基于格的高效通用累加器与被累加值的零知识证明

谭子欣[1,2,3], 邓 燚[1,2,3], 马 丽[1,3]

[1] 中国科学院信息工程研究所 信息安全国家重点实验室 北京 中国 100093
[2] 密码科学技术国家重点实验室 北京 中国 100878
[3] 中国科学院大学 网络空间安全学院 北京 中国 101408

摘要 通用累加器作为一种具有数据压缩性质的重要密码学元件, 其多应用于隐私保护相关的区块链系统、身份认证系统以及各类权限管理系统。研究发现目前已有的基于小整数解(SIS)问题困难性假设的通用累加器内部计算效率不高, 且更新效率低。因此, 本文设计并实现了首个基于环小整数解(Ring-SIS)问题困难性假设的高效通用累加器, 其更新开销在平均意义上远低于以往方案, 更加适用于更新操作频繁, 成员数量更庞大的应用场景。另外针对 Ring-SIS 通用累加器内的所有成员, 本文基于 Schnorr-like 协议框架提出了首个单轮次执行合理性错误可忽略的被累加值的零知识证明协议。

关键词 通用累加器; 知识的零知识证明
中图法分类号 TP309.7 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2021.07.06

Lattice-Based Efficient Universal Accumulator and Zero-Knowledge Proofs of an Accumulated Value

Tan Zixin[1,2,3], Deng Yi[1,2,3], Ma Li[1,3]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2] State Key Laboratory of Cryptology , Beijing 100878, China
[3] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 101408, China

Abstract As an important cryptographic primitive with data compression property, the universal accumulator is mostly used in block chain system, identity authentication system and various privilege management system related to privacy protection. It is found that the existing universal accumulator based on the assumption of the difficulty of solving small integer solution problem(SIS) is inefficient to compute and update. So this paper designs and implements the first universal accumulator based on the hypothesis that there is a difficulty in solving ring small integer solution(Ring-SIS) problem to realize a more efficient universal accumulator, whose update overhead is much lower on average than previous schemes, and it is more suitable for application scenarios where update operations are frequent and the member size is larger. In addition, aiming at all of the members of the Ring-SIS universal accumulator, this paper proposes the first protocol of zero-knowledge proofs of an accumulated value, which is based on Schnorr-like framework and has negligible soundness error in a single round.

Key words universal accumulator; zero-knowledge proofs

1 引言

在过去, 当用户需要访问数据, 通常需要向数据管理者发送访问请求, 然后管理者通过查表检查该用户是否具有访问权限。在这样的权限管理过程中, 查表操作带来的开销往往会随着列表的大小呈线性增长, 这样低效的管理方法显然将无法适应未来高速的大数据时代, 为了解决这样的问题, 1994 年 Benaloh 和 de Mare[1]提出了累加器的概念。累加器是指将某个集合中的所有元素压缩成一个较短输出, 并能够为所有被累加值生成其对应的成员关系证据, 通过成员关系证据可以向他人证明被累加值的成员身份, 故而用户可直接将其身份和成员关系证据发送给数据管理者, 数据管理者再通过一个确定的检验算法来判定该用户的合法性, 这样的过程大大缩减了权限管理中的验

通讯作者: 邓燚, 博士, 研究员, Email: deng@iie.ac.cn。
本课题得到国家自然科学基金项目(No.61772521); 中科院前沿科学重点研究项目, CAS(No.QYZDB-SSW-SYS035); 密码科学技术国家重点实验室开放项目资助。
收稿日期: 2019-07-24; 修改日期: 2019-09-12; 定稿日期: 2021-06-24

证时间。除此之外, 累加器在数字签名、匿名凭证、范围证明、集合成员关系证明等领域也有相当多的应用场景。

近二十年来, 累加器的发展日新月异, 功能性及安全性也在不断的更新和完善。1997 年 Barić 和 Pfitzmann[2]提出了无碰撞累加器的概念, 并且首次给出了累加器的安全性定义。2002 年 Camenisch 和 Lysyanskaya[3]提出了动态累加器的概念, 即累加器增删元素的更新操作的时间复杂度独立于集合大小。2005 年 Nguyen[4]提出了基于强 Diffie-Hellman 假设的动态累加器, 2008 年 Damgård[5]和 Triandopoulos 又在此基础上设计出了能够支持非成员关系证明的双线性映射累加器。2007 年 Jiangtao Li 等人[6]提出了通用累加器的概念, 同时基于强 RSA 假设实现了动态通用累加器。通用累加器既能为被累加成员生成成员关系证据, 又能为非被累加成员生成非成员关系证据, 而动态通用累加器则在通用累加器的基础上增加了动态更新的功能。到了 2012 年 Camacho 等人[7]和 Lipmaa[8]分别独立研究出了能够去除可信第三方前提假设限制的通用累加器, 前者是基于哈希树结构的通用累加器, 后者是建立在欧氏环上的通用累加器。

随着量子计算的发展, 人们开始忧心以往那些安全性基于强 RSA 假设, 离散对数假设等数论假设的累加器方案, 未来在超高速量子计算机面前将会不堪一击, 因而开始向格领域探索。2016 年 Libert 等人[9]在利用基于 SIS 问题困难性假设[10]的抗碰撞哈希函数, 构造出了第一个抗量子攻击的 SIS 累加器, 随后 Ling 等人[11]在此基础上又提出了抗量子攻击的动态 SIS 累加器。2018 年 Libert 等人[12]又设计出了基于 SIS 问题困难性假设的通用累加器, 同年 Zuoxia Yu 等人[13]设计出了首个基于 SIS 问题困难性假设的非成员关系累加器。

基于格上困难性假设的零知识证明系统通常有基于 Stern-like 框架[9,11-15]和基于 Schnorr-like 框架[16-20]两种模式。基于 Stern-like 框架的协议结合统计隐藏, 计算绑定的承诺方案可以实现具有完美完整性以及统计零知识性的零知识协议, 但缺点在于其单轮协议执行具有 2/3 的合理性错误, 为了降低合理性错误, 需要多次执行协议, 所以整体执行效率低, 在效率上不适用于现实场景; 而基于 Schnorr-like 协议框架相对前者执行效率较高, 且合理性错误较低, 甚至可以实现单轮次执行合理性错误可忽略的零知识协议[1819], 但需要拒绝样保证协议零知识性, 因此存在一定的协议终止概率, 不过可以通过参数设置将该终止概率降至任意小, 所以相对前者更具有现实应用价值。

被累加值的零知识证明是指证明者以零知识的方式向他人证明自己知道一个累加器中的被累加值, 即其所知秘密满足通用累加器成员关系验证算法, 属于累加器在零知识证明领域上的一种应用。在过去, 基于格上困难性假设的被累加值的零知识证明[9]和非被累加值的零知识证明[16]都主要采用 Stern-like 协议框架来实现, 却无法基于 Schnorr-like 协议框架来实现, 主要原因在于该框架存在着知识的提取性错误, 即无法保证提取出的证据满足范数限制要求或范围限制要求。今年 Bootle 等人[19]解决了这一问题, 并基于 Schnorr- like 协议框架实现了短秘密 $\bar{s}$ 满足 $\mathrm{A}\vec{s}=\vec{u} \bmod q$ 的零知识证明, 同时其 $1/n$ 的合理性错误远小于基于 Stern-like 协议框架实现的版本。同年, Rupeng Yang 等人[20]基于 Schnorr-like 的协议框架实现了合理性错误 1/poly 的通用零知识协议框架, 并且针对 SIS 累加器[9], 提出了相应的被累加值的零知识证明方案[20], 合理性错误可降低至 1/poly 。

