**。整数 q、n、m、r 且 $n\mid m$，$n\log q<m\leqslant\frac{q}{2n^4}$，$f(x)=x^n+1$，$n$ 是 2 的幂。

(2) **密钥**。在 $\mathbf{Z}_q^n$ 中随机独立选取的 m/n 个向量 $\boldsymbol{a}_1,\cdots,\boldsymbol{a}_{m/n}$。

(3) **哈希函数**。$h_{\boldsymbol{A}}:\{0,\cdots,r-1\}^m\to\mathbf{Z}_q^n$，$h_{\boldsymbol{A}}(\boldsymbol{y})=[\boldsymbol{F}\cdot\boldsymbol{a}_1\mid\cdots\mid\boldsymbol{F}\cdot\boldsymbol{a}_{m/n}]\boldsymbol{y}\bmod q$

上述哈希函数的碰撞会产生格 $\boldsymbol{\Lambda}_q^{\perp}([\boldsymbol{F}\cdot\boldsymbol{a}_1\mid\cdots\mid\boldsymbol{F}\cdot\boldsymbol{a}_{m/n}])$ 中的一个短向量，当 f 为不可约多项式且对任意的单位向量 $\boldsymbol{u}$、$\boldsymbol{v}$，$[\boldsymbol{F}\cdot\boldsymbol{u}]\boldsymbol{v}$ 是一个小范数的向量时，找到此哈希函数的碰撞与解决最坏情况下理想格上的 app-SVP 问题是同等困难的，所以此算法构建的哈希函数具有抗碰撞的安全属性。由于理想格的特殊结构，算法将密钥空间从 Ajtai 算法的 nm 个 $\mathbf{Z}_q$ 中的元素降低到只需要 m 个元素，同时在算法中可利用离散傅里叶变换 DFT(Discrete Fourier Transform) 使函数运算的时间复杂度从 $O(mn)$ 降为 $O(m)$。

1.4　SWIFFT 哈希函数

V. Lyubashevsky 和 D. Micciancio[8] 等提出的 SWIFFT 哈希函数是对算法 2 的优化，利用快速傅里叶变换 FFT(Fast Fourier Transform) 将运算的时间复杂度降低为 $O(n\log n)$，进一步提高了算法的有效性。令 $f(x)=x^n+1$，$n=2^k$，素数 q 满足 $2n\mid q-1$，在这种参数选择下，乘法群 $\mathbf{Z}_q^*$ 中有一个阶为 $2n$ 的元 w。令 $\boldsymbol{W}=[w^{(2i-1)(j-1)}]_{i=1,j=1}^{n,n}$，则 $\boldsymbol{W}$ 是由 $w,w^3,w^5,\cdots,w^{2n-1}$ 构成的范德蒙矩阵，且对任意的向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$，有 $\boldsymbol{W}([\boldsymbol{F}\cdot\boldsymbol{a}]\boldsymbol{b})=(\boldsymbol{W}\boldsymbol{a})\odot(\boldsymbol{W}\boldsymbol{b})\bmod q$，SWIFFT 哈希函数借助此范德蒙矩阵通过 FFT 算法加速运算。

算法 3：SWIFFT 哈希函数

(1) **参数**。整数 n、m、q、r，且 n 是 2 的幂，q 是素数，$2n\mid q-1$，$n\mid m$。

(2) **密钥**。在 $\mathbf{Z}_q^n$ 中随机独立选取 m/n 个向量 $\boldsymbol{a}^{(1)},\cdots,\boldsymbol{a}^{(m/n)}$，$\boldsymbol{W}$ 为范德蒙德矩阵，密钥为 m/n 个向量 $\bar{\boldsymbol{a}}^{(1)},\cdots,\bar{\boldsymbol{a}}^{(m/n)}$，其中 $\bar{\boldsymbol{a}}^{(i)}=\boldsymbol{W}\boldsymbol{a}^{(i)}$。

(3) **输入**。m/n 个向量 $\boldsymbol{y}^{(1)},\cdots,\boldsymbol{y}^{(m/n)}\in\{0,\cdots,r-1\}^n$。

