格上基于身份的云存储完整性检测方案

田苗苗[1]，高闯[1]，陈洁[2]
（1. 安徽大学计算机科学与技术学院，安徽 合肥 230601；2. 华东师范大学计算机科学与软件工程学院，上海 200062）

摘 要：随着云存储的快速发展，越来越多的用户将数据存储在云端。为了验证云端的用户数据是否损坏，一种有效的方法是采用云存储完整性检测方案。利用理想格上的小整数解问题，设计了一种基于身份的云存储完整性检测方案，并在随机预言模型下证明了所提方案可以抵抗云服务器的适应性选择身份攻击。为了验证方案的效率，通过实验将所提方案与现有的 2 种基于身份的云存储完整性检测方案分别进行了比较。实验结果显示，所提方案的标签在线生成时间降低了 88.32%~93.74%，证据验证时间降低了 98.81%~99.73%。

关键词：格；基于身份的密码学；云存储；完整性检测
中图分类号：TP309
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2019073

Identity-based cloud storage integrity checking from lattices

TIAN Miaomiao[1], GAO Chuang[1], CHEN Jie[2]
1. School of Computer Science and Technology, Anhui University, Hefei 230601, China
2. School of Computer Science and Software Engineering, East China Normal University, Shanghai 200062, China

Abstract: With the rapid development of cloud storage, more and more users are storing their data in the cloud. To verify whether the users' data stored in the cloud is corrupted, one effective method is to adopt cloud storage integrity checking schemes. An identity-based cloud storage integrity checking scheme was proposed on the small integer solution problem over ideal lattices, and it was proven to be secure against the adaptive identity attacks of clouds in the random oracle model. To validate the efficiency of the scheme, extensive experiments were conducted to make performance- comparisons between the scheme and the existing two identity-based cloud storage integrity checking schemes. The experimental results show that the online tag-generation time and the proof-verification time of the scheme are respectively reduced by 88.32%~93.74% and 98.81%~99.73%.

Key words: lattice, identity-based cryptography, cloud storage, integrity checking

1 引言

云存储是当前应用最广泛的云计算服务之一。用户通过云存储服务将其数据外包给云服务器，之后可以在任何地方通过网络访问其外包数据，从而节省了本地存储开销，提高了使用数据的灵活性。随着越来越多的用户使用云存储服务，云存储安全风险越加凸显[1]，例如硬件故障、软件错误、操作失误、恶意破坏等都可能使云端数据出现损坏或缺失。由于一些商业原因，云存储发生数据损坏后用户通常不能及时获悉，因此用户必须主动检查存储在云服务器中数据的完整性。

由于云端外包数据的规模通常非常庞大，所以下载全部云端数据至本地进行完整性检测的方法是不切实际的。为此，研究人员提出了一些高效的云端数据完整性检测方法，本文统称为云存储完整性检测方案。云存储完整性检测方案通过交互式手段，无需下载全部数据即可验证云端外

收稿日期：2018−08−08；**修回日期**：2019−01−11
基金项目：国家自然科学基金资助项目（No. 61502443, No. 61772001, No.61472142, No.U1705264）
Foundation Item: The National Natural Science Foundation of China (No.61502443, No.61772001, No.61472142, No.U1705264)

包数据的完整性。具体地，用户在上传数据时会对数据进行分块处理并计算每块的标签，然后将数据和标签一同上传到云服务器。当验证云端数据的完整性时，验证者将发送一个随机挑战给云服务器，根据块标签的线性同态特性，云服务器能够计算出对应该挑战的一个较短证明。最后，验证者只需验证该证明是否正确，即可判定云端外包数据是否完整。

首个高效的云存储完整性检测方案在 2007 年由 Ateniese 等[2]和 Juels 等[3]分别提出，其中 Ateniese 等的方案支持公开验证，即允许任意的第三方验证者检测数据的完整性，并支持无限次的挑战询问；而 Juels 等的方案仅支持秘密验证，即只允许数据所有者对数据完整性进行检测，并且仅支持有限次的挑战询问。此后，新的云存储完整性检测方案[4-7]被相继提出，其中大多数方案都支持公开验证和无限次挑战询问。然而这些公开验证方案大都依赖公钥基础设施(PKI, public key infrastructure)，即需要使用数字证书来保证用户公钥和身份的对应关系，所以存在复杂的证书管理问题。

为了消除证书管理问题，一些基于身份的云存储完整性检测方案[8-10]被相继提出。在基于身份的密码方案中，用户的身份即为其公钥，对应的私钥由私钥生成中心(KGC，key generation center)生成[11]。上述方案的安全性都依赖于离散对数等传统数学问题的困难性，而这些数学问题都难以抵抗量子计算机的攻击[12]。由于格上困难问题具有安全性高、量子算法难以破解等优势[13]，使基于格的密码方案受到研究人员的广泛关注。

为了提高云存储完整性检测方案的安全性，近年来一些基于格问题的设计方案相继涌现。2012 年，Xu 等[14]利用小整数解（SIS, small integer solution）问题[15]设计了首个基于格的云存储完整性检测方案。2014 年，Liu 等[16]设计了一种格上支持公开验证的云存储完整性检测方案，但没有给出严格的安全性证明。随后，Zhang 等[17]指出该方案存在安全漏洞，恶意云服务器可以欺骗验证者通过完整性检测。2017 年，Liu 等[18]在文献[17]的基础上提出了一种格上基于身份的云存储完整性检测方案，但是由于该方案的块标签涉及云服务器的公钥，云服务器可以利用其私钥伪造数据，所以该方案实际上也不能抵抗恶意云服务器的攻击。

本文提出了一种新的格上基于身份的云存储完整性检测方案，并在随机预言模型下证明了该方案的安全性。与现有同类方案[18]相比，本文方案的系统参数和用户私钥都更短，计算效率也更高。本文的主要贡献如下。

1) 新的标签生成算法。受文献[19]的同态签名算法启发，本文设计了一种新的块标签生成算法。令公钥为 $\boldsymbol{a}\in\mathcal{R}_q^m$ 和辅助向量 $\boldsymbol{u}\in\mathcal{R}_q^d$，文件标识符为 τ，则第 i 块数据 $\boldsymbol{z}_i$ 的标签 $\boldsymbol{x}_i$ 是一个小向量，满足 $\boldsymbol{a}\boldsymbol{x}_i^{\mathrm{T}}=H(\tau,i)+\boldsymbol{z}_i\boldsymbol{u}^{\mathrm{T}}\pmod q$，其中 $H(\cdot)$ 是一个安全的散列函数。与文献[19]每次仅处理 1 bit 数据相比，本算法允许计算较大的数据块，因而能够减少文件的块标签总数，提高标签生成的效率。

2) 高效的私钥提取算法。现有同类方案的私钥提取算法较为复杂，需要较大的计算和存储开销。本文根据文献[20]提出的新陷门算法及相关算法，设计了一种高效的私钥提取算法。对于身份为 id 的用户 U_{id}，令 $\boldsymbol{a}_{\mathrm{id}}=\left[\boldsymbol{a}\middle|\tilde{H}(\mathrm{id})\right]$，其中，$\boldsymbol{a}$ 是私钥生成中心的公钥，$\tilde{H}(\cdot)$ 是一个安全的散列函数。利用文献[20]中的相关算法，私钥生成中心可以容易地为用户 U_{id} 生成较小的 $\boldsymbol{R}_{\mathrm{id}}$ 作为其私钥，其中 $\boldsymbol{R}_{\mathrm{id}}$ 满足 $\boldsymbol{a}\boldsymbol{R}_{\mathrm{id}}=\boldsymbol{g}-\tilde{H}(\mathrm{id})\pmod q$，$\boldsymbol{g}\in\mathcal{R}_q^k$ 是一个公开向量。利用函数 $f_{\boldsymbol{g}}(\boldsymbol{x})=\boldsymbol{g}\boldsymbol{x}^{\mathrm{T}}\pmod q$ 的高效求逆性和关系 $\boldsymbol{a}_{\mathrm{id}}\begin{bmatrix}\boldsymbol{R}_{\mathrm{id}}\\ \boldsymbol{I}_k\end{bmatrix}=\boldsymbol{g}$，用户 U_{id} 可以容易地计算任意文件块的标签。

3) 实用的理想格设计。本文的方案是基于多项式环 $\mathcal{R}_q=\mathbb{Z}_q[x]/(x^n+1)$ 结构的理想格[21]构造的。与基于一般格的方案相比，本文方案的系统参数和用户私钥长度都减少了 $O(n)$倍，计算开销也更低，更适合实际应用。

4) 严格的安全性证明。本文方案在随机预言模型下对云服务器的适应性选择身份攻击是安全的，其安全性可以规约到环上的小整数解问题[22-23]（Ring-SIS，ring small integer solution），该问题与理想格上最坏情况下的经典问题一样困难[22]。