1.1 本文贡献

本文在 SIS 通用累加器[14]的基础上通过替换底层假设以及改进通用累加器的存储结构等手段, 设计并实现了第一个基于 Ring-SIS 问题困难性假设的通用累加器, 可为被累加成员生成较短成员关系证据, 也可为非被累加成员生成较短非成员关系证据; 该通用累加器打破了原有格上通用累加器[14]和非成员关系累加器[15]只能全局更新的限制, 可实现局部更新; 图 1 展示的是本文设计的 Ring-SIS 哈希函数同 SIS 哈希函数[15]在计算效率上的比较, 当安全参数 N 越大, Ring-SIS 哈希函数的计算效率越高。图 2 选用参数 a_1=64, N=171, 此时 Ring-SIS 哈希函数在计算效率上会低于 SIS 哈希函数, 但随着集合大小 M 的增加, Ring-SIS 通用累加器存储结构带来的效率优势也越发明显, 所以综合图 1 和图 2 可以得出结论: 随着安全参数 N 的增大或者被累加集合大小 M 的增大, Ring-SIS 通用累加器内部计算效率以及更新效率要远优于以往的 SIS 通用累加器[15]。实验结果表明本文所设计的通用累加器将更加适用于更新操作频繁且数据量更大的应用场景。

另外针对本文所设计的 Ring-SIS 通用累加器, 本文基于 Schnorr-like 协议框架, 并选用尺寸大于

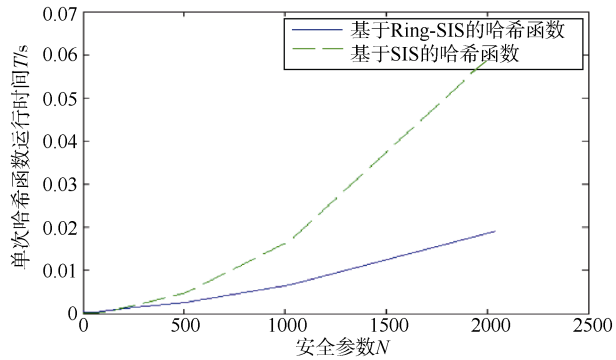


图 1　当 $q=1024$, Ring-SIS 哈希函数与 SIS 的哈希函数运行时间对比图

Figure 1　The comparison of execution time between the Ring-SIS hash function and the SIS hash function when $q=1024$

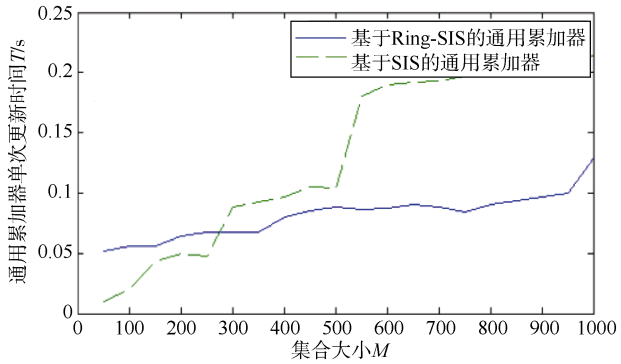


图 2　当 q=64, N=171, Ring-SIS 通用累加器与 SIS 通用累加器单次更新运行时间对比图

Figure 2　The comparison of single update time between the Ring-SIS accumulator and the SIS accumulator when q=64, N=171

2^{256} 的挑战值空间, 设计出了第一个合理性错误可忽略的被累加值的零知识证明协议。表 1 所展示的是本文所设计零知识协议同以往协议[9,20]的工作比较。

1.2　本文结构

本文将在第 2 章节给出全文所需要的准备工作, 其中包括符号定义, 密码学工具定义以及相关安全性定义; 第 3 章介绍了基于 Ring-SIS 问题困难性假设的通用累加器的具体设计以及通用累加器的安全性证明; 第 4 章详细介绍了 Ring-SIS 通用累加器所相应的被累加值的零知识证明协议, 包括设计思路、具体构造以及协议的安全性证明; 第 5 章对本文工作进行了简要总结。

2　预备知识

2.1　符号定义

全文所需要用到的符号定义如表 1 所示。

2.2　工具介绍

2.2.1　格上问题与困难性假设

定义 1(Ring-SIS$^{p}_{q,l,\beta}$ 困难性假设)[21]　已知环 R, 整数模数 $q\geqslant 2$, 整数 $l\geqslant 1$, $\beta>0$ 和 $p\geqslant 1$。给定多项式向量 $\boldsymbol{a}=(a_1,\cdots,a_l)\xleftarrow{\$}R_q^l$, 对于任意概率多项式时间 (PPT) 算法求取一个非零短向量 $\boldsymbol{e}=(e_1,\cdots,e_l)\xleftarrow{\$}R^l$, 满足 $\sum_{i=1}^{l}a_i\cdot e_i=0 \bmod qR$ 且 $\|e\|_p\leqslant\beta$ 是困难的。

定义 2(SKS$^{p}_{n,k,\beta}$ 问题)[18]　已知环 R, 整数模数 $q\geqslant 2$, 整数 $k>n\geqslant 1$, $\beta>0$ 和 $p\geqslant 1$。给定一个随机矩阵 $\boldsymbol{E}\xleftarrow{\$}R_q^{n\times(k-n)}$, 找到一个非零短向量 $\boldsymbol{y}$ 满足 $[\mathrm{I}_n \quad \boldsymbol{E}]\cdot\boldsymbol{y}=0^n$ 且 $\|\boldsymbol{y}\|_p\leqslant\beta$。

根据[18]中的定义, 如果存在算法 $\mathcal{A}$ 和函数 ε, 对无穷多个 n, 使得

$$\left|\Pr\left[\begin{bmatrix}[\mathrm{I}_n \quad \boldsymbol{E}]\cdot\boldsymbol{y}=0^n\\ \wedge\|\boldsymbol{y}\|_p\leqslant\beta\end{bmatrix}\middle|\begin{matrix}\boldsymbol{E}\xleftarrow{\$}\mathbb{R}_q^{n\times(k-n)};\\ 0\neq\boldsymbol{y}\leftarrow\mathcal{A}(\boldsymbol{E})\end{matrix}\right]\right|\geqslant\varepsilon(n),$$

则称算法 $\mathcal{A}$ 能以优势 $\varepsilon(n)$ 解决 SKS$^{p}_{n,k,\beta}$ 问题。

定义 3(DKS$^{p}_{n,k,\beta}$ 问题)[18]　已知环 R, 整数模数 $q\geqslant 2$, 整数 $k>n\geqslant 1$, $\beta>0$ 和 $p\geqslant 1$。区分 $[\mathrm{I}_n\|\mathrm{A}]\cdot\boldsymbol{y}$ 的分布与 R_q^n 上均匀随机分布, 其中 $\mathrm{A}\xleftarrow{\$}R_q^{n\times(k-n)}$, 且 $\boldsymbol{y}\xleftarrow{\$}\mathcal{S}^{k}_{p,\beta}$ 为非零短向量。

表 1　现有工作对比

Table 1　The comparison of current schemes

方案	合理性错误	承诺方案	协议类型	累加器类型
[9]	$\frac{2}{3}$	[22]	Stern-like	SIS 累加器[9]
[20]	$\frac{1}{\mathrm{poly}}$	[18]	Schnorr-like	SIS 累加器[9]
本文方法	可忽略	[18]	Schnorr-like	Ring-SIS 通用累加器