(4) **输出**。$\boldsymbol{W}\cdot f_{\boldsymbol{A}}(\boldsymbol{y})=\sum_{i=1}^{m/n}\bar{\boldsymbol{a}}^{(i)}\otimes(\boldsymbol{W}\boldsymbol{y}^{(i)})\in\mathbf{Z}_q^n$，其中 $\mathbf{A}=[\boldsymbol{F}\cdot\boldsymbol{a}^{(1)},\cdots,\boldsymbol{F}\cdot\boldsymbol{a}^{(m/n)}]$，$\otimes$ 为分块向量乘法($\boldsymbol{a}\otimes\boldsymbol{b}=(a_1,\cdots,a_n)\otimes(b_1,\cdots,b_n)=(a_1b_1,\cdots,a_nb_n)$)。

在算法中可以通过 FFT 有效计算矩阵-向量乘法 $\boldsymbol{W}\boldsymbol{y}^{(i)}$，降低算法的计算复杂度。以 $n=4$ 为例，$\boldsymbol{W}_{4,w}\cdot\boldsymbol{a}$ 可以通过 $\boldsymbol{W}_{2,w^2}\cdot\boldsymbol{a}'$ 的计算得出，$\boldsymbol{W}_{4,w}\cdot\boldsymbol{a}=\begin{pmatrix}\boldsymbol{W}_{2,w^2}\cdot\boldsymbol{a}_{\mathrm{e}}+(w,w^3)\otimes(\boldsymbol{W}_{2,w^2}\cdot\boldsymbol{a}_{\mathrm{o}})\\ \boldsymbol{W}_{2,w^2}\cdot\boldsymbol{a}_{\mathrm{e}}-(w,w^3)\otimes(\boldsymbol{W}_{2,w^2}\cdot\boldsymbol{a}_{\mathrm{o}})\end{pmatrix}$，其中 $\boldsymbol{a}_{\mathrm{e}}=(a_0,a_2)$，$\boldsymbol{a}_{\mathrm{o}}=(a_1,a_3)$，$\boldsymbol{W}_{2,w^2}=\begin{bmatrix}w^{2\cdot0}&w^{2\cdot1}\\ w^{2\cdot0}&w^{2\cdot3}\end{bmatrix}$。

2　陷门函数

自从 Ajtai 将格上难题的平均情况复杂度和最坏情况复杂度关联起来之后，基于最坏情况的格上难题

来构建密码系统就成为研究的热点，并且在随后取得了不少的研究成果。直到最近，大多数的研究还都是限定在了单向、抗碰撞的哈希函数和公钥密码系统上。Gentry，Peikert 和 Vaikuntanathan 根据 GGH 的启发，提出了一种基于最坏情况下格上难题来构建带陷门的哈希函数的方法，说明了如果将格上的短基作为陷门则容易得到哈希函数的碰撞，Gentry 将构建的陷门函数称为原像采样陷门函数 PSFs。

2.1 基于随机格的陷门生成算法

本节介绍由 Alwen 和 Peikert 提出的基于随机格的陷门生成算法。对于任意常数 $\delta > 0$、$r \geqslant 2$，安全参数 n，参数 m_1、m_2、q，且满足 $m = m_1 + m_2$，q 是一个奇素数，输入随机矩阵 $\boldsymbol{A}_1 \in \mathbf{Z}_q^{n \times m_1}$，算法可在多项式时间内输出($\boldsymbol{A} = [\boldsymbol{A}_1 | \boldsymbol{A}_2]$，$\boldsymbol{S}$)$\in \mathbf{Z}_q^{n \times m} \times \mathbf{Z}^{m \times m}$，其中 $\boldsymbol{S}$ 是 $\Lambda_q^{\perp}(\boldsymbol{A})$的一组短基，且对任意的 $\omega(\sqrt{\log n})$有压倒性的优势满足 $\| \boldsymbol{S} \| \leqslant (m_1 + n\log_r q)\omega(\sqrt{\log n})$。