2　预备知识

2.1　符号说明

本文的安全参数为 n；实数域和整数域分别记

为 $\mathbb{R}$和$\mathbb{Z}$。向量默认为行向量，用黑体小写字母表示；向量 $\boldsymbol{a}$ 的转置表示为 $\boldsymbol{a}^{\mathrm{T}}$。矩阵用黑体大写字母表示。向量 $\boldsymbol{a}=[a_1,a_2,\ldots,a_m]\in\mathbb{R}^m$ 的欧几里得范数为 $\|\boldsymbol{a}\|=\sqrt{\sum_{i=1}^{m}a_i^2}$，矩阵 $\boldsymbol{R}$ 的范数 $\|\boldsymbol{R}\|=\max_i\|\boldsymbol{r}_i^{\mathrm{T}}\|$，其中，$\boldsymbol{r}_i^{\mathrm{T}}$ 为 $\boldsymbol{R}$ 的第 i 个列向量。矩阵 $\boldsymbol{R}\in\mathbb{R}^{m\times w}$ 的最大奇异值，记为 $s_1(\boldsymbol{R})=\max_{\boldsymbol{x}}\|\boldsymbol{R}\boldsymbol{x}^{\mathrm{T}}\|$，其中 $\boldsymbol{x}$ 为 $\mathbb{R}^w$ 中的单位向量。矩阵 $\boldsymbol{R}$ 的 Gram-Schmidt 正交化记为 $\tilde{\boldsymbol{R}}$。矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times\bar{m}}$ 和 $\boldsymbol{B}\in\mathbb{R}^{n\times m}$ 的行连接记为 $[\boldsymbol{A}|\boldsymbol{B}]\in\mathbb{R}^{n\times(\bar{m}+m)}$；矩阵 $\boldsymbol{A}\in\mathbb{R}^{w\times m}$ 和 $\boldsymbol{B}\in\mathbb{R}^{k\times m}$ 的列连接记为 $[\boldsymbol{A},\boldsymbol{B}]$，即 $[\boldsymbol{A},\boldsymbol{B}]=\begin{bmatrix}\boldsymbol{A}\\ \boldsymbol{B}\end{bmatrix}\in\mathbb{R}^{(w+k)\times m}$。

对于任意的实数 $\lambda>0$，记 $\mathrm{negl}(n)=O\left(\frac{1}{n^{\lambda}}\right)$ 是以 n 为参数的可忽略函数。如果某事件发生的概率不低于 $1-\mathrm{negl}(n)$，则称该事件以极大概率成立。对于有限域 D 上的概率分布 X 和 Y，定义它们的统计距离为 $\Delta(X,Y)=\frac{1}{2}\sum_{z\in\mathrm{D}}|X(z)-Y(z)|$。如果统计距离 $\Delta(X,\ Y)=\mathrm{negl}(n)$，则称 X 和 Y 统计接近。

2.2 格

定义 1 设 $\boldsymbol{B}\in\mathbb{R}^{m\times m}$ 是由一组线性无关的向量所组成的矩阵，则由其生成的 m 维格定义为 $\Lambda=\mathcal{L}(\boldsymbol{B})=\{\boldsymbol{B}\boldsymbol{x}:\boldsymbol{x}\in\mathbb{Z}^m\}$，其中 $\boldsymbol{B}$ 被称为格 Λ 的基。

对正整数 m 和 q，矩阵 $\boldsymbol{A}\in\mathbb{R}_q^{n\times m}$，定义如下格。

$$\Lambda^{\perp}(\boldsymbol{A})=\{\boldsymbol{z}\in\mathbb{R}^m:\boldsymbol{A}\boldsymbol{x}=0\bmod q\} \tag{1}$$

对任意的向量 $\boldsymbol{u}\in\mathbb{R}_q^n$ 及满足 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{u}\bmod q$ 的任意向量 $\boldsymbol{v}\in\mathbb{R}^m$，定义

$$\Lambda^{\boldsymbol{u}}(\boldsymbol{A})=\{\boldsymbol{x}\in\mathbb{R}^m:\boldsymbol{A}\boldsymbol{x}=\boldsymbol{u}\bmod q\}=\Lambda^{\perp}(\boldsymbol{A})+\boldsymbol{v} \tag{2}$$

本文基于多项式环结构的理想格[21]。多项式环可表示为 $\mathcal{R}=\mathbb{Z}[x]/F(x)$，其中，$F(x)$ 为 n 阶分圆多项式，当 n 为 2 的幂时，$F(x)=x^n+1$。环 $\mathcal{R}$ 与 $\mathbb{Z}^n$ 是同构的，环 $\mathcal{R}$ 中的每个元素 $f=\sum_{i=0}^{n-1}a_ix^i$ 都对应于系数向量 $[a_0,a_1,\cdots,a_{n-1}]\in\mathbb{Z}^n$，范数 $\|f\|$ 表示为相应系数向量的范数。令整数 $q\geqslant 2$，q 模环表示为 $\mathcal{R}_q=\mathcal{R}/q\mathcal{R}$，环中多项式的系数 $a_i\in\mathbb{Z}_q$。本文采用的多项式环为 $\mathcal{R}_q=\mathbb{Z}_q[x]/(x^n+1)$。

2.3 离散高斯

离散高斯分布[15,24]是格密码学中的一个重要概念，下面介绍离散高斯的定义和相关性质。

定义 2 令参数 $s>0$，中心 $\boldsymbol{c}\in\mathbb{R}^m$ 的连续高斯函数 $\rho:\mathbb{R}^m\to(0,1]$ 为

$$\rho_{s,\boldsymbol{c}}(\boldsymbol{x})=\mathrm{e}^{-\pi\frac{\|\boldsymbol{x}-\boldsymbol{c}\|^2}{s^2}} \tag{3}$$

其中，格 Λ 上的以 s 为参数，$\boldsymbol{c}$ 为中心的离散高斯分布定义为 $D_{\Lambda,s,\boldsymbol{c}}(\boldsymbol{x})=\frac{\rho_{s,\boldsymbol{c}}(\boldsymbol{x})}{\rho_{s,\boldsymbol{c}}(\Lambda)}$，其中，$\boldsymbol{x}\in\Lambda$，$\rho_{s,\boldsymbol{c}}(\Lambda)=\sum_{\boldsymbol{v}\in\Lambda}\rho_{s,\boldsymbol{c}}(\boldsymbol{v})$。

当中心 $\boldsymbol{c}=0$ 时，将连续高斯分布和离散高斯分布分别简写为 $\rho_s(\boldsymbol{x})$ 和 $D_{\Lambda,s}(\boldsymbol{x})$。

平滑参数[15]是格中另一个重要概念。以下引理给出了平滑参数大小的一个上界[25]。

引理 1 设 $\boldsymbol{B}\in\mathbb{R}^{n\times n}$ 是格 $\Lambda\subset\mathbb{R}^n$ 的任意一组基，实数 $\varepsilon>0$，则平滑参数 $\eta_{\varepsilon}(\Lambda)$ 满足式(4)。

$$\eta_{\varepsilon}(\Lambda)\leqslant\|\tilde{\boldsymbol{B}}\|\sqrt{\frac{\ln\left(2n\left(1+\frac{1}{\varepsilon}\right)\right)}{\pi}} \tag{4}$$

特别地，对任意的函数 $\omega\left(\sqrt{\log n}\right)$，除可忽略的概率外，总有 $\eta_{\varepsilon}(\Lambda)\leqslant\|\tilde{\boldsymbol{B}}\|\omega\left(\sqrt{\log n}\right)$。

离散高斯分布有以下基本性质[24]。

引理 2 令格 $\Lambda\subset\mathbb{R}^m$，任意实数 $\varepsilon\in(0,1)$，向量 $\boldsymbol{c}\in\mathrm{span}(\Lambda)$，如果高斯参数 $s\geqslant\eta_{\varepsilon}(\Lambda)$，那么

$$\frac{1-\varepsilon}{1+\varepsilon}\rho_s(\Lambda)\leqslant\rho_s(\Lambda+\boldsymbol{c})\leqslant\rho_s(\Lambda) \tag{5}$$

$$\Pr\left[\|D_{\Lambda+\boldsymbol{c},s}\|\geqslant s\sqrt{m}\right]\leqslant 2^{-m}\frac{1+\varepsilon}{1-\varepsilon} \tag{6}$$

从 $\mathcal{R}^{m\times k}$ 中按照离散高斯分布选择的矩阵，其最大奇异值满足以下条件[21]。

引理 3 设实数 $s>0$，如果矩阵 $\boldsymbol{R}\leftarrow D_{\mathcal{R}^{m\times k},s}$，则 $s_1(\boldsymbol{R})\leqslant s\sqrt{n}O\left(\sqrt{m}+\sqrt{k}+\omega\left(\sqrt{\log n}\right)\right)$ 以极大概率成立，其中隐含的常数因子约为 $\frac{1}{\sqrt{2\pi}}$。

文献[21]给出了以下的平滑引理。

引理 4 给定整数 $n\geqslant 4$，$q\geqslant 2$ 和 $m\geqslant 2\lceil\log q\rceil+2$，实数 $s\geqslant\omega\sqrt{\log nm}$ 和随机向量 $\boldsymbol{a}\in\mathcal{R}_q^m$，

如果 $\boldsymbol{x} \leftarrow D_{\mathcal{R}^m,s}$，则 $\boldsymbol{a}\boldsymbol{x}^{\mathrm{T}}$ 统计接近于 $\mathcal{R}_q$ 上的均匀分布。

2.4 困难问题

本文方案的安全性基于 Ring-SIS 问题[22-23]，如定义 3 所示。

定义 3 给定随机向量 $\boldsymbol{a}=[a_1,a_2,\cdots,a_m]\in\mathcal{R}_q^m$，定义参数为 (q,m,β) 的 Ring-SIS 问题为：求非零向量 $\boldsymbol{x}=[x_1,x_2,\cdots,x_m]\in\mathcal{R}^m$，满足 $\|\boldsymbol{x}\|\leqslant\beta$，以及

$$\boldsymbol{a}\boldsymbol{x}^{\mathrm{T}}=\sum a_i x_i=0 \bmod q \tag{7}$$

文献[22]证明了当 $q\geqslant\beta\sqrt{n}\omega(\log n)$ 时，求解 Ring-SIS 问题的困难程度至少接近于求解理想格中最坏情况下因子为 $\gamma(n)\geqslant\beta\sqrt{n}\omega\left(\sqrt{\log n}\right)$ 的最短向量问题（SVP，shortest vectors problem）。