表 2 标记符注释列表

Table 2 Tag comment list

符号	注释	符号	注释
$\mathbb{R}$	实数域	$\times$ 或 $\cdot$	普通乘法
$\mathbb{Z}$	整数域	$\mathbb{Z}_q$	对于 $\forall z \in \mathbb{Z}_q$， $z \in \{0,\cdots,q-1\}$
R	$\mathbb{Z}[x]/(x^N+1)$	R_q	$\mathbb{Z}_q[x]/(x^N+1)$
$\mathbb{N}$	自然数域	$\bmod qR$	模上 q 和 x^N+1
ε	可忽略函数	$\|$	连接符，$\{0,1\}^n \| \{0,1\}^m \to \{0,1\}^{n+m}$
$\mathcal{C}$	$\mathcal{C}=\{c \in R_q \mid \|c\|_\infty=1, \|c\|_1=60\}$	$[n]$	整数集合 $\{1,\cdots,n\}$
$\mathcal{S}_{p,b}$	$\mathcal{S}_{R,\beta}=\{s \in R_q \mid \|s\|_p=\beta\}$	$\mathcal{N}_{v,s}$	期望为 v，标准差为 σ 的正态分布
$\boldsymbol{a}[i,:]$ $\boldsymbol{a}[:,i]$	$\boldsymbol{a}$ 为一个矩阵，$\boldsymbol{a}[i,:]$ 表示矩阵 $\boldsymbol{a}$ 的第 i 行; $\boldsymbol{a}[:,i]$ 表示矩阵 $\boldsymbol{a}$ 的第 i 列	*	$\forall b \in R^L$，$\forall f \in R^{(L\cdot l)\times N}$，$y=b*f \in R^{(L\cdot l)\times N}$，其中对 $\forall i \in [L]$， $y[(i-1)\cdot l+1:i\cdot l,\ :]=b[i]\cdot f[(i-1)\cdot l+1:i\cdot l,\ :]$
mtc$(\cdot)$	单个矩阵转换为单个列向量，对 $\forall z \in R^{g\times N}$， $t=\text{mtc}(z) \in R^{g\cdot N\times 1}$，$t[(i-1)\cdot N+1:i\cdot N,1]=z[i,\ :]$， 其中 $\forall i \in [g]$	mtc$_k(\cdot)$	k 个矩阵转换为 1 个列向量，$\forall z_i \in R^{g_i\times N_i}$， $t=\text{mtc}_k(z_1,\cdots,z_k) \in R^{\sum_{i\in[k]} g_i\cdot N_i\times 1}$，其中 $\forall j \in [k]$， $t[(j-1)\cdot g_j\cdot N_j+1:j\cdot g_j\cdot N_j,1]=\text{mtc}(z_j)$
ptb$(\cdot)$	多项式转换为二进制串，令 $R'=R_2/\{x^N\}$， 对 $\forall t=\sum_{i\in[N]} t_i\cdot x^{i-1} \in R'$， $\text{ptb}(t)=(t_N,\ \cdots,\ t_1) \in \{0,1\}^N$	ptb$_k(\cdot)$	k 个多项式转换为二进制串，令 $R'=R_2/\{x^N\}$，对 $\forall \delta \in R'^k$， $\text{ptb}_k(\delta)=(\text{ptb}(\delta_k)\|\cdots\|\ \text{ptb}(\delta_1)) \in \{0,1\}^{k\cdot N}$
btI$(\cdot)$	二进制串转换为整数，对 $\forall b \in \{0,1\}^N$， $\text{Int}(b)=\sum_{i\in[N]} 2^{i-1}\cdot b_i \in \mathbb{Z}$	Itb$_k(\cdot)$	整数转换为 k 长二进制串，对 $\forall t \in \mathbb{Z}$， $\text{Itb}_k(t)=(t_k,\cdots,t_1) \in \{0,1\}^k$，其中 $t=\sum_{i\in[N]} 2^{i-1}\cdot t_i$
$\|\cdot\|_p$	p 范数。对任意 $t \in R$，$\|t\|_p=\sqrt[p]{\sum_{i\in[N]} \|t_i\|^p}$; 当 $p=\infty$，$\|t\|_\infty=\text{Max}(\|t_i\|_{i\in N})$	$\circ$	$\forall s,r \in R^N$，$t=s\circ r$，其中 $t[i]=s[i]\cdot r[i]$，对 $\forall i \in [N]$

根据文献[18]在定义，如果存在算法 $\mathcal{A}$ 和函数 ε，对于无穷多个 n，使得

$$\left|\Pr\left[b=1 \middle| \begin{matrix}\mathrm{A} \xleftarrow{\$} R_q^{n\times(k-n)}; y \xleftarrow{\$} \mathcal{S}_{p,\beta}^k; \\ b \leftarrow \mathcal{A}(\mathrm{A}, [\mathrm{I}_n \| \mathrm{A}]\cdot y)\end{matrix}\right] - \Pr\left[b=1 \middle| \begin{matrix}\mathrm{A} \xleftarrow{\$} R_q^{n\times(k-n)}; u \xleftarrow{\$} R_q^n; \\ b \leftarrow \mathcal{A}(\mathrm{A}, u)\end{matrix}\right]\right| \geqslant \varepsilon(n),$$

则称算法 $\mathcal{A}$ 能以优势 $1/\text{poly}(n)$ 解决 $\text{DKS}_{n,k,\beta}^p$ 问题。

2.2.2 抗碰撞哈希函数

定义 4 已知环 R，整数模数 $q \geqslant 2$，整数 $n,m \geq 0$。如果函数 $\mathrm{H}: R_q^n \to R_q^m$ 满足下面三条性质:

易于计算: 对于任意 $x \in R_q^n$，$\mathrm{H}(x)$ 能在多项式时间内计算完成;

压缩性: $m<n$;

抗碰撞性: 对于任意 PPT 算法 $\mathcal{A}$，都存在一个可忽略函数 ε，对所有安全参数 $n \in \mathbb{N}$，

$$\Pr\left[(x_0,x_1) \leftarrow \mathcal{A}(1^n,h): x_0 \neq x_1 \wedge \mathrm{H}(x_0)=\mathrm{H}(x_1)=h\right] < \varepsilon(n),$$

则称函数 H 为抗碰撞哈希函数。

2.2.3 通用累加器

通用累加器主要由以下 4 个算法组成:

启动算法(Setup): 输入安全参数 N，输出一个公共参数 pp;

累加器生成算法(Acc): 输入公共参数 pp 和集合 S，输出累加值 u;

证据生成算法(Wit): 输入公共参数 pp、累加值 u、元素 δ 以及类型 $type$。当 $type=0$，输出 $\delta \in S$ 的证据 w，否则输出 $\delta \notin S$ 的证据 w;

验证算法(Ver): 输入公共参数 pp、累加值 u、元素 δ、证据 w 以及类型 $type$。当 $type=0$，验证 w 是否为 $\delta \in S$ 的证据，否则验证 w 是否为 $\delta \notin S$ 的证据。

如果通用累加器对于所有的 PPT 敌手 $\mathcal{A}$，都存在一个可忽略函数 ε，满足:

$$\Pr\left[\begin{array}{l}\mathrm{pp}\leftarrow\mathrm{Setup}(N);(S,\delta,w_1,w_2,type)\leftarrow\mathcal{A}(\mathrm{pp}):\\ (\delta\in S\wedge\mathrm{Ver}_{\mathrm{pp}}(\mathrm{Acc}(\mathrm{pp},S),\delta,w_1,type=1)=1)\\ \vee(\delta\notin S\wedge\mathrm{Ver}_{\mathrm{pp}}(\mathrm{Acc}(\mathrm{pp},S),\delta,w_2,type=0)=1)\end{array}\right]<\varepsilon(N),$$