对于随机矩阵 $\boldsymbol{A}_1 \in \mathbf{Z}_q^{n \times m_1}$，算法首先计算出随机格 $\Lambda_q^{\perp}(\boldsymbol{A}_1)$一组基的 Hermite 范式 $\boldsymbol{H} \in \mathbf{Z}_q^{m_1 \times m_1}$。随后构造矩阵 $\boldsymbol{U} = \boldsymbol{G} + \boldsymbol{J}$，令 $\boldsymbol{A}_2 \equiv -\boldsymbol{A}_1\boldsymbol{U} (\bmod q)$，$\boldsymbol{F} = [\boldsymbol{H} | \boldsymbol{U}; 0 | \boldsymbol{I}_{m_2}]$，则 $\boldsymbol{AF} \equiv 0 \bmod q$。最后，构造一个幺模矩阵 $\boldsymbol{Q} = [-\boldsymbol{I}_{m_1} | 0; \boldsymbol{P} | \boldsymbol{B}]$，定义 $\boldsymbol{S} = \boldsymbol{FQ}$，则 $\boldsymbol{AS} = 0$，至此得到格 $\Lambda_q^{\perp}(\boldsymbol{A})$的一组短基 $\boldsymbol{S}$。

基于格的陷门生成算法主要是构造出相关矩阵 $\boldsymbol{B} \in \mathbf{Z}^{m_2 \times m_2}$、$\boldsymbol{J} \in \mathbf{Z}^{m_1 \times m_2}$、$\mathbf{G} \in \mathbf{Z}^{m_1 \times m_2}$、$\boldsymbol{P} \in \mathbf{Z}^{m_2 \times m_1}$。令 $\boldsymbol{H}' = \boldsymbol{H} - \boldsymbol{I}_{m_1}$，$c_i$ 和 c_i'分别为 $\boldsymbol{H}$ 和 $\boldsymbol{H}'$的第 i 个对角元素，$l_i = |\log_r c_i|$。

(1) $\boldsymbol{J}$ 的定义：$\boldsymbol{J} = [\boldsymbol{J}'; 0] \in \mathbf{Z}^{m_1 \times m_1}$，$\boldsymbol{J}'$在$\{-1, 0, 1\}^{d \times m_2}$中随机选取。

(2) $\boldsymbol{B}$ 的定义：取 $\boldsymbol{B} = \mathrm{diag}(\boldsymbol{T}_{l_1}, \cdots, \boldsymbol{T}_{l_{m_1}}, \boldsymbol{I}_{m_2 - s}) \in \mathbf{Z}^{m_2 \times m_2}$(其中 $\boldsymbol{T}_k = \begin{bmatrix} 1 & -r & & & \\ & 1 & -r & & \\ & & & 1 & \cdots \\ & & & \cdots & -r \\ & & & & 1 \end{bmatrix} \in \mathbf{Z}^{k \times k}$)。

(3) $\boldsymbol{G}$ 的定义：$\boldsymbol{G}$ 分解为 $m_1 + 2$ 个分块矩阵：$\boldsymbol{G} = [\boldsymbol{G}^{(1)} | \cdots | \boldsymbol{G}^{(m_1)} | \boldsymbol{M} | 0]$，其中 $\boldsymbol{G}^{(k)} = \{g_{i,j}^{(k)}\}_{i \in [m_1],\, j \in [l_k]} \in \mathbf{Z}^{m_1 \times l_k}$，$g_{i,j}^{(k)} = \begin{cases} r^j & i = k \\ 0 & \text{其它} \end{cases}$。

(4) $\boldsymbol{P}$ 的定义：令 $\boldsymbol{P} = [\boldsymbol{P}^{(1)}; \cdots; \boldsymbol{P}^{(m_1)}; 0] \in \mathbf{Z}^{m_2 \times m_1}$，其中 $\boldsymbol{P}^{(i)} = [\boldsymbol{p}_1^{(i)} | \cdots | \boldsymbol{p}_{m_1}^{(i)}]$，$\boldsymbol{p}_j^{(i)} = (p_{1,j}^{(i)}, \cdots, p_{l_i,j}^{(i)}) \in \mathbf{Z}^{l_i}$，$p_{k,j}^{(i)} \in \{0, \cdots, r-1\}$。