2.5 格陷门

格陷门是格密码学的重要工具之一。传统的格陷门是格的一组较短的基[25]。本节回顾文献[20]提出的新格陷门的概念及相关算法。

文献[20]指出，如果整数 $q\geqslant 2$，$k=\lceil\log q\rceil$，向量 $\boldsymbol{g}=\left[1,2,\cdots,2^{k-1}\right]\in\mathcal{R}_q^k$，则容易计算格 $\Lambda^{\perp}(\boldsymbol{g})$ 的一个短基 $\boldsymbol{S}\in\mathbb{Z}^{k\times k}$，从而对于任意的 $e\in\mathcal{R}_q$，计算小向量 $\boldsymbol{x}\in\mathcal{R}^k$ 满足 $\boldsymbol{g}\boldsymbol{x}^{\mathrm{T}}=e \bmod q$ 将是简单的。

定理 1 令整数 $q\geqslant 2$，$k=\lceil\log q\rceil$，格 $\Lambda^{\perp}(\boldsymbol{g})$ 的基 $\boldsymbol{S}\in\mathbb{Z}^{k\times k}$ 满足 $\|\tilde{\boldsymbol{S}}\|\leqslant\sqrt{5}$ 和 $\|\boldsymbol{S}\|\leqslant\max\left\{\sqrt{5},\sqrt{k}\right\}$；特别地，当 $q=2^k$ 时，$\boldsymbol{S}$ 满足 $\|\tilde{\boldsymbol{S}}\|\leqslant 2$ 和 $\|\boldsymbol{S}\|\leqslant\sqrt{5}$。

文献[20]提出了 $\boldsymbol{g}$–陷门的概念，如定义 4 所示。

定义 4 令向量 $\boldsymbol{a}\in\mathcal{R}_q^m$，$\boldsymbol{g}\in\mathcal{R}_q^k$，其中整数 $m>k$，$k=\lceil\log q\rceil$。$\boldsymbol{a}$ 的 $\boldsymbol{g}$–陷门是最大奇异值较小的矩阵 $\boldsymbol{R}\in\mathcal{R}^{(m-k)\times k}$ 满足 $\boldsymbol{a}[\boldsymbol{R},\boldsymbol{I}_k]=h\boldsymbol{g}$，其中 $h\in\mathcal{R}_q$ 为可逆多项式，称为 $\boldsymbol{R}$ 的标签。

文献[20]给出了一种 $\boldsymbol{g}$–陷门生成算法。

引理 5 令整数 $w\geqslant 1$，$q\geqslant 2$，$k=\lceil\log q\rceil$，存在一个多项式时间算法 GenTrap(h,σ)，输入可逆多项式 $h\in\mathcal{R}_q$ 和高斯参数 $\sigma>\omega\left(\sqrt{\log nw}\right)$，输出统计接近于 $\mathcal{R}_q$ 上均匀分布的向量 $\boldsymbol{a}\in\mathcal{R}_q^m$ 及对应标签 h 的 $\boldsymbol{g}$–陷门 $\boldsymbol{R}\leftarrow D_{\mathcal{R}^{wk},\sigma}$，其中 $m=w+k$。

根据引理 3 可知，陷门 $\boldsymbol{R}$ 将以极大概率满足 $s_1(\boldsymbol{R})\leqslant s\sqrt{n}O\left(\sqrt{w}+\sqrt{k}+\omega\left(\sqrt{\log n}\right)\right)$。

令高斯参数 $s\geqslant\sqrt{7\left(s_1(\boldsymbol{R})^2+1\right)}\omega\left(\sqrt{\log n}\right)$，对任意的 $e\in\mathcal{R}_q$，利用 $\boldsymbol{a}$ 的 $\boldsymbol{g}$–陷门 $\boldsymbol{R}\in\mathcal{R}^{w\times k}$ 可计算小向量 $\boldsymbol{x}\in\mathcal{R}^m$ 满足 $\boldsymbol{a}\boldsymbol{x}^{\mathrm{T}}=e \bmod q$。具体过程如下。

首先选择扰动向量 $\boldsymbol{p}\leftarrow D_{\mathcal{R}^m,\sqrt{\Sigma_p}}$，其中 $\Sigma_p=s^2\boldsymbol{I}-5\omega(\log n)[\boldsymbol{R},\boldsymbol{I}]\left[\boldsymbol{R}^{\mathrm{T}}\mid\boldsymbol{I}\right]$，$\sqrt{\Sigma_p}$ 的计算采用 Cholesky 分解方法；根据 $\Lambda^{\perp}(\boldsymbol{g})$ 的基 $\boldsymbol{S}$ 计算向量 $\boldsymbol{z}\in\mathcal{R}^k$ 使 $\boldsymbol{g}\boldsymbol{z}^{\mathrm{T}}=h^{-1}\left(e-\boldsymbol{a}\boldsymbol{p}^{\mathrm{T}}\right) \bmod q$；最后输出 $\boldsymbol{x}^{\mathrm{T}}=\boldsymbol{p}^{\mathrm{T}}+[\boldsymbol{R},\boldsymbol{I}_k]\boldsymbol{z}^{\mathrm{T}}$。容易验证如式(8)所示的等式。

$$\begin{aligned}&\boldsymbol{a}\boldsymbol{x}^{\mathrm{T}}=\boldsymbol{a}\boldsymbol{p}^{\mathrm{T}}+\boldsymbol{a}[\boldsymbol{R},\boldsymbol{I}_k]\boldsymbol{z}^{\mathrm{T}}=\boldsymbol{a}\boldsymbol{p}^{\mathrm{T}}+h\boldsymbol{g}\boldsymbol{z}^{\mathrm{T}}=\\&\boldsymbol{a}\boldsymbol{p}^{\mathrm{T}}+hh^{-1}\left(e-\boldsymbol{a}\boldsymbol{p}^{\mathrm{T}}\right)=e \bmod q\end{aligned} \tag{8}$$

正式地，文献[20]给出了原像采样引理，如引理 6 所示。

引理 6 令整数 $q\geqslant 2$，$k=\lceil\log q\rceil$，$m>k$，存在一个多项式时间算法 SampleD$(\boldsymbol{a},\boldsymbol{R},e,s)$，输入向量 $\boldsymbol{a}\in\mathcal{R}_q^m$，可逆标签 $h\in\mathcal{R}_q$ 及其所对应的 $\boldsymbol{g}$–陷门 $\boldsymbol{R}\in\mathcal{R}^{(m-k)\times k}$，任意的多项式 $e\in\mathcal{R}_q$ 以及高斯参数 $s\geqslant\sqrt{7\left(s_1(\boldsymbol{R})^2+1\right)}\omega\left(\sqrt{\log n}\right)$，算法输出统计接近分布 $D_{\Lambda^e(\boldsymbol{a}),s}$ 的向量 $\boldsymbol{x}\in\mathcal{R}^m$ 且满足 $\boldsymbol{a}\boldsymbol{x}^{\mathrm{T}}=e \bmod q$。

利用上述采样算法可得以下陷门委派算法[20]，如引理 7 所示。

引理 7 令整数 $w\geqslant 1$，$q\geqslant 2$，$k=\lceil\log q\rceil$ 和 $m=w+k$，向量 $\boldsymbol{a}\in\mathcal{R}_q^m$ 和 $\boldsymbol{a}_1\in\mathcal{R}_q^k$，存在一个多项式时间算法 DelTrap$(\boldsymbol{a}',\boldsymbol{R},h,s)$，输入向量 $\boldsymbol{a}'=[\boldsymbol{a}|\boldsymbol{a}_1]\in\mathcal{R}_q^{m+k}$，$\boldsymbol{a}$ 的 $\boldsymbol{g}$–陷门 $\boldsymbol{R}\in\mathcal{R}^{w\times k}$ 和相应的标签 $h\in\mathcal{R}_q$，以及高斯参数 s 满足 $s\geqslant\sqrt{7\left(s_1(\boldsymbol{R})^2+1\right)}\omega\left(\sqrt{\log n}\right)$，算法输出 $\boldsymbol{a}'$ 的 $\boldsymbol{g}$–陷门 $\boldsymbol{R}'\in\mathcal{R}^{m\times k}$ 统计接近于分布 $D_{\Lambda^t(\boldsymbol{a}),s}$，且满足 $\boldsymbol{a}\boldsymbol{R}'=\boldsymbol{t}\ (\bmod q)$，其中 $\boldsymbol{t}=\boldsymbol{g}-\boldsymbol{a}_1$。

为简便起见，本文将标签 h 均设为 1。

3 系统描述

3.1 系统模型

如图 1 所示，基于身份的云存储完整性检测方案由 4 个实体组成，分别为用户、私钥生成中心、云服务器和审计者。用户是数据所有者，通过云存

储服务将其数据外包给云服务器。私钥生成中心根据用户的身份为用户分配私钥。云服务器拥有巨大的存储空间和计算资源，可以为用户提供云存储服务。审计者是一个专业实体，可以代表用户对存储在云服务器中的数据进行完整性检测，并及时将审计结果返回给用户。