则称该通用累加器是安全的。

2.2.4 知识的零知识证明

定义 5(知识的零知识证明系统) 对于 NP 关系 $P=\{(x, w)\}\in\{0,1\}^*\times\{0,1\}^*$，假设 $\mathcal{P}$ 为证明者，$\mathcal{V}$ 为验证者，如果 $\langle\mathcal{P}(x,w),\mathcal{V}(x)\rangle$ 是知识的零知识证明系统，则满足下面性质:

完整性: 如果 $(x, w)\in P$，则

$$\Pr[\langle\mathcal{P}(x,w),\mathcal{V}(x)\rangle=1]>1-error_c,$$

其中 $error_c$ 为完整性错误。

m-特殊合理性: 如果存在 PPT 提取器，以 $\langle\mathcal{P}^*,\mathcal{V}(x)\rangle$ 产生的 m 条合法副本 (a,e_1,z_1)，…，(a,e_m,z_m) 作为输入，可以输出证据 w' 满足 $(x,w')\in P$，其中对于任意 $i,j\in[m]$，$e_i\neq e_j$。

诚实验证者零知识性: 存在一个 PPT 模拟器 **S**，使得 $\mathbf{S}(x,r)$ 输出副本 $tr'=(a', e', z')$ 的分布与验证者 $\mathcal{V}$ 诚实执行协议 $\langle\mathcal{P}(x,w),\mathcal{V}(x)\rangle$ 所输出副本的分布是计算不可区分，其中 r 为诚实验证者使用的随机数。

2.2.5 正态分布与拒绝抽样

在域 $\mathbb{R}^N$ 上的，期望值为 $v\in\mathbb{R}^N$ 同时标准差为 $\sigma>0$ 的正态分布，其概率密度函数通常被定义为

$$\rho_{v,\sigma}^N(x)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(\frac{-\|x-v\|_2^2}{2\sigma^2}\right).$$

本文所使用的是在域 R^k 上，期望值为 $v\in R^k$，标准差为 $\sigma>0$ 的离散正态分布，其分布函数被定义为

$$\mathcal{N}_{v,\sigma}^{k\cdot N}(x)=\frac{\rho_{v,\sigma}^{k\cdot N}(x)}{\sum_{y\in R^k}\rho_\sigma^{k\cdot N}(y)},$$

其中 $\rho_\sigma^{k\cdot N}(x)=\rho_{0,\sigma}^{k\cdot N}(x)$。

下面是关于正态分布随机抽样的范数界限定理和拒绝抽样定理。

定理 1[18,23] 对于任意 $z\xleftarrow{\$}\mathcal{N}_\sigma^k$，$k>0$

$$\Pr(\|z\|_2\leqslant 2\sigma\sqrt{k\cdot N})\geqslant 1-2^{-\frac{k\cdot N}{2}}.$$

当 $k\cdot N$ 足够大时，该范数界限成立的概率接近于 1。

定理 2[23] 集合 V 为 R^m 中所有 $\|\cdot\|_2<T$ 的元素构成的子集，$\sigma=\omega(T\sqrt{\log m})\in\mathbb{R}$，同时 $h:V\to\mathbb{R}$ 为一种概率分布。存在常数 $M=\mathrm{O}(1)$，使得算法 $\mathcal{A}$ 与算法 $\mathcal{F}$ 输出分布的统计距离将小于 $\frac{2^{-\omega(\log m)}}{M}$。

算法 $\mathcal{A}$:

(1) $v\xleftarrow{\$}h$;

(2) $z\xleftarrow{\$}\mathcal{N}_{v,\sigma}^m$;

(3) 以概率 $\min(1,\frac{\mathcal{N}_\sigma^m(z)}{\mathrm{M}\cdot\mathcal{N}_{v,\sigma}^m(z)})$ 输出 (z, v)。

算法 $\mathcal{F}$:

(1) $v\xleftarrow{\$}h$;

(2) $z\xleftarrow{\$}\mathcal{N}_\sigma^m$;

(3) 以概率 $\frac{1}{M}$ 输出 (z, v)。

由文献[23]可知，如果对于任意正整数 α，令 $\sigma=\alpha\cdot T$，则 $M=\exp(12/\alpha+1/(2\alpha^2))$。算法 $\mathcal{A}$ 将会以至少 $\frac{1-2^{-100}}{M}$ 的概率正常输出，且其输出分布与算法 $\mathcal{F}$ 输出分布之间的统计距离将小于 $\frac{2^{-100}}{M}$。

2.2.6 挑战值空间

挑战值空间是指验证者在收到来自证明者的第一轮消息后，选择挑战值的值域空间，通常挑战值空间的尺寸需要足够大，才能为了保证零知识证明协议较低的合理性错误，所以本文选择挑战集合 $\mathcal{C}=\{c\in R_q\mid\|c\|_\infty=1,\|c\|_1=60\}$，当 $N=512$，其大小满足 $|\mathcal{C}|=C_{512}^{60}>2^{256}$。

定理 3[18,24] 当 N，d 为 2 的幂次，且满足 $N\geqslant d>1$，模质数满足 $q\equiv 2d+1\bmod 4d$，则

(1) 多项式 x^N+1 可以分解为 d 个不可约多项式: $x^{N/d}-r_j \bmod q$，其中 $r_j\in\mathbb{Z}_q^*$;

(2) 对于所有 $y\in R_q\setminus\{0\}$ 且满足 $\|y\|_\infty<\frac{1}{\sqrt{d}}\cdot q^{1/d}$ 或 $\|y\|_2<q^{1/d}$，在域 R_q 中可逆。

在集合 $\overline{\mathcal{C}}=\{2c''-c-c'\mid c\neq c'\neq c''\in\mathcal{C}\}$ 中，对于任意元素 $\overline{c}\in\overline{\mathcal{C}}$ 都满足 $\overline{c}\neq 0\wedge\|\overline{c}\|_\infty=4$，所以当质数 $q=2d+1+4d\cdot k$，k 为正整数，同时 d 与 k 满足 $4^d\cdot d^{d/2}<2d+1+4d\cdot k$，则根据定理 3，元素

$\overline{c} \in \overline{\mathcal{C}}$ 在域 R_q 中可逆。

2.2.7 承诺方案

承诺方案由承诺阶段 Com 和打开阶段 Open 组成。在承诺阶段, 承诺者 $\mathcal{S}$ 和接收者 $\mathcal{R}$ 需要先执行一个交互协议或者非交互协议, 为一个具体的承诺方案生成所需参数, 如消息 x。然后 $\mathcal{S}$ 从自身的随机带中是随机挑选一个随机数 r, 计算承诺值 $c=\mathrm{Com}(x;r)$, 并将 c 发送给接收者; 在打开阶段, $\mathcal{S}$ 将被承诺的消息 x 和随机数 r 发送给 $\mathcal{R}$, $\mathcal{R}$ 随即通过验证 $c=\mathrm{Com}(x;r)$ 来判断 x, r 是否为 c 的被承诺值。