2.2 基于随机格的原像采样陷门函数

Gentry、Alwen 结合 Alwen 和 Peikert 的陷门生成方法、Ajtai 的哈希函数构造方法和采样算法，提出了一种基于随机格来构造抗碰撞的原像采样陷门函数方案，此方案称为 LPSF。下面是对方案 LPSF 的描述。

(1) TrapGen(1^n)。根据 Alwen 和 Peikert 提出的基于随机格的陷门生成算法，输出($\boldsymbol{A}$，$\boldsymbol{T}$)。其中 $\boldsymbol{A} \in \mathbf{Z}_q^{n \times m}$，$\boldsymbol{T} \subset \Lambda_q^{\perp}(\boldsymbol{A})$是一组好基且满足 $\| \tilde{\boldsymbol{T}} \| \leqslant O(\sqrt{n \log q})$($\tilde{\boldsymbol{T}}$ 为格基 $\boldsymbol{T}$ 的 Gram-Schmidt 正交化)。

(2) Eval($\boldsymbol{A}$，$\boldsymbol{x}$)。根据 Ajtai 基于随机格的哈希函数构造算法，输出 $\boldsymbol{Ax} \bmod q$。

(3) SamplePre($\boldsymbol{A}$，$\boldsymbol{T}$，s，$\boldsymbol{u}$)。从原像集合中进行采样。首先利用代数方法生成 $\boldsymbol{t} \in \mathbf{Z}^m$ 满足 $\boldsymbol{At} = \boldsymbol{u} \bmod q$，根据 Gentry 提出的格的离散高斯采样算法，采样 $\boldsymbol{v} \leftarrow \mathbf{D}_{\Lambda_q^{\perp}(\boldsymbol{A}), s, -\boldsymbol{t}}$($\mathbf{D}_{\Lambda_q^{\perp}(\boldsymbol{A}), s, -\boldsymbol{t}}$为 $\Lambda_q^{\perp}(\boldsymbol{A})$上的以 $-\boldsymbol{t}$ 为中心的离散高斯分布)，输出 $\boldsymbol{e} = \boldsymbol{t} + \boldsymbol{v}$。

原像采样陷门函数可看做是一个解码器，根据像 $\boldsymbol{u}$ 采样出原像 $\boldsymbol{e}$，满足 $\boldsymbol{Ae} = \boldsymbol{u} \bmod q$。如果 $\delta > 0$ 且 $m \geqslant (5 + 3\delta)\log q$ 时，上述 LPSF 方案构建的原像采样陷门函数具有抗碰撞的安全属性，其安全性可规约到 $\mathrm{SIS}_{q, m, 2s\sqrt{m}}$问题[9]。

2.3 基于理想格的陷门生成算法

为了得到基于理想格的抗碰撞的原像采样函数，本节先介绍由 Stehle，Steinfeld 等提出的对 Alwen 的基于随机格的陷门生成算法的改进。

参数为 q，整数 n，d，m_1，m_2，σ，$f=x^{2^k}+1$，随机多项式向量 $\bar{a}_1\in\mathbf{R}_{f,q}^{m_1}$，其中 $\mathbf{R}_{f,q}=\mathbf{Z}_q[x]/<f>$，$m=m_1+m_2$，在合适的参数选择下，输入随机多项式向量 $\bar{a}_1\in\mathbf{R}_{f,q}^{m_1}$，Stehle 算法可在多项式时间内输出 $(\bar{a},\boldsymbol{S})$，$\boldsymbol{S}$ 为格 $\mathbf{M}^{\perp}(\bar{a})=\{\bar{e}\in\mathbf{R}_f^m\mid\overline{ae}\equiv 0(\bmod q)\}$ 的一组短基。