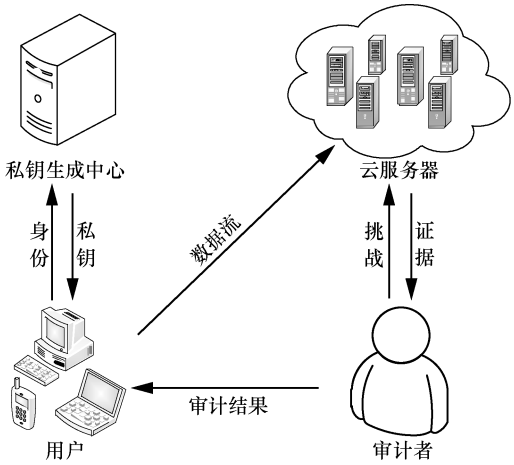

图 1 基于身份的云存储完整性检测方案模型

不失一般性，本文假定私钥生成中心与用户之间、用户与云服务器之间，以及云服务器与审计者之间均有安全的传输通道。同时，假定每个用户只有一个文件存储在云端。

定义 5 基于身份的云存储完整性检测方案由 6 个概率多项式时间算法组成，分别为系统建立（setup）算法，私钥提取（extract）算法、标签生成（taggen）算法、审计（audit）算法，证据生成（prove）算法和证据验证（verify）算法。

setup 算法：该算法输入为安全参数 n，输出为系统公钥 mpk 和主私钥 msk。

extract 算法：该算法由私钥生成中心执行。输入系统公钥 mpk，主私钥 msk 和用户的身份 id，输出对应的用户私钥 $\mathrm{sk}_{\mathrm{id}}$。

taggen 算法：该算法由用户执行。输入为系统公钥 mpk，用户的私钥 $\mathrm{sk}_{\mathrm{id}}$ 和文件 F_{id}，输出为文件的标识符 τ_{id}，总块数 L_{id} 和块标签集合 Φ_{id}。最后，用户将 $\{\mathrm{id},\tau_{\mathrm{id}},F_{\mathrm{id}},\Phi_{\mathrm{id}}\}$ 上传到云服务器，将 $\{\mathrm{id},\tau_{\mathrm{id}},L_{\mathrm{id}}\}$ 发送给审计者，然后删除本地副本。

audit 算法：该算法由审计者执行。审计者根据用户身份 id，文件的唯一标识符 τ_{id} 和文件总块数 L_{id}，输出为一个随机挑战 chal。

prove 算法：该算法由云服务器执行。输入挑战 chal，文件 F_{id} 及其块标签集合 Φ_{id}，输出证据 P。

verify 算法：该算法由审计者执行。输入系统公钥 mpk，用户的身份 id，审计挑战 chal 及对应的证据 P，输出“0”或“1”。输出“0”表示云服务器中的文件是不完整的，输出“1”则表示文件是完整的。

显然，对于一个正确的基于身份的云存储完整性检测方案来说，如果方案的所有参与方都诚实，那么算法 verify 应该以极大概率输出“1”。

3.2 安全模型

本文假设私钥生成中心、用户和审计者均是可信的，而云服务器是半可信的，即在数据已损坏的情况下，云服务器可能通过给出有效的证据来极力隐藏数据损坏的事实。根据文献[4]的结论，本文仅需考虑方案的顽健性，它是指云服务器应难以给出一个与诚实证据不同且能通过审计者检查的证据。

本文考虑适应性选择身份攻击下的顽健性，如定义 6 所述，其中挑战者 $\mathcal{C}$ 模拟私钥生成中心、用户和审计者，而敌手 $\mathcal{A}$ 模拟云服务器。

定义 6 如果不存在多项式时间的敌手 $\mathcal{A}$ 能够以不可忽略的概率赢得以下游戏，则称该基于身份的云存储完整性检测方案满足适应性选择身份攻击下的顽健性。游戏由 4 个阶段组成，分别为建立阶段、询问阶段、审计阶段和伪造阶段。

1) 建立阶段。在该阶段，挑战者 $\mathcal{C}$ 扮演私钥生成中心的角色。挑战者 $\mathcal{C}$ 执行 setup 算法生成系统公钥 mpk 和主私钥 msk，并将 mpk 发送给敌手 $\mathcal{A}$。

2) 询问阶段。在该阶段，挑战者 $\mathcal{C}$ 扮演私钥生成中心和用户的角色。敌手 $\mathcal{A}$ 可以适应性地向挑战者 $\mathcal{C}$ 进行提取询问和标签询问。

① 提取询问。敌手 $\mathcal{A}$ 询问身份 id 对应的私钥。挑战者 $\mathcal{C}$ 执行算法 extract 获得私钥 $\mathrm{sk}_{\mathrm{id}}$，并将其发送给敌手 $\mathcal{A}$。

② 标签询问。敌手 $\mathcal{A}$ 询问身份为 id 的用户 U_{id} 对文件 F_{id} 的标签。挑战者 $\mathcal{C}$ 执行算法 taggen 生成对应的块标签集合 Φ_{id}，并将其发送给敌手 $\mathcal{A}$。

3) 审计阶段。在该阶段，挑战者 $\mathcal{C}$ 扮演审计者的角色。对于已经执行过标签询问的文件，挑战者 $\mathcal{C}$ 执行 audit 算法生成一个随机挑战 chal，并发送给敌手 $\mathcal{A}$。然后敌手 $\mathcal{A}$ 执行 prove 算法生成相应

的证据 P，并将其发送给挑战者 $\mathcal{C}$。

4) 伪造阶段。挑战者 $\mathcal{C}$ 提交关于身份 id^*的挑战询问 chal^*敌手，敌手 $\mathcal{A}$ 输出一个相应的证据 P^*。如果 P^*与诚实的证据不同，$\mathcal{A}$ 未对 id^*进行过提取询问且 $\mathrm{verify}(\mathrm{mpk},\mathrm{id}^*,\mathrm{chal}^*,P^*)=1$，则敌手 $\mathcal{A}$ 赢得游戏。

4　方案描述

4.1　具体构造

本文提出的格上基于身份的云存储完整性检测方案如下。

1) 系统建立算法

setup：输入安全参数 n，选取整数 q、$p\geqslant 2$，$k=\lceil \log q \rceil$，$d>1$，$\bar{m}\geqslant 2k+2$，$m=\bar{m}+k$，3 个高斯参数 σ、s、s'，向量 $\boldsymbol{g}=\left[1,2,\cdots,2^{k-1}\right]\in\mathcal{R}_q^k$，以及 2 个安全的散列函数 $H_1:\{0,1\}^*\to\mathcal{R}_q^k$ 和 $H_2:\{0,1\}^*\to\mathcal{R}_q$。执行下列操作。

① 运行算法 $\mathrm{GenTrap}(1,\sigma)\to(\boldsymbol{a},\boldsymbol{R})$。

② 选取随机向量 $\boldsymbol{u}\in\mathcal{R}_q^d$。

③ 输出系统公钥 $\mathrm{mpk}=(\boldsymbol{a},\boldsymbol{u})$ 和主私钥 $\mathrm{msk}=\boldsymbol{R}$。

2) 私钥提取算法

extract：输入系统公钥 mpk、主私钥 msk 和身份 $\mathrm{id}\in\{0,1\}^*$，执行下列步骤。

① 计算 $\boldsymbol{h}_{\mathrm{id}}=H_1(\mathrm{id})$。

② 令 $\boldsymbol{a}_{\mathrm{id}}=[\boldsymbol{a}\,|\,\boldsymbol{h}_{\mathrm{id}}]\in\mathcal{R}_q^m$。

③ 运行算法 $\mathrm{DelTrap}(\boldsymbol{a}_{\mathrm{id}},\boldsymbol{R},s)\to\boldsymbol{R}_{\mathrm{id}}$。

④ 输出用户私钥 $\mathrm{sk}_{\mathrm{id}}=\boldsymbol{R}_{\mathrm{id}}$。

3) 标签生成算法

taggen：输入系统公钥 mpk、用户私钥 $\mathrm{sk}_{\mathrm{id}}$ 和文件 F_{id}，用户执行下列步骤。

① 选取随机的 $\tau_{\mathrm{id}}\in\mathbb{Z}_q^n$ 作为文件 F_{id} 的标识符。

② 将文件 F_{id} 分为 L_{id} 块，其中每块均属于 $\mathcal{R}_p^d$，即 $F_{\mathrm{id}}=\left\{\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_{L_{\mathrm{id}}}\right\}$ 且 $\boldsymbol{z}_i\in\mathcal{R}_p^d$。

③ 计算 $\alpha_i=\mathrm{H}_2(\tau_{\mathrm{id}},i)$，其中，$i=1,2,\cdots,L_{\mathrm{id}}$。

④ 计算每块的标签，其中第 i 块 $\boldsymbol{z}_i$ 的标签

$$\boldsymbol{x}_i=\mathrm{SampleD}\left(\boldsymbol{a}_{\mathrm{id}},\boldsymbol{R}_{\mathrm{id}},\alpha_i+\boldsymbol{z}_i\boldsymbol{u}^{\mathrm{T}},s'\right)\in\mathcal{R}^m$$

⑤ 输出标签集合 $\Phi_{\mathrm{id}}=\left\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_{L_{\mathrm{id}}}\right\}$。

⑥ 将 $\{\mathrm{id},\tau_{\mathrm{id}},L_{\mathrm{id}}\}$ 和 $\{\mathrm{id},\tau_{\mathrm{id}},\Phi_{\mathrm{id}},F_{\mathrm{id}}\}$ 分别发送给审计者和云服务器，然后删除本地副本。

4) 审计算法

audit：输入身份 id、标识符 $\tau_{\mathrm{id}}\in\mathbb{Z}_q^n$ 和文件的总块数 L_{id}，审计者执行下列步骤。

① 从 $[1,L_{\mathrm{id}}]$ 中随机选取 ℓ 个整数作为挑战数据块的序号，记为 $I=\{r_1,r_2,\cdots,r_\ell\}$，并为每个序号选取随机权重 $v_i\leftarrow\mathbb{Z}_p^*$。

② 发送 $\mathrm{chal}=\left(\mathrm{id},\tau_{\mathrm{id}},\{i,v_i\}_{i\in I}\right)$ 给云服务器。

5) 证据生成算法

prove：当收到来自审计者的挑战 $\mathrm{chal}=\left(\mathrm{id},\tau_{\mathrm{id}},\{i,v_i\}_{i\in I}\right)$ 后，云服务器找到对应的文件 F_{id} 及标签集合 Φ_{id}，计算 $\boldsymbol{x}=\sum_{i\in I}v_i\boldsymbol{x}_i$ 和 $\boldsymbol{z}=\sum_{i\in I}v_i\boldsymbol{z}_i$，然后将证据 $P=(\boldsymbol{x},\boldsymbol{z})$ 返回。