若承诺方案 $\langle \mathcal{S}, \mathcal{R}^* \rangle$ 是 ε-隐藏的, 则对于任意算法 $\mathcal{R}^*$, 使得

$$\Pr\left[b=b' \left| \begin{array}{l} \mathrm{pp} \leftarrow \mathrm{Setup}(1^n); \\ (x_0, x_1, r_0, r_1) \leftarrow \mathcal{R}^*(\mathrm{pp}); \\ b \xleftarrow{\$} \{0,1\}; c=\mathrm{Com}(x_b; r_b); \\ b' \leftarrow \mathcal{R}^*(\mathrm{pp}, c) \end{array} \right. \right] < \varepsilon(n);$$

另外, 当算法 $\mathcal{R}^*$ 被限制为多项式时间算法, 则上述承诺方案是计算隐藏的, 当不限制算法 $\mathcal{R}^*$ 的计算能力, 则称承诺方案是统计隐藏的。

若承诺方案 $\langle \mathcal{S}^*, \mathcal{R} \rangle$ 是 ε-绑定的, 则对任意算法 $\mathcal{S}^*$, 有

$$\Pr\left[\begin{array}{l} c=\mathrm{Com}(x_0; r_0) \\ \wedge c=\mathrm{Com}(x_1; r_1) \\ \wedge\ x_0 \neq x_1 \end{array} \left| \begin{array}{l} \mathrm{pp} \leftarrow \mathrm{Setup}(1^n); \\ (x_0, x_1, r_0, r_1, c) \leftarrow \mathcal{S}^*(\mathrm{pp}) \end{array} \right. \right] < \varepsilon(n).$$

另外, 当算法 $\mathcal{S}^*$ 被限制为多项式时间算法, 则上述承诺方案是计算绑定的, 当不限制算法 $\mathcal{S}^*$ 的计算能力, 则称承诺方案是统计绑定的。

本文所构造的知识的零知识证明方案, 需要使用具有加法同态性质的承诺方案, 形如:

$$\mathrm{Com}(x_1; r_1) + \mathrm{Com}(x_2; r_2) = \mathrm{Com}(x_1 + x_2; r_1 + r_2),$$

承诺方案主要用于保证协议的零知识性, 故而选用文献[17]中的多项式向量承诺方案, 具体算法如下所示:

密钥生成算法(CkeyGen): 输入安全参数 $(1^n, 1^k)$, 输出公共参数 $E=\begin{bmatrix} 1 & E_1 & E_2 \\ 0^{(n-1)\times 1} & \mathrm{I}_{n-1} & E_3 \end{bmatrix} \in R_q^{n\times k}$, 其中 $E_1 \xleftarrow{\$} R_q^{1\times(n-1)}$, $E_2 \xleftarrow{\$} R_q^{1\times(k-n)}$, $E_3 \xleftarrow{\$} R_q^{(n-1)\times(k-n)}$。

承诺算法(Com): 输入消息 $x \in R_q^{n-1}$, 选择随机数 $r \xleftarrow{\$} \mathcal{S}_{R,\beta}^k$, 计算承诺值:

$$c=\mathrm{Com}_{\mathrm{E}}(x;r)=E\times r+\begin{pmatrix}0\\ x\end{pmatrix},$$

然后承诺者将 c 发送给接收者。

验证算法(Verify): 承诺者向接收者发送 $x \in R_q^{n-1}$, $r \xleftarrow{\$} \mathcal{S}_{\infty,\beta}^k$, 接收者验证:

$$E\times r+\begin{pmatrix}0\\ x\end{pmatrix}=c$$

是否成立;

除此之外, 承诺者还可以发送 $x \in R_q^{n-1}$, $r \in R_q^k$ 以及较小的 $f \in \{\mathcal{C}-\mathcal{C}\}$, 验证者再验证:

$$E\times r+f\cdot\begin{pmatrix}0\\ x\end{pmatrix}=f\cdot c$$

是否成立。

定理 4 [17] 如果存在算法 $\mathcal{A}$ 能以优势 ε 打破承诺方案 $\mathrm{Com}_{\mathrm{E}}$ 的隐藏性, 则存在算法 $\mathcal{A}'$ 在相同时间量级内以优势 ε 解决 $\mathrm{DKS}_{n,k,\beta}^{\infty}$ 问题。

定理 5 [17] 如果存在算法 $\mathcal{A}$ 能以概率 ε 打破承诺方案 $\mathrm{Com}_{\mathrm{E}}$ 的绑定性, 则存在算法 $\mathcal{A}'$ 在相同时间量级内以优势 ε 解决 $\mathrm{SKS}_{1,k,16\sigma\sqrt{\kappa\cdot N}}^2$ 问题。

3 通用累加器设计

目前已有的 SIS 通用累加器[13,15,16], 都是基于 Merkle 哈希树的结构实现的, 而哈希函数作为哈希树的主要部件, 决定着累加器的安全性和计算的整体效率。另外, 对于以往 SIS 通用累加器, 都要求叶子节点呈升序排列, 故而每当有成员的加入或是撤销, 都需要重新排序, 再计算整棵 Merkle 哈希树。所以本文设计通用累加器的大致思想是通过替换哈希函数的底层假设, 从根本上提升内部计算效率。同时, 将单棵哈希树结构拆分成多棵子树结构, 如图 3 所示, 当通用累加器需要更新时, 只需对被操作元素对应的子树进行重新计算, 通过局部更新来实现通用累加器更新效率的大幅度提升。本节将主要介绍基于 Ring-SIS 的哈希函数以及基于 Ring-SIS 的通用累加器的具体构造。

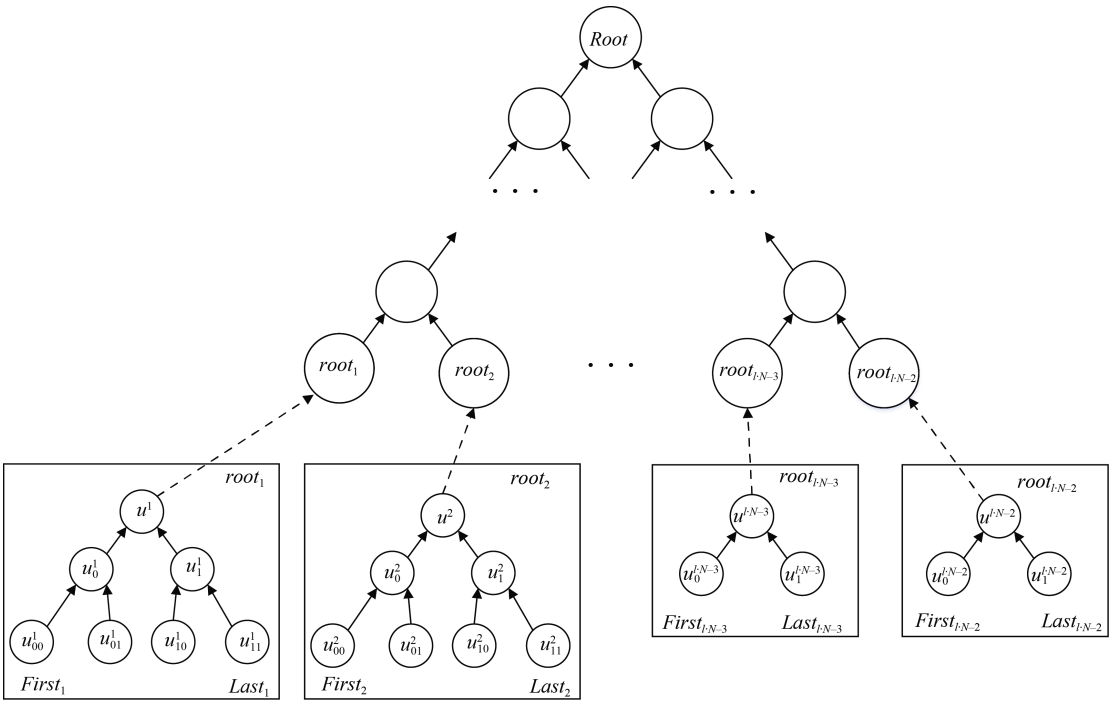


图 3　Ring-SIS 通用累加器设计图

Figure 3　The design of the Ring-SIS universal accumulator

3.1　抗碰撞哈希函数

在给出高效通用累加器算法之前, 本节首先将介绍保证高效通用累加器安全性的基础部件: 基于 Ring-SIS 问题困难性假设的抗碰撞哈希函数。本文所使用的哈希函数为了能与树型累加器所需要的数据形式相结合, 在一般背包函数形式的哈希函数[25-26]的基础上稍作修改, 具体构造如下所定义:

定义 6(哈希函数) 给定模数 q, 整数 $l=\lceil \log q \rceil$, $\mathrm{G}=(2^{l-1}, \cdots, 1)^{\mathrm{T}}$, 向量 $\boldsymbol{a}=(a_1, \cdots, a_{2l}) \xleftarrow{\$} R_q^{2l}$, 以多项式环向量 $\boldsymbol{e}=(e_1, \cdots, e_{2l}) \in R_2^{2l}$ 作为输入, 哈希函数 $\mathrm{H_a}$ 定义为

$$\mathrm{H_a}(e)=\Gamma(a_1\times e_1+\cdots+a_{2l}\times e_{2l} \bmod qR),$$

操作 $\Gamma:=R_q \to R_2^l$ 被定义为 $\Gamma(t)=(c_l, \cdots, c_1)$, 其中 $t=\sum_{i=1}^{N} t_i x^{i-1} \in R_q$。另外对于所有 $i\in[N]$ 与 $j\in[l]$, $\mathrm{Itb}_l(t_i)=(t_i^l, \cdots, t_i^1)$, 所以 $c_j=\sum_{k=1}^{N} t_k^j x^{k-1}$。

由上面哈希函数的结构可以推出:

$$\mathrm{H_a}(e)\times G=(a_1\times e_1+\cdots+a_{2l}\times e_{2l} \bmod qR).$$

定理 6 如果 Ring-$\mathrm{SIS}^{\infty}_{q,2l,2}$ 问题对于所有 PPT 算法是困难的, 则上述给出的哈希函数 $\mathrm{H_a}$ 是抗碰撞哈希函数。

证明. 假设函数 $\mathrm{H_a}$ 不是抗碰撞的, 则存在 PPT 算法 $\mathcal{A}$ 和一个多项式 poly(·), 对于无穷多个 n,

$$\Pr\left[(e,r)\leftarrow\mathcal{A}(1^n,h): e\neq r\wedge \mathrm{H_a}(e)=\mathrm{H_a}(r)=h\right] \geq 1/\mathrm{poly}(n),$$

即至少能以 $1/\mathrm{poly}(n)$ 的概率找到一对碰撞 (e,h_0) 和 (r,h_1) 满足下面关系:

$$\begin{cases} h_0\times \mathrm{G}=\sum_{i=1}^{2l} a_i\times e_i \bmod qR; \\ h_1\times \mathrm{G}=\sum_{i=1}^{2l} a_i\times r_i \bmod qR; \\ h_0=h_1. \end{cases}$$

通过这对碰撞可以在多项式时间内计算出 Ring-$\mathrm{SIS}^{\infty}_{q,2l,2}$ 问题的一个有效解:

$$\begin{aligned} z&=(z_1, \cdots, z_{2l}) \\ &=(e_1-r_1, \cdots, e_{2l}-r_{2l}) \bmod qR, \end{aligned}$$

使得 $\sum_{i=1}^{2l} a_i z_i \bmod qR=0$, 且 $\|z\|_{\infty}=2$, 该结果与前提假设矛盾。故函数 $\mathrm{H_a}$ 是抗碰撞的哈希函数。

3.2　通用累加器

本文设计的高效通用累加器是以 3.1 小节中基于 Ring-SIS 问题困难性假设的哈希函数 $\mathrm{H_a}$ 为基础部件, 一共涉及四个基础算法以及一个通用累加器更新算法。

令 $p=l\cdot N-2$，另外要求安全参数 N 使得 p 满足 $p=2^{g_0}$，其中 g_0 为正整数。这里定义被累加值的选择空间为:

$$\mathcal{K}:=\left\{s \mid s\in R_2^l \wedge s\notin\{t_1,t_2,t_3,t_4,\{First_j,Last_j\}_{j\in[p]}\}\right\},$$

其中

$$\begin{cases} t_1=(0,\ \cdots,\ 0,0)\in R_2^l; \\ t_2=(0,\ \cdots,\ 0,1)\in R_2^l; \\ t_3=(0,\ \cdots,\ 0,x)\in R_2^l; \\ t_4=(0,\ \cdots,\ 0,x+1)\in R_2^l; \end{cases}$$

另外，对于 $First_j$ 与 $Last_j$ 的定义将在随后的启动算法中给出。

启动算法(Setup):

(1) 输入安全参数 N。

(2) 均匀随机选择参数 $\mathrm{a}\xleftarrow{\$}R_q^{2l}$，设置如下辅助节点:

对 $\forall j\in[p]$，令 $First_j=(First_j^l,\ \cdots,\ First_j^1)\in R_2^l$，且对任意 $k\in[l]$，满足:

$$First_j^k=\begin{cases} x^{(j+1)\bmod N}, & k=(j+1)/N+1 \\ 0, & k\neq(j+1)/N+1 \end{cases}$$

对 $\forall j\in[p]$，令 $Last_j=(Last_j^l,\ \cdots,\ Last_j^1)\in R_2^l$，且对任意 $k\in[l]$，满足:

$$Last_j^k=\begin{cases} \sum_{m=0}^{N-1}x^m, & k<(j+1)/N+1 \\ \sum_{m=0}^{(j+1)\bmod N}x^m, & k=(j+1)/N+1 \\ 0, & k>(j+1)/N+1 \end{cases}$$

(3) 输出公共参数为 $\mathrm{pp}=\{\mathrm{a},\{First_j,Last_j\}_{j\in[p]}\}$。

累加器生成算法(Acc):

1) 输入公共参数 pp 以及升序集合 $S\subseteq\mathcal{K}$。

2) 将集合 S 按照如下方法划分为 p 个互不相交的升序子集，即 $S=\{S_i\}_{i\in[p]}$。对于任意 $s=(s_l,\ \cdots,\ s_1)\in S_i$，要求 $\mathrm{deep}(\mathrm{ptb}_l(s))-2=i$，其中 $\mathrm{deep}(\mathrm{ptb}_l(s))\in[l\cdot N]$ 表示比特串 $\mathrm{ptb}_l(s)\in\{0,1\}^{l\cdot N}$ 中的从右至左的非零最高位。

另外令 $k_i=|S_i|$，$k=\sum_{i=1}^{p}k_i$，任意 S_i 可以表示为形如 $S_i=\{s_i^1,\ \cdots,\ s_i^{k_i}\}$ 的形式。

3) 对于所有子集 S_i，$\forall i\in[p]$，加入辅助元素，各自计算新的子集 $\tilde{S}_i=\{First_i,S_i,Last_i\}=\{\tilde{s}_i^1,\ \cdots,\ \tilde{s}_i^{k_i+2}\}$，令 $g_i=\lceil\log(k_i+2)\rceil$，通过算法 $\mathrm{acc}_{\mathrm{pp}}(\tilde{S}_i,g_i)$ 对新子集 $\tilde{S}_i$ 计算子树 T_i，并返回 T_i 的根 $root_i$。