与 Alwen 算法的构造思想类似，对于随机多项式向量 $\bar{a}_1=(a_1,\cdots,a_{m_1})$，构建多项式向量 $\bar{a}_2$，令 $\bar{a}=[\bar{a}_1\mid\bar{a}_2]$，求取 $\mathbf{M}^{\perp}(\bar{a})$ 的一组短基 $\boldsymbol{S}$。因为 f 在 $\mathbf{Z}_q$ 上是不可约多项式，所以没法定义 $\mathbf{M}^{\perp}(\bar{a}_1)$ 基的 Hermite 范式，区别于 Alwen 的算法，用 $\mathbf{M}^{\perp}(\bar{a}_1)$ 中类似于 Hermite 范式的基 $\boldsymbol{H}$ 代替，算法构造过程中涉及到的矩阵 $\boldsymbol{G}$、$\boldsymbol{J}$、$\boldsymbol{P}$、$\boldsymbol{B}$ 的构造思想均类似于 Alwen 的算法，在此不再累述。

2.4　基于理想格的带陷门原像采样函数

Damien 和 Steinfeld 等通过替换 LPSF 方案中的陷门生成算法得到基于理想格、抗碰撞的原像采样陷门函数，即 ILPSF 方案[10]，提高了 LPSF 的有效性。下面是对 ILPSF 方案的描述。

(1) TrapGen(1^n)。根据 Stehle，Steinfeld 等提出的短基生成算法[11]，输出($\boldsymbol{a}$，$\boldsymbol{T}$)，$\boldsymbol{a}\in\mathbf{R}_{f,q}^m$ 定义了一个哈希函数 $h_{\boldsymbol{a}}(\boldsymbol{x})=(\boldsymbol{F}\cdot\boldsymbol{a})\cdot\boldsymbol{x}\bmod q$，定义域为 $\mathbf{D}_n=\{\boldsymbol{x}\in\mathbf{Z}^m:\|\boldsymbol{x}\|_\infty\leqslant s\log m\}$，值域 $\mathbf{R}_n=\mathbf{Z}_q^n$。$\boldsymbol{T}'=\boldsymbol{F}\cdot\boldsymbol{T}\subset\boldsymbol{\Lambda}_q^{\perp}(\boldsymbol{F}\cdot\boldsymbol{a})$ 是一组好基且 $\|\bar{\boldsymbol{T}}'\|\leqslant O(\sqrt{n\log q})$。

(2) Eval($\boldsymbol{a}$，$\boldsymbol{x}$)。Eval($\boldsymbol{a}$，$\boldsymbol{x}$)$=(\boldsymbol{F}\cdot\boldsymbol{a})\cdot\boldsymbol{x}\bmod q$。

(3) SamplePre($\boldsymbol{a}$，$\boldsymbol{T}$，s，$\boldsymbol{u}$)。首先利用代数方法生成 $\boldsymbol{t}\in\mathbf{Z}^m$ 使得 $(\boldsymbol{F}\cdot\boldsymbol{a})t=\boldsymbol{u}\bmod q$，根据格的离散高斯采样算法采样 $\boldsymbol{v}\leftarrow\mathbf{D}_{\boldsymbol{\Lambda}_q^{\perp}(\boldsymbol{F}\cdot\boldsymbol{a}),s,-\boldsymbol{t}}$，输出 $\boldsymbol{e}=\boldsymbol{t}+\boldsymbol{v}$。

当 $f(x)=x^n+1$，$n=2^k\geqslant 32$，m 是整数且 $m\geqslant 41\log q$，q 是素数且 $q\equiv 3\bmod 4$ 时，$f-\mathrm{SIS}_{q,m,2s\sqrt{mn}}$ 问题是难解问题，所以上述 ILPSF 方案具有抗碰撞的安全属性。

3　基于格的密码函数的应用

3.1　基于格的签名方案

签名方案的哈希-签名范例最早是由 Diffie 和 Hellman 在 1976 年提出的，Bellare 等称这种基本方案为 FDH(Full-Domain Hash)。FDH 中密钥为一个陷门函数 f 和签名密钥 f^{-1}，对于消息 m，首先求得哈希值 $y=h(m)$，然后输出签名 $\sigma=f^{-1}(y)$，只要简单的确认 $f(\sigma)=h(m)$ 即可验证签名。