6) 证据验证算法

verify：输入系统公钥 mpk，随机挑战 chal 和证据 P，审计者执行下列步骤。

① 解析 $\mathrm{chal}=\left(\mathrm{id},\tau_{\mathrm{id}},\{i,v_i\}_{i\in I}\right)$ 和 $P=(\boldsymbol{x},\boldsymbol{z})$。

② 计算 $\boldsymbol{h}_{\mathrm{id}}=H_1(\mathrm{id})$ 和 $\boldsymbol{a}_{\mathrm{id}}=[\boldsymbol{a}\,|\,\boldsymbol{h}_{\mathrm{id}}]\in\mathcal{R}_q^m$。

③ 令 $\alpha=\sum_{i\in I}v_i\alpha_i \mod q$，其中 $\alpha_i=H_2(\tau_{\mathrm{id}},i)$。

④ 计算 $B=\ell(p-1)^2+1$。

⑤ 判断下列条件是否均成立。

a) $\|\boldsymbol{x}\|\leqslant\ell(p-1)s'\sqrt{nm}$ 且 $\boldsymbol{z}\in\mathcal{R}_B^d$。

b) $\boldsymbol{a}_{\mathrm{id}}\boldsymbol{x}^{\mathrm{T}}=\alpha+\boldsymbol{z}\boldsymbol{u}^{\mathrm{T}}\pmod q$。

若是，则返回“1”；否则，返回“0”。

4.2　参数设置

本文的安全参数 n 设为 2 的幂次。根据引理 1，设函数 $\omega\left(\sqrt{\log n}\right)$ 为 $\sqrt{\dfrac{\ln\left(2n\left(1+\dfrac{1}{\varepsilon}\right)\right)}{\pi}}$。

根据引理 5，GenTrap 算法的高斯参数设为 $\sigma\geqslant\omega\left(\sqrt{\log n}\right)$，则系统公钥 $\boldsymbol{a}\in\mathcal{R}_q^{\bar{m}}$ 统计接近于 $\mathcal{R}_q^{\bar{m}}$ 上的均匀分布，主私钥 $\boldsymbol{R}\leftarrow D_{\mathcal{R}^{(\bar{m}-k)\times k},\sigma}$。因此根据引理 3，$s_1(\boldsymbol{R})\leqslant\sigma\sqrt{n}O\left(\sqrt{\bar{m}-k}+\sqrt{k}+\omega\left(\sqrt{\log n}\right)\right)$ 以极大概率成立，其中隐含的常数因子约为 $\dfrac{1}{\sqrt{2\pi}}$。

根据引理 7，DelTrap 算法的高斯参数设为 $s\geqslant\sqrt{7\left(s_1(\boldsymbol{R})^2+1\right)}\omega\left(\sqrt{\log n}\right)$，用户私钥 $\boldsymbol{R}_{\mathrm{id}}\in\mathcal{R}^{\bar{m}\times k}$

统计接近于分布 $D_{\Lambda^{\perp}(\boldsymbol{a}),s}$，其中 $\boldsymbol{t}=\boldsymbol{g}-\boldsymbol{h}_{\mathrm{id}}$。根据引理 2 和引理 3，$s_1(\boldsymbol{R}_{\mathrm{id}})\leqslant s\sqrt{n}O\left(\sqrt{m}+\sqrt{k}+\omega\left(\sqrt{\log n}\right)\right)$ 也以极大概率成立。

根据引理 6，SampleD 算法的高斯参数设为 $s'\geqslant\sqrt{7\left(s_1(\boldsymbol{R}_{\mathrm{id}})^2+1\right)}\omega\left(\sqrt{\log n}\right)$，块标签 $\boldsymbol{x}_i\in\mathcal{R}^m$ 统计接近于 $D_{\Lambda^{\boldsymbol{e}}(\boldsymbol{a}_{id}),s'}$，其中 $\boldsymbol{e}=\alpha_i+\boldsymbol{z}_i\boldsymbol{u}^{\mathrm{T}}$。根据引理 2，$\|\boldsymbol{x}_i\|\leqslant s'\sqrt{nm}$ 以极大概率成立。

为了确保 Ring-SIS 规约的困难性，相关参数须满足 $q\geqslant\beta\sqrt{n}\omega(\log n)$。令审计挑战选择的元素个数为 ℓ，$\beta=\ell(p-1)\sqrt{n}\left(2s'\sqrt{m}+(p-1)\sqrt{d}\right)$，$p\geqslant 2$，则 q 须满足如下关系。

$$q\geqslant\ell(p-1)n\left(2s'\sqrt{m}+(p-1)\sqrt{d}\right)\omega(\log n)$$

5 方案分析

5.1 正确性

定理 2 如果本文方案中的用户、私钥生成中心、云服务器和审计者都是诚实的，那么 verify 算法将以极大概率输出“1”。

证明 给定用户的身份 id 和文件 F_{id}，令文件标识符为 τ_{id}。对第 i 个数据块 $\boldsymbol{z}_i$，计算 $\alpha_i=H_2(\tau_{\mathrm{id}},i)$，执行算法 $\boldsymbol{x}_i=\mathrm{SampleD}(\boldsymbol{R}_{\mathrm{id}},\boldsymbol{a}_{\mathrm{id}},\alpha_i+\boldsymbol{z}_i\boldsymbol{u}^{\mathrm{T}},s')$。根据引理 2 及引理 6，标签 $\boldsymbol{x}_i$ 将以极大概率满足 $\|\boldsymbol{x}_i\|\leqslant s'\sqrt{nm}$ 和 $\boldsymbol{a}_{\mathrm{id}}\boldsymbol{x}_i^{\mathrm{T}}=\alpha_i+\boldsymbol{z}_i\boldsymbol{u}^{\mathrm{T}}$。

令随机挑战 $\mathrm{chal}=\left(\tau_{\mathrm{id}},\{i,v_i\}_{i\in I}\right)$，假设 I 包含的元素个数为 ℓ，云服务器返回的证据为 $P=(\boldsymbol{x},\boldsymbol{z})$，其中，$\boldsymbol{x}=\sum_{i\in I}v_i\boldsymbol{x}_i$，$\boldsymbol{z}=\sum_{i\in I}v_i\boldsymbol{z}_i$。显然 $\boldsymbol{z}\in\mathcal{R}_B^d$ 成立，其中 $B=\ell(p-1)^2+1$。同时，由三角不等式可得

$$\|\boldsymbol{x}\|=\left\|\sum_{i\in I}v_i\boldsymbol{x}_i\right\|\leqslant\ell(p-1)s'\sqrt{nm}\tag{9}$$

因此，verify 算法中的条件 a)成立。

另一方面，令 $\alpha=\sum_{i\in I}v_i\alpha_i\pmod q$，则有

$$\begin{aligned}&\boldsymbol{a}_{\mathrm{id}}\boldsymbol{x}^{\mathrm{T}}=\boldsymbol{a}_{\mathrm{id}}\sum_{i\in I}v_i\boldsymbol{x}_i^{\mathrm{T}}=\sum_{i\in I}v_i\left(\boldsymbol{a}_{\mathrm{id}}\boldsymbol{x}_i^{\mathrm{T}}\right)=\\&\sum_{i\in I}v_i\left(\alpha_i+\boldsymbol{z}_i\boldsymbol{u}^{\mathrm{T}}\right)=\sum_{i\in I}v_i\alpha_i+\left(\sum_{i\in I}v_i\boldsymbol{z}_i\right)\boldsymbol{u}^{\mathrm{T}}=\\&\alpha+\boldsymbol{z}\boldsymbol{u}^{\mathrm{T}}\pmod q\end{aligned}\tag{10}$$

因此，verify 算法中的条件 b)也成立。

5.2 顽健性

定理 3 如果 Ring-SIS 问题是困难的，那么在随机预言模型下任何多项式时间的敌手都不能以不可忽略的概率破解本文方案的顽健性。

证明 根据定义 6，假设存在一个多项式时间的敌手 $\mathcal{A}$ 能够以不可忽略的概率 ε 破解本文方案的顽健性，则下面证明，存在一个多项式时间的算法 $\mathcal{B}$，能以不低于 $\frac{\varepsilon(Q_{\mathrm{H}}-Q_{\mathrm{E}})}{Q_{\mathrm{H}}^2}$ 的概率破解 Ring-SIS 问题，其中，Q_{H} 是 H_1 询问的最大次数，Q_{E} 是提取询问的最大次数。具体过程如下。

1) 建立阶段。给定安全参数 n，令 $\ell\geqslant 1$ 为单次审计挑战选择的元素总个数，算法 $\mathcal{B}$ 选择随机整数 q、$p\geqslant 2$，$k=\lceil\log q\rceil$，$\bar{m}\geqslant 2k+2$，$m=\bar{m}+k$，$d>1$，$B=\ell(p-1)^2+1$，公开向量 $\boldsymbol{g}=[1,2,\cdots,2^{k-1}]\in\mathcal{R}_q^k$，高斯参数 $s\geqslant 2\sqrt{nk}\omega(\log n)$ 和 $s'\geqslant 7nk\omega\left(\log^{\frac{3}{2}}n\right)$，以及 2 个作为随机预言机的散列函数 $H_1:\{0,1\}^*\to\mathcal{R}_q^k$ 和 $H_2:\{0,1\}^*\to\mathcal{R}_q$。输入随机向量 $\bar{\boldsymbol{a}}\in\mathcal{R}_q^{m+d}$ 作为 Ring-SIS 实例，令 $[\boldsymbol{a}\,|\,\boldsymbol{b}\,|\,\boldsymbol{u}]=\bar{\boldsymbol{a}}$，实数 $\beta=\ell(p-1)\sqrt{n}\left(2s'\sqrt{m}+(p-1)\sqrt{d}\right)$，则算法 $\mathcal{B}$ 试图寻找一个非零向量 $\boldsymbol{y}\in\mathcal{R}^{m+d}$，使得 $\|\boldsymbol{y}\|\leqslant\beta$ 且 $\bar{\boldsymbol{a}}\boldsymbol{y}^{\mathrm{T}}=0\bmod q$。算法 $\mathcal{B}$ 最后将系统公钥 $\mathrm{mpk}=(\boldsymbol{a},\boldsymbol{u})$ 发送给敌手 $\mathcal{A}$。