算法 $\mathrm{acc}_{\mathrm{pp}}(\tilde{S}_i,g_i)$:

(1) 对所有 $b=(b_1,\ \cdots,b_{g_i})\in\{0,1\}^{g_i}$，令 $t=\mathrm{btI}(b)+1$，初始化 $u^i_{(b_1,\cdots,b_{g_i})}=\tilde{s}_i^t\in R_2^l$。

(2) 对于 $\forall j\in\{g_i-1,\ \cdots,\ 1\}$ 与 $(b_1,\ \cdots,\ b_j)\in\{0,1\}^j$ 计算 $u^i_{(b_1,\cdots,b_j)}=\mathrm{H_a}(u^i_{(b_1,\cdots,b_j,\,0)},u^i_{(b_1,\cdots,b_j,1)})\in R_2^l$。

(3) 计算 $u^i=\mathrm{H_a}(u_0^i,u_1^i)\in R_2^l$，令 $root_i=u^i$，返回 $root_i$。

4) 调用算法 $\mathrm{acc}_{\mathrm{pp}}(root,g_0)$，为集合 $root=\{root_i\}_{i\in[p]}$ 构造一颗最终树 T，并返回根 $Root$。

5) 输出累加值 $\hat{u}=Root$。

证据生成算法(Wit):

1) 输入公共参数 pp，累加值 $\hat{\mathrm{u}}$，元素 $\delta\in\mathcal{K}$ 以及类型标识 $type$。

2) 当 $type=0$ 且 $\delta\in S$，输出关于 $\delta\in S$ 的证据 w:

(1) 令 $n_d=\mathrm{deep}(\delta)$，$i=n_d-2$，故 $\delta\in S_i\subseteq S$。

(2) 在子树 T_i 中找到元素 δ 对应的叶子节点 $u^i_{(b_1,\cdots,b_{g_i})}\in R_2^l$，则关于 $\delta\in S$ 的证据可以表示成如下形式:

$$w=((b_1,\ \cdots,\ b_{g_i+g_0}),\ (w_{g_i+g_0},\cdots,\ w_1)),$$

其中

$$(b_{g_i+1},\ \cdots,\ b_{g_i+g_0})=\mathrm{Itb}_{g_0}(i-1);$$
$$w_{g_i+g_0}=u^i_{(b_1,\ \cdots,\ b_{g_i-1},\ \bar{b}_{g_i})},\ \cdots,\ w_{g_0+1}=u^i_{(\bar{b}_1)};$$
$$w_{g_0}=\hat{\mathrm{u}}_{(b_{g_i+1},\ \cdots,\ b_{g_i+g0-1},\ \bar{b}_{g_i+g0})},\ \cdots,\ w_1=\hat{\mathrm{u}}_{(\bar{b}_{g_i+1})}。$$

(3) 输出证据 w。

3) 当 $type=1$ 且 $\delta\notin S$，输出关于 $\delta\notin S$ 的证据 w:

(1) 令 $n_d=\mathrm{deep}(\delta)$，$i=n_d-2$，故 $d\notin S_i$。

(2) 集合 $\tilde{S}_i=\{First_i,S_i,Last_i\}$ 中存在一对相邻元素 δ_0，δ_1 满足 $\mathrm{ptb}_l(\delta_0)<\mathrm{ptb}_l(\delta)<\mathrm{ptb}_l(\delta_1)$。

(3) 在子树 T_i 中找到 δ_0，δ_1 对应的叶子节点，即 $u^i_{(b_1^0,\ \ldots,\ b_{g_i}^0)}=\delta_0$，$u^i_{(b_1^1,\ \ldots,\ b_{g_i}^1)}=\delta_1$，元素 $\delta\notin S$ 的证

据可以表示成如下形式:

$$w=\left((b_1,\ \cdots,\ b_{2g_i+g_0}),\ (w_{2(g_i+1)+g_0},\ \cdots,\ w_1)\right),$$

其中

$$(b_1,\ \cdots,\ b_{2g_i})=(b_1^0,\ \cdots,\ b_{g_i}^0,\ b_1^1,\ \cdots,\ b_{g_i}^1);$$
$$(b_{2g_i+1},\ \cdots,\ b_{2g_i+g_0})=\mathrm{Itb}_{g_0}(i+1);$$
$$w_{2(g_i+1)+g_0}=u^0_{(b_1^0,\ \cdots,\ b_{g_i}^0)};\ w_{g_i+g_0+1}=u^1_{(b_1^1,\ \cdots,\ b_{g_i}^1)};$$
$$w_{2(g_i+1)+g_0-1}=u^0_{(b_1^0,\ \cdots,\ b_{g_i-1}^0,\ \bar{b}_{g_i}^0)},\ \cdots,\ w_{g_i+g_0+2}=u^i_{(\bar{b}_1^0)};$$
$$w_{g_i+g_0}=u^1_{(b_1^1,\ \cdots,\ b_{g_i-1}^1,\ \bar{b}_{g_i}^1)},\ \cdots,\ w_{g_0+1}=u^i_{(\bar{b}_1^1)};$$
$$w_{g_0}=\hat{u}_{(b_{2g_i+1},\ \cdots,\ b_{2g_i+g_0-1},\ \bar{b}_{2g_i+g_0})},\ \cdots,\ w_1=\hat{u}_{(\bar{b}_{2g_i+1})}。$$

(4) 输出证据 w。

验证算法(Ver **)**:

1) 输入公共参数 pp，元素 $\delta\in\mathcal{K}$，证据 w，累加值 $\hat{u}$ 以及类型标识 $type$。

2) 当 $type=0$，检查是否 $\delta\in S$:

(1) 令 $n_d=\mathrm{deep}(\delta)$，$i=n_d-2$，$v_{g_i+g_0}=\ \delta\in R_2^l$。

(2) 计算子树 T_i 中叶子节点 $u^i_{(b_1,\ \cdots,\ b_{g_i})}$ 至树 T 根节点 $Root$ 的沿路路径, 即对 $\forall j\in\{g_i-1+g_0,\ \cdots,\ 0\}$，计算 $v_j=\bar{b}_{j+1}\mathrm{H_a}(v_{j+1},w_{j+1})+b_{j+1}\mathrm{H_a}(w_{j+1},v_{j+1})\in R_2^l$。

(3) 若 $v_0=\hat{u}$，则输出 1, 否则输出 0。

3) 当 $type=1$，检查是否 $\delta\notin S$:

(1) 若 $\mathrm{ptb}_l(w_{2(g_i+1)+g_0})<\mathrm{ptb}_l(\delta)<\mathrm{ptb}_l(w_{g_i+1+g_0})$ 满足, 则继续, 否则输出 0。

(2) 若 $(b_1,\ \cdots,\ b_{g_i})+1=(b_{g_i+1},\ \cdots,\ b_{2g_i})$ 满足, 则继续, 否则输出 0。

(3) 令

$$\begin{cases}\tilde{w}_1=\begin{pmatrix}(b_1,\ \cdots,\ b_{g_i},\ b_{2g_i+1},\ \cdots,\ b_{2g_i+g_0}),\\ (w_{2(g_i+1)+g_0-1},\ \cdots,\ w_{g_i+2+g_0},\ w_{g_0},\ \cdots,\ w_1)\end{pmatrix};\\ \tilde{w}_2=\begin{pmatrix}(b_{g_i+1},\ \cdots,\ b_{2g_i},\ b_{2g_i+1},\ \cdots,\ b_{2g_i+g_0}),\\ (w_{g_i+g_0},\ \cdots,\ w_{g_0+1},\ w_{g_0},\ \cdots,\ w_1)\end{pmatrix};\end{cases}$$