基于格的签名方案最早是由 Gentry 等提出的，称为 Gentry-Peikert-Vaikuntanathan 签名方案(GPV 签名方案)。由于格上计算的有效性，基于格上的哈希函数来构建签名方案成为格上密码函数的一个重要应用。GPV 方案是建立在 2.2 节介绍的 LPSF 上，GPV 签名方案可抵御选择明文攻击，其安全性证明可规约到 SIS 难解问题。下面给出 GPV 签名方案的详细描述。

(1) SigKeyGen(1^n)。利用 LPSF 的 TrapGen(1^n)算法，输出($\boldsymbol{A}$，$\boldsymbol{T}$)，在签名方案中，$\boldsymbol{T}$ 为签名密钥、$\boldsymbol{A}$ 则为验证密钥。

(2) Sign($\boldsymbol{T}$，$\boldsymbol{m}$)。利用 LPSF 的 SamplePre($\boldsymbol{A}$，$\boldsymbol{T}$，s，$h(\boldsymbol{m})$)算法，输出 $\boldsymbol{\sigma}_m$。其中，h 为 oracle 随机函数 $h:\{0,1\}^*\rightarrow\mathbf{Z}_q^n$。

(3) Verify($\boldsymbol{A}$，$\boldsymbol{m}$，$\boldsymbol{\sigma}$)。如果 $\boldsymbol{A\sigma}=h(\boldsymbol{m})\bmod q$，验证通过。否则，验证不通过。

将上述方案中的 LPSF 替换为 2.4 节介绍的 ILPSF，即可构建基于理想格的签名方案，在此不再累述。

3.2　基于格的 IBE 方案

基于身份的加密方案 IBE(Identity-Based Encryption)是一种公钥加密方法，与传统的公钥加密方法相比较，IBE 方案为用户和管理者大大减少了加密过程的复杂性。IBE 方案可以由一个签名方案(KeyGen、Sign、Verify)和一个加密方案(Encode、Decode)结合得到。本节介绍基于格的 IBE 方案，最早也是由 Gentry 等人在 GPV 签名方案的基础之上构建的，且此方案已证明是可抵御选择明文攻击的。下面给出 Gentry 提

出的 IBE 方案的详细描述。

(1) Setup(1^n)。根据安全参数 n,生成公钥 para 和主私钥 msk。利用 LPSF 方案中的 TrapGen(1^n)算法,得到($\boldsymbol{A}$, $\boldsymbol{T}$),$\boldsymbol{A}$ 为公钥,$\boldsymbol{T}$ 为主私钥。

(2) Extract($\boldsymbol{A}$, $\boldsymbol{T}$, $\boldsymbol{I}$)。根据主私钥和用户身份 $\boldsymbol{I}$,生成用户的解密密钥。输入用户身份 $\boldsymbol{I}$,计算 $\boldsymbol{U}_{\boldsymbol{I}}=h(\boldsymbol{I})$($h$ 为 oracle 随机函数 $h:\{0,1\}^*\rightarrow\mathbf{Z}_q^{n\times l}$),$\boldsymbol{U}_{\boldsymbol{I}}=[\boldsymbol{u}_{\boldsymbol{I},1},\cdots,\boldsymbol{u}_{\boldsymbol{I},l}]$。然后利用 LPSF 中的原像采样算法 SamplePre($\boldsymbol{A}$, $\boldsymbol{T}$, s, $\boldsymbol{u}_{\boldsymbol{I},i}$)得到 $\boldsymbol{e}_{\boldsymbol{I},i}$。用户的解密密钥即为 $\boldsymbol{E}_{\boldsymbol{I}}=[\boldsymbol{e}_{\boldsymbol{I},1},\cdots,\boldsymbol{e}_{\boldsymbol{I},l}]$。