2） 询问阶段。敌手 $\mathcal{A}$ 可以适应性地向算法 $\mathcal{B}$ 进行一系列询问，包括散列询问、提取询问和标签询问。在这个阶段，算法 $\mathcal{B}$ 将建立并维持多个表，其中表 L_1 用于记录 H_1 询问，表 L_2 用于记录 H_2 询问，表 L_t 用于记录标签询问。

① H_1 询问。敌手 $\mathcal{A}$ 可以适应性地对身份进行 H_1 询问。不失一般性，假设 H_1 询问在其他类型询问之前且敌手每次提交 H_1 询问的身份都不同。算法 $\mathcal{C}$ 随机选取 $j\in\{1,\cdots,Q_{\mathrm{H}}\}$。对敌手 $\mathcal{A}$ 的第 i 次 H_1 询问，如果 $i=j$，则算法 $\mathcal{B}$ 将 $(\mathrm{id},\boldsymbol{b},\perp)$ 存于表 L_1 中，并将 $\boldsymbol{b}$ 返回给敌手 $\mathcal{A}$。如果 $i\neq j$，则算法 $\mathcal{B}$ 按照分布 $D_{\mathcal{R}^{\bar{m}},s}$ 选取 k 个随机的列向量构成 $\boldsymbol{R}_{\mathrm{id}}\in\mathcal{R}^{\bar{m}\times k}$，计算 $\boldsymbol{h}_{\mathrm{id}}=\boldsymbol{g}-\boldsymbol{a}\boldsymbol{R}_{\mathrm{id}}$。若 $\boldsymbol{h}_{\mathrm{id}}$ 已在表 L_1 中，则重新选取 $\boldsymbol{R}_{\mathrm{id}}$。最后，算法 $\mathcal{B}$ 将 $(\mathrm{id},\boldsymbol{h}_{\mathrm{id}},\boldsymbol{R}_{\mathrm{id}})$ 存于表 L_1 中，并将 $\boldsymbol{h}_{\mathrm{id}}$ 返回给敌手 $\mathcal{A}$。根据引理 4，$\boldsymbol{h}_{\mathrm{id}}$ 统计接近于均

匀分布。

② 提取询问。敌手 $\mathcal{A}$ 向算法$\mathcal{B}$适应性地进行提取询问，以获取身份 id 对应的私钥。此时，算法$\mathcal{B}$检索表 L_1 找到匹配项。若匹配项为$(\mathrm{id},\boldsymbol{h}_{\mathrm{id}},\boldsymbol{R}_{\mathrm{id}})$，则返回 $\boldsymbol{R}_{\mathrm{id}}$；若匹配项为$(\mathrm{id},\boldsymbol{b},\perp)$，则终止。

③ H_2 询问。算法$\mathcal{B}$建立并维护一个表 L_2 用于响应敌手 $\mathcal{A}$ 的 H_2 询问。一般地，对敌手 $\mathcal{A}$ 的一个 H_2 询问，算法$\mathcal{B}$首先查找表 L_2，如果找到匹配项则直接将结果返回，否则选择一个随机的 $\alpha\in\mathcal{R}_q$ 并返回。如果 $\mathcal{A}$ 的 H_2 询问形如(τ_{id},i)，则算法$\mathcal{B}$将以一种特殊方式进行响应，具体过程见标签询问。

④ 标签询问。敌手 $\mathcal{A}$ 适应性地进行标签询问，以获得身份为 id 的用户 U_{id} 关于文件 F_{id} 的标签。此时，算法$\mathcal{B}$首先查找表 L_t，如果找到匹配项$(\mathrm{id},\tau_{\mathrm{id}},\Phi_{\mathrm{id}},F_{\mathrm{id}})$，则直接将其返回给敌手 $\mathcal{A}$。否则算法$\mathcal{B}$执行下列步骤：首先选取随机的$\tau_{\mathrm{id}}\in\mathbb{Z}_q^n$作为文件 F_{id} 的标识符，并将文件 F_{id} 拆分为 L_{id} 块 $F_{\mathrm{id}}=\{\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_{L_{\mathrm{id}}}\}$，其中，每块$\boldsymbol{z}_i\in\mathcal{R}_p^d$；然后计算标签集合$\Phi_{\mathrm{id}}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_{L_{\mathrm{id}}}\}$，其中，$\boldsymbol{x}_i\in\mathcal{R}^m$是数据块 $\boldsymbol{z}_i$ 的标签；最后将$(\mathrm{id},\tau_{\mathrm{id}},\Phi_{\mathrm{id}},F_{\mathrm{id}})$存于表 L_t 中，并将其返回给敌手 $\mathcal{A}$。为了计算每个块标签，算法$\mathcal{B}$按照离散高斯分布 $D_{\mathcal{R}^m,s'}$ 随机选取 $\boldsymbol{x}_i$，计算$\alpha_i=[\boldsymbol{a}|\boldsymbol{h}_{\mathrm{id}}]\boldsymbol{x}_i^{\mathrm{T}}-\boldsymbol{z}_i\boldsymbol{u}^{\mathrm{T}}\ (\mathrm{mod}\ q)$，将$(\tau_{\mathrm{id}},i,\alpha_i)$存于表 L_2 中，并令 $H_2(\tau_{\mathrm{id}},i)=\alpha_i$。根据引理 4，$\alpha_i$ 统计接近于均匀分布。

3) 审计阶段。算法$\mathcal{B}$可以对已执行过标签询问的文件进行审计询问。假设算法$\mathcal{B}$希望检测外包文件 F_{id} 的完整性。输入文件所有者的身份 id，标识符 τ_{id}，总块数 L_{id}，算法$\mathcal{B}$执行算法 $\mathrm{audit}(\mathrm{id},\tau_{\mathrm{id}},L_{\mathrm{id}})$，生成挑战 $\mathrm{chal}=(\mathrm{id},\tau_{\mathrm{id}},\{i,v_i\}_{i\in I})$ 并将其发送给敌手 $\mathcal{A}$。敌手 $\mathcal{A}$ 收到挑战后，找到匹配的 F_{id} 和 Φ_{id}，执行算法 $\mathrm{prove}(\mathrm{chal},F_{\mathrm{id}},\Phi_{\mathrm{id}})$，生成相应的证据并返回。

4) 伪造阶段。敌手 $\mathcal{A}$ 以概率ε输出对应挑战 $\mathrm{chal}=(\mathrm{id}^*,\tau_{\mathrm{id}},\{i,v_i\}_{i\in I})$ 的一个与诚实证据不同的有效证据 $P^*=(\boldsymbol{x}^*,\boldsymbol{z}^*)$。

如果身份 id^* 不满足 $H_1(\mathrm{id}^*)=\boldsymbol{b}$，则终止。否则，算法$\mathcal{B}$查询表 L_t，找到匹配项$(\mathrm{id}^*,\tau_{\mathrm{id}^*},\Phi_{\mathrm{id}^*},F_{\mathrm{id}^*})$。假设第 i 块数据为 $\boldsymbol{z}_i$，相应的标签为 $\boldsymbol{x}_i$，则由上述构造过程和引理 2 可知，$\|\boldsymbol{x}_i\|\leqslant s'\sqrt{nm}$ 且 $\boldsymbol{a}_{\mathrm{id}}\boldsymbol{x}_i^{\mathrm{T}}=\alpha_i+\boldsymbol{z}_i\boldsymbol{u}^{\mathrm{T}}\ (\mathrm{mod}\ q)$ 以极大概率成立，其中 $\alpha_i=H_2(\tau_{\mathrm{id}^*},i)$。算法$\mathcal{B}$输出诚实证据 $P=(\boldsymbol{x},\boldsymbol{z})$，其中 $\boldsymbol{x}=\sum_{i\in I}v_i\boldsymbol{x}_i$，$\boldsymbol{z}=\sum_{i\in I}v_i\boldsymbol{z}_i$。

根据三角不等式可知，$\|\boldsymbol{x}\|\leqslant\ell(p-1)s'\sqrt{nm}$，其中，$\ell$ 是 I 中元素的总个数。同时，易知式(11)成立。

$$\boldsymbol{a}_{\mathrm{id}^*}\boldsymbol{x}^{\mathrm{T}}=\sum_{i\in I}v_i\alpha_i+\boldsymbol{z}\boldsymbol{u}^{\mathrm{T}}\ (\mathrm{mod}\ q)\tag{11}$$

由于证据 $P^*=(\boldsymbol{x}^*,\boldsymbol{z}^*)$ 也满足验证条件，即 $\|\boldsymbol{x}^*\|\leqslant\ell(p-1)s'\sqrt{nm}$，$\boldsymbol{z}^*\in\mathcal{R}_B^d$ 且

$$\boldsymbol{a}_{\mathrm{id}^*}\boldsymbol{x}^{*\mathrm{T}}=\sum_{i\in I}v_i\alpha_i+\boldsymbol{z}^*\boldsymbol{u}^{\mathrm{T}}\ (\mathrm{mod}\ q)\tag{12}$$

式(12)与式(11)相减，可得

$$\boldsymbol{a}_{\mathrm{id}^*}(\boldsymbol{x}^*-\boldsymbol{x})^{\mathrm{T}}=(\boldsymbol{z}^*-\boldsymbol{z})\boldsymbol{u}^{\mathrm{T}}\ (\mathrm{mod}\ q)\tag{13}$$

式(13)等价于

$$[\boldsymbol{a}_{\mathrm{id}^*}|\boldsymbol{u}]\left[(\boldsymbol{x}^*-\boldsymbol{x})|(\boldsymbol{z}-\boldsymbol{z}^*)\right]^{\mathrm{T}}=0\ (\mathrm{mod}\ q)\tag{14}$$

将 $\boldsymbol{a}_{\mathrm{id}^*}=[\boldsymbol{a}|\boldsymbol{b}]$ 和 $\bar{\boldsymbol{a}}=[\boldsymbol{a}|\boldsymbol{b}|\boldsymbol{u}]$ 代入式（14），并令 $\boldsymbol{y}=\left[(\boldsymbol{x}^*-\boldsymbol{x})|(\boldsymbol{z}-\boldsymbol{z}^*)\right]$，可得 $\bar{\boldsymbol{a}}\boldsymbol{y}^{\mathrm{T}}=0\ (\mathrm{mod}\ q)$。

由于以上两证据是不同的，所以 $\boldsymbol{y}\neq 0$。

又由于 $\boldsymbol{z}$ 和 $\boldsymbol{z}^*$ 均属于 $\mathcal{R}_B^d$，因此

$$\begin{aligned}&\|\boldsymbol{y}\|=\|(\boldsymbol{x}^*-\boldsymbol{x})|(\boldsymbol{z}-\boldsymbol{z}^*)\|\leqslant\\&\|\boldsymbol{x}^*-\boldsymbol{x}\|+\|\boldsymbol{z}-\boldsymbol{z}^*\|\leqslant\\&\ell(p-1)\sqrt{n}\left(2s'\sqrt{m}+(p-1)\sqrt{d}\right)=\beta\end{aligned}\tag{15}$$

即算法$\mathcal{B}$破解了 Ring-SIS 问题。

现在计算算法$\mathcal{B}$成功的概率。

算法$\mathcal{B}$成功意味着身份 id^* 在私钥提取阶段未被询问且 $H_1(\mathrm{id}^*)=\boldsymbol{b}$。由于身份 id^* 在私钥提取阶段未被询问的概率不低于 $\frac{Q_{\mathrm{H}}-Q_{\mathrm{E}}}{Q_{\mathrm{H}}}$，而身份 id^* 满足 $H_1(\mathrm{id}^*)=\boldsymbol{b}$ 的概率不低于 $\frac{1}{Q_{\mathrm{H}}}$，所以算法$\mathcal{B}$成功的概率不低于 $\frac{\varepsilon(Q_{\mathrm{H}}-Q_{\mathrm{E}})}{Q_{\mathrm{H}}^2}$。证毕。

6 性能评估

将本文方案分别与文献[18]和文献[9]中的方案进行实验比较来进行性能评估，其中，文献[18]中的 IBRDICL(identity based remote data integrity checking from lattices)方案是基于格的同类方案，而文献[9]中的 IBPDP(identity based provable data possession)方案是基于传统 CDH(computational Diffie-Hellman)假设的方案。这 3 种方案的主要特征如表 1 所示。

表 1 3 种云存储完整性检测方案的特征比较

方案	基于身份	随机预言模型	顽健性	困难假设
IBPDP 方案	是	是	是	CDH
IBRDICL 方案	是	是	否	SIS
本文方案	是	是	是	Ring-SIS

性能评估包括方案的存储开销，通信开销和计算开销 3 个方面。实验代码用 C++ 11 编写，通过 g++ 5.4.0 编译。实验由配置了 Intel Core i5-4590 处理器，8 GB 内存和 Ubuntu 16.04 LTS 操作系统的 PC 机运行。所有实验结果均是 10 次实验的平均值。

本文方案的实现使用了 NFLlib 函数库[26]，该函数库提供了多项式环上任意操作的快速实现；SampleD 算法利用了文献[27]任意模下格 $\Lambda^{\perp}(\boldsymbol{g})$ 的采样算法，该算法将文献[20]任意模下格 $\Lambda^{\perp}(\boldsymbol{g})$ 采样算法的计算开销从 $O(n\log^2 q)$ 降低到 $O(n\log q)$；散列函数使用 SHA-256 实现，并通过伪随机数生成器生成随机数进行填充。

根据 Ateniese 等[2]的结论，当文件的块损坏比例为 1%时，如果挑战请求包含的元素个数 $\ell=460$，则审计者能以不低于 99%的概率检测出损坏。因此，本节所有算法均选择 $\ell=460$。

6.1 对比同类方案

首先将本文方案与 IBRDICL 方案进行对比。IBRDICL 方案是一种一般格上基于身份的云存储完整性检测方案，该方案所采用的私钥提取算法和原像采样算法效率都较低，实验中将使用与本文方案相同的算法代替；此外，由于本文方案未涉及保护隐私，因此实验移除了 IBRDICL 方案中隐私保护所需的计算，主要包括证据生成阶段的一次原像采样运算和证据验证阶段的一次散列运算。

本实验中，2 种方案的安全参数 n 均设为 128，其他相关参数如表 2 所示。特别地，IBRDICL 方案中块标签向量的元素个数为 nm，单个数据块的元素个数为 nd，因此 2 种方案每个数据块的大小均为 2 KB。根据文献[28]可知，2 种方案在此参数下具有相近的安全级别。

表 2 实验参数设置（n=128 时）

n	q	p	k	m	d	σ
128	$\approx 2^{62}$	2	62	188	128	4.41

1) 存储开销

存储开销主要比较系统公钥长度、用户私钥长度和块标签长度。具体的存储开销比较如表 3 所示。

表 3 与 IBRDICL 方案的存储开销比较

方案	系统公钥/KB	用户私钥/KB	块标签/KB
IBRDICL 方案	31 496.00	203 112.00	67.56
本文方案	246.06	1 708.88	70.50

由表 3 可知，IBRDICL 方案的系统公钥长度和用户私钥长度都远大于本文方案，而块标签长度比本文方案略小。因此，本文方案的用户存储开销和审计者存储开销都更低，云服务器的存储开销稍高。

2) 通信开销

通信开销主要比较挑战长度和证据长度，具体的通信开销比较如表 4 所示。

表 4 与 IBRDICL 方案的通信开销比较

方案	挑战/KB	证据/KB
IBRDICL 方案	0.84	112.00
本文方案	0.84	114.94

由表 4 可知，本文方案与 IBRDICL 方案的通信开销相近，仅在证据长度上略大。

3) 计算开销

计算开销包括各算法的运算时间，其中标签生成算法分为离线阶段和在线阶段，离线阶段不依赖于数据，仅计算 SampleD 算法所需的扰动向量 $\boldsymbol{p}$ 及 $\boldsymbol{ap}^{\mathrm{T}}$，因此可以预先执行并将结果保存在本地。标签生成算法的计算开销比较如图 2 所示，其他算法的具体计算开销如表 5 所示。

图 2 中，文件数据块总数的范围是 1 000~10 000 块，每个数据块的大小为 2 KB。由图 2 可知，2 种方案离线阶段所需的计算开销都大于在线阶段的计算开销；相比较而言，本文方案离线阶段的计算开销与 IBRDICL 方案比略低；但本文方案在

表 5　　与 IBRDICL 方案的计算开销对比

方案	系统建立算法/ms	密钥提取算法/ms	审计算法/ms	证据生成算法/ms	证据验证算法/ms
IBRDICL 方案	171 893.66	553 469.70	0.02	11.32	215.16
本文方案	16.35	1816.45	0.02	49.04	2.57

线阶段的计算效率比 IBRDICL 方案提高了约 93.74%。

根据表 5，本文方案的系统建立算法和私钥提取算法的计算开销与 IBRDICL 方案相比都更小。虽然与 IBRDICL 方案相比，本文方案证据生成算法的计算效率较低，但证据验证算法的计算效率提高了 98.81%且总的绝对数值远小于 IBRDICL 方案。因此，本文方案降低了私钥生成中心、用户和审计者的计算开销。

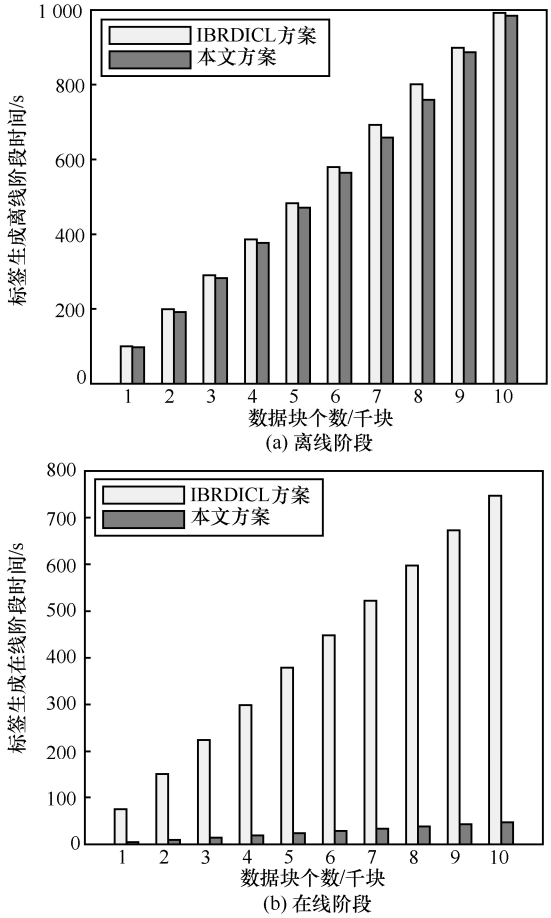

图 2　与 IBRDICL 方案标签生成算法计算开销对比

综上，相对于 IBRDICL 方案中的方案，本文方案降低了用户的存储开销，提高了系统建立、私钥提取、标签生成和证据验证等算法的计算效率，并且数据审计过程的总体计算开销也更低。