(3) Enc($\boldsymbol{A}$, $\boldsymbol{I}$, $\boldsymbol{m}$)。根据公钥 $\boldsymbol{A}$ 和用户 $\boldsymbol{I}$,加密消息 $\boldsymbol{m}$。首先计算 $\boldsymbol{U}_{\boldsymbol{I}}=h(\boldsymbol{I})$,然后随机采样 $\boldsymbol{a}\leftarrow\mathbf{Z}_q^n$ 和 $\boldsymbol{x}\leftarrow\chi^s$,计算 $\boldsymbol{p}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{a}+\boldsymbol{x}\in\mathbf{Z}_q^s$。对于消息 $\boldsymbol{m}\in\mathbf{Z}_q^l$,利用加密方案的加密算法 $\boldsymbol{w}=\mathrm{encode}(\boldsymbol{m})\in\mathbf{Z}_q^l$,输出密文为($\boldsymbol{p}$, $\boldsymbol{c}=\boldsymbol{U}^{\mathrm{T}}\boldsymbol{a}+\boldsymbol{w}$)。

(4) Dec($\boldsymbol{E}_{\boldsymbol{I}}$, $\boldsymbol{C}=(\boldsymbol{p},\boldsymbol{c})$)。用户通过自己的解密密钥对密文解密。计算 $\boldsymbol{g}=\boldsymbol{c}-\boldsymbol{E}_{\boldsymbol{I}}^{\mathrm{T}}\boldsymbol{p}\in\mathbf{Z}_q^l$,利用加密方案的解密算法进行解密 decode($\boldsymbol{g}$),最终得到明文 m。

4 结束语

本文介绍了基于格和理想格构建密码函数的方法。从 Ajtai 的哈希函数开始,详细描述了方案的构建思想、构造过程,逐步优化直到构造在实际应用中效率较高、安全性可证明、具有抗碰撞性质的 SWIFFT 哈希函数。由于格的优良性质,本文进一步描述了构造带陷门的哈希函数即带陷门原像采样函数的方法,详细的介绍了陷门的产生过程、函数的构造过程,并给出了安全性证明。最后,本文介绍了在基于格的密码函数的基础上构建签名方案和 IBE 方案的方法。

参考文献

[1] Ajtai M. Generating hard instances of lattice problems(extended abstract)[C]// 28th Annual ACM Symposium on Theory Of Computing(STOC'96). USA:ACM Press,1996:99-108.

[2] Gentry C,Peikert C,Vaikuntanathan V. Trapdoors for hard lattices and new cryptographic constructions[C]// 40th Annual ACM Symposium on Theory Of Computing(STOC 2008). Victoria BC,Canada:ACM Press,2008:197-206.

[3] Goldreich O,Goldwasser S,Halevi S. Eliminating decryption errors in the Ajtai-dwork cryptosystem[C]// CRYPTO. California,USA:Springer Press,1997:105-111.

[4] Arora S,Babai L,Stern J,et al. The hardness of approximate optima in lattices,codes,and systems of linear equations[J]. Journal of Computer and System Sciences,1997,54(2):317-331.

[5] Goldreich O,Goldwasser S. On the limits of non-approximability of lattice problems[J]. Journal of Computer and System Sciences,2000,60(3):540-563.

[6] Micciancio D. Generalized compact knapsacks,cyclic lattices,and efficient one-way functions[C]// Computational Complexity. Providence,Rhode Island:Springer,2007:365-411.

[7] Peikert C,Rosen A. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices[C]// Lecture Notes in Computer Science. New York:Springer,2006:145-166.

[8] Lyubashevsky V,Micciancio D,Peikert C,et al. SWIFFT:a modest proposal for FFT hashing[C]// Fast Software Encryption(FSE 2008). Lausaanne:Springer,2008:54-72.

[9] Ajtai M,Dwork C. A public-key cryptosystem with worst-case/average-case equivalence[C]// ACM Symposium on Theory Of Computing(STOC 1997). EI Paso:ACM Press,1997:284-293.

[10] Micciancio D,Regev O. Lattice-based cryptography[C]// Post Quantum Cryptography. Berlin-Heidleberg:Springer,2009:147-191.

[11] Stehle D,Steinfeld R,Tanaka K,et al. Efficient public key encryption based on ideal lattices[C]// ASIACRYPT 2009. Japan:Springer,2009:617-635.