6.2　对比传统方案

将本文方案与 IBPDP 方案进行对比。IBPDP 方案是利用双线映射构造的一种基于身份的云存储完整性检测方案。本实验使用 PBC 库（版本为 0.5.14）实现 IBPDP 方案，其中实验参数选自 PBC 库的参数文件 a.param（安全级别约为 80 bit），扇区数设为 205 个（单个数据块长度约为 4 KB）。为了达到相近的安全级别和数据块大小，根据文献[28]，本文方案的安全参数 n 设为 512，其他相关参数如表 6 所示。

表 6　　实验参数设置（n=512 时）

n	q	p	k	m	d	σ
512	$\approx 2^{62}$	2	62	188	64	4.46

1) 存储开销

存储开销仍然比较系统公钥长度，用户私钥长度和块标签长度。具体的存储开销比较如表 7 所示。

表 7　　与 IBPDP 方案的存储开销比较

方案	系统参数/KB	用户私钥/KB	块标签/KB
IBPDP 方案	0.38	0.13	0.13
本文方案	736.25	7323.75	305.50

由表 7 可知，IBPDP 方案的存储开销与本文方案相比都更小。

2) 通信开销

通信开销也主要比较挑战长度和证据长度。具体的通信开销比较如表 8 所示。

表 8　　与 IBPDP 方案的通信开销比较

方案	挑战/KB	证据/KB
IBPDP 方案	9.77	30.38
本文方案	0.84	447.25

由表 8 可知，本文方案与 IBPDP 方案相比，仅挑战长度较小，总通信开销仍然较大。

3) 计算开销

本文方案与 IBPDP 方案在不同数据块情况下标签生成算法的计算开销如图 3 所示，其他算法的计算开销比较如表 9 所示。

表 9　　与 IBPDP 方案的计算开销对比

方案	系统建立算法/ms	私钥提取算法/ms	审计算法/ms	证据生成算法/ms	证据验证算法/ms
IBPDP 方案	3.04	4.03	117.18	561.66	2 038.46
本文方案	46.43	1 039.70	0.02	131.63	5.50

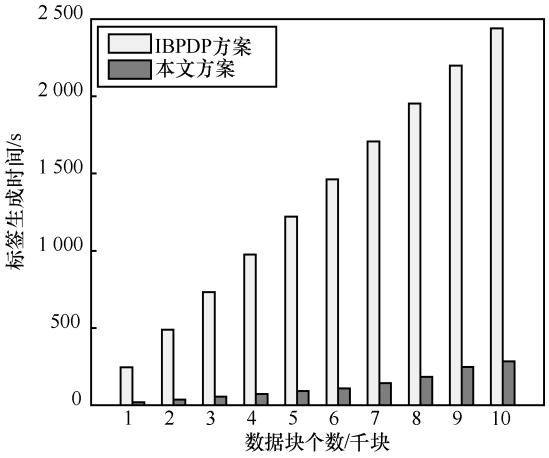

图 3　与 IBPDP 方案的标签生成算法计算开销对比

由于本文方案的标签生成算法分离线和在线两阶段，且离线阶段与数据无关，所以图 3 仅列出本文方案标签生成算法在线阶段的计算开销，并通过随机替换方法生成扰动向量 $\boldsymbol{p}$。

由图 3 可知，相比于 IBPDP 方案中，本文方案标签生成算法在线阶段的计算效率提高了约 88.32%。由表 9 可知，本文方案的系统建立算法和私钥提取算法的计算开销相对较大，但审计算法、证据生成和验证算法的计算效率与 IBPDP 方案相比分别提高了 99.98%、76.56%和 99.73%。由于这些算法执行较为频繁，因此本文方案大大降低了用户、云服务器和审计者的计算开销。

综上，相比于 IBPDP 方案，本文方案提高了用户、云服务器和审计者的计算效率。

7　结束语

本文基于理想格上的困难问题构造了一种基于身份的云存储完整性检测方案。在随机预言模型下该方案可以抵抗云服务器的适应性选择身份攻击。实验结果表明：与同类方案相比，本文方案整体更优，更适合实际应用；与传统方案相比，本文方案数据审计过程的计算开销也更低。

参考文献：

[1] BABCOCK C. 9 worst cloud security threats[N]. 2014. http://www.informationweek.com/cloud/infrastructure-as-a-service/9-worst-cloud-security-threats/d/d-id/1114085.

[2] ATENIESE G, BURNS R, CURTMOLA R, et al. Provable data possession at untrusted stores[C]//ACM Conference on Computer and communications security. ACM, 2007: 598-609.

[3] JUELS A, KALISKI JR B S. Pors: proofs of retrievability for large files[C]//ACM Conference on Computer and Communications Security. ACM, 2007:584-597.

[4] SHACHAM H, WATERS B. Compact proofs of retrievability[C]// International Conference on the Theory and Application of Cryptology and Information Security. Springer, 2008: 90-107.

[5] WANG Q, WANG C, REN K, et al. Enabling public auditability and data dynamics for storage security in cloud computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2011, 22(5): 847-859.

[6] WANG C, CHOW S S M, WANG Q, et al. Privacy-preserving public auditing for secure cloud storage[J]. IEEE Transactions on Computers, 2013, 62(2): 362-375.

[7] XU Y, DING R, CUI J, et al. Intrusion-resilient public auditing protocol for data storage in cloud computing[C]//Australasian Conference on Information Security and Privacy. Springer, 2018: 399-416.

[8] WANG H, WU Q, QIN B, et al. Identity-based remote data possession checking in public clouds[J]. IET Information Security, 2014, 8(2):114-121.

[9] YU Y, ZHANG Y, MU Y, et al. Provably secure identity based provable data possession[C]//International Conference on Provable Security. Springer, 2015:310-325.

[10] TIAN M, YE S, ZHONG H, et al. Identity-based proofs of storage with enhanced privacy[C]//International Conference on Algorithms and Architectures for Parallel Processing. Springer, 2018: 461-480.

[11] SHAMIR A. Identity-based cryptosystems and signature schemes[C]//Annual International Cryptology Conference. Springer, 1984:47–53.

[12] SHOR P W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer[J]. SIAM Journal on Computing, 1997, 26(5): 1484-1509.

[13] REGEV O. Lattice-based cryptography[C]//Annual International Cryptology Conference. Springer, 2006: 131-141.

[14] XU W, FENG D, LIU J. Public verifiable proof of storage protocol from lattice assumption[C]//International Conference on Intelligent Control, Automatic Detection and High-End Equipment. IEEE, 2012:133-137.

[15] MICCIANCIO D, REGEV O. Worst-case to average-case reductions based on Gaussian measures[C]//Symposium on Foundations of

Computer Science. IEEE, 2004:372-381.

[16] LIU H, CAO W. Public proof of cloud storage from lattice assumption[J]. Chinese Journal of Electronics, 2014, 23(1):186-190.

[17] ZHANG X, XU C, ZHANG Y, et al. Insecurity of a public proof of cloud storage from lattice assumption[J]. Chinese Journal of Electronics, 2017, 26(1):88-92.

[18] LIU Z, LIAO Y, YANG X, et al. Identity-based remote data integrity checking of cloud storage from lattices[C]//International Conference on Big Data Computing and Communications. IEEE, 2017:128–135.

[19] GORBUNOV S, VAIKUNTANATHAN V, WICHS D. Leveled fully homomorphic signatures from standard lattices[C]//Annual ACM Symposium on Theory of Computing. ACM, 2015:469-477.

[20] MICCIANCIO D, PEIKERT C. Trapdoors for lattices: simpler, tighter, faster, smaller[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 2012:700-718.

[21] DUCAS L, MICCIANCIO D. Improved short lattice signatures in the standard model[C]//Annual International Cryptology Conference. Springer, 2014:335- 352.

[22] LYUBASHEVSKY V, MICCIANCIO D. Generalized compact knapsacks are collision resistant[C]//International Colloquium on Automata, Languages, and Programming. Springer, 2006:144-155.

[23] PEIKERT C, ROSEN A. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices[C]//Theory of Cryptography Conference. Springer, 2006:145-166.

[24] REGEV O. New lattice-based cryptographic constructions[J]. Journal of the ACM, 2004, 51(6): 899-942.

[25] GENTRY C, PEIKERT C, VAIKUNTANATHAN V. Trapdoors for hard lattices and new cryptographic constructions[C]//Annual ACM Symposium on Theory of Computing. ACM, 2008: 197-206.

[26] AGUILAR-MELCHOR C, BARRIER J, GUELTON S, et al. NFLlib: NTT-based fast lattice library[C]//Cryptographers' Track at the RSA Conference. Springer, 2016: 341-356.

[27] GENISE N, MICCIANCIO D. Faster gaussian sampling for trapdoor lattices with arbitrary modulus[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 2018: 174-203.

[28] RUCKERT M, SCHNEIDER M. Estimating the Security of Lattice-based Cryptosystems[J]. IACR Cryptology ePrint Archive, 2010, 20101006: 091355.

[作者简介]

田苗苗（1987– ），男，安徽阜阳人，博士，安徽大学副教授，主要研究方向为密码学与信息安全。

高闯（1994– ），男，河南长垣人，安徽大学硕士生，主要研究方向为密码学与信息安全。

陈洁（1985– ），男，江苏苏州人，博士，华东师范大学研究员，主要研究方向为密码学与信息安全。