若

$$\begin{cases}\mathrm{Ver_{pp}}(\hat{u},w_{2(g_i+1)+g_0},\tilde{w}_1,0)=1;\\ \mathrm{Ver_{pp}}(\hat{u},w_{g_i+1+g_0},\tilde{w}_2,0)=1;\end{cases}$$

同时满足, 则输出 1。

累加器更新算法(AccUpdate **)**:

1) 输入公共参数 pp，被操作元素 $\delta\in\mathcal{K}$ 以及操作标识 opt。

2) 当 $opt=0\wedge\delta\notin S$，表示向通用累加器中添加新元素 δ:

(1) 令 $i=\mathrm{deep}(\delta)-2$，将 δ 加入到升序子集 $S_i\subseteq\mathrm{S}$，得到新的升序集 $S_i=\{S_i,\delta\}$，此时 $k_i=|S_i|$。

(2) 令 $\tilde{S}_i=\{First_i,S_i,Last_i\}$，$g_i=\lceil\log(k_i+2)\rceil$，调用算法 $\mathrm{acc_{pp}}(\tilde{S}_i,g_i)$ 重新计算子树 T_i 中所有节点，得到子树 T_i 的根节点 $root_i\in R_2^l$。

(3) 令 $(\hat{b}_1,\cdots,\hat{b}_{g_0})=\mathrm{Itb}_{g_0}(i-1)$，$\hat{u}_{(\hat{b}_1,\ \cdots,\ \hat{b}_{g0})}=root_i$，更新最终树 T 中叶子节点 $\hat{u}_{(\hat{b}_1,\ \cdots,\ \hat{b}_{g_0})}$ 至根节点 $Root$ 的沿路节点。即对所有 $j\in\{g_0-1,\ \cdots,\ 1\}$，计算 $\hat{u}_{(\hat{b}_1,\ \cdots,\ \hat{b}_j)}=\mathrm{H_a}(\hat{u}_{(\hat{b}_1,\ \cdots,\ \hat{b}_{j-1},\ 0)},\hat{u}_{(\hat{b}_1,\ \cdots,\ \hat{b}_{j-1},\ 1)})$, 最后再计算 $\hat{u}=\mathrm{H_a}(\hat{u}_0,\hat{u}_1)$。

(4) 输出最终树 T 的根节点 $Root=\hat{u}$。

3) 当 $opt=1\wedge\delta\in S$，表示从通用累加器中删除元素 δ:

(1) 令 $i=\mathrm{deep}(\delta)-2$，将 δ 从升序子集 $S_i\subseteq S$ 中删除，得到新的升序集 $S_i=S_i/\delta$，此时 $k_i=|S_i|$。

(2) 随后按照步骤 2 中的(2)至(4)执行，最终输出最终树 T 更新后的根节点 $Root$。

3.3 安全性分析

定理 7 如果 $\text{Ring-SIS}^{\infty}_{q,2l,2}$ 问题对于所有 PPT 算法是困难的, 根据 2.2.3 小节中通用累加器的安全性定义可知, 3.2 小节给出通用累加器 ACC 是安全的。

证明. 假设 ACC 不是安全的，则存在一个 PPT 敌手算法 $\mathcal{B}$ 以及一个多项式 poly(·)，对于无穷多个 N，使得

$$\Pr\left[\begin{array}{l}\mathrm{pp}\leftarrow\mathrm{Setup}(N);(S,\delta,w_1,w_2,type)\leftarrow\mathcal{B}(\mathrm{pp}):\\ (\delta\in S\wedge\mathrm{Ver_{pp}}(\mathrm{Acc}(\mathrm{pp},S),\delta,w_1,\mathrm{type}=1)=1)\\ \vee(\delta\notin S\wedge\mathrm{Ver_{pp}}(\mathrm{Acc}(\mathrm{pp},S),\delta,w_2,\mathrm{type}=0)=1)\end{array}\right]\geqslant 1/\mathrm{poly}(N),$$

然后可以利用敌手算法 $\mathcal{B}$ 构造出另一个 PPT 敌手算法 $\mathcal{A}$ 来打破哈希函数 $\mathrm{H_a}$ 的抗碰撞性，进而打破格上 $\text{Ring-SIS}^{\infty}_{q,2l,2}$ 问题的困难性假设，以此证明假设的矛盾性 。

令事件 A 为 $\delta\in S$，同时 PPT 算法 $\mathcal{B}$ 能伪造 $\delta\notin S$ 的有效非成员关系证据; 令事件 B 为 $\delta\notin S$，同时 PPT 算法 $\mathcal{B}$ 能伪造 $\delta\in S$ 的有效成员关系证据, 根据假设可以得出:

$$\Pr[\mathrm{A}]+\Pr[\mathrm{B}]\geqslant 1/\mathrm{poly}(N).$$

令事件 C 为存在一个 PPT 算法找到哈希函数 $\mathrm{H_a}$ 的一对碰撞, 那么事件 C 发生的概率

$$\Pr[\mathrm{C}]\geqslant \Pr[\mathrm{C}\,|\,\mathrm{A}]\cdot\Pr[\mathrm{A}]+\Pr[\mathrm{C}\,|\,\mathrm{B}]\cdot\Pr[\mathrm{B}].$$

下面将讨论: 在事件 A 发生的前提下, 事件 C 发生的概率。

若 $\delta\in S\wedge type=1$, 令 $i=\mathrm{deep}(\delta)-2$, 则算法 $\mathcal{B}$ 伪造的证据为

$$w=\begin{pmatrix}(b_1^0,\ \cdots,\ b_{g_i}^0,\ b_1^1,\ \cdots,\ b_{g_i}^1,\ \hat{b}_1,\ \cdots,\ \hat{b}_{g_0}),\\ (w_{g_i+1}^0,\ \cdots,\ w_1^0,\ w_{g_i+1}^1,\ \cdots,\ w_1^1,\ \hat{w}_{g_0},\ \cdots,\ \hat{w}_1)\end{pmatrix}.$$

通过累加器生成算法 $\mathrm{Acc(pp},\ S)\to\hat{u}$ 计算出的累加器中所有节点, 可以获得子树 T_i 中叶子节点 $u^i_{(b_1^0,\ \cdots,\ b_{g_i}^0)}$ 至最终树 T 中根节点 *Root* 的路径:

$$\begin{aligned}\tilde{l}^0&=(u^i_{(b_1^0,\ \cdots,\ b_{g_i}^0)},\ \cdots,\ u^i_{(b_1^0)},\ \hat{u}_{(\hat{b}_1,\ \cdots,\ \hat{b}_{g_0})},\ \cdots,\ \hat{u}_{(\hat{b}_1)},\ \hat{u})\\&=(\tilde{l}^0_{g_i+g_0},\ \cdots,\ \tilde{l}^0_0),\end{aligned}$$

子树 T_i 中叶子节点 $u^i_{(b_1^1,\ \cdots,\ b_{g_i}^1)}$ 至最终树 T 中根节点 *Root* 的路径:

$$\begin{aligned}\tilde{l}^1&=(u^i_{(b_1^1,\ \cdots,\ b_{g_i}^1)},\ \cdots,\ u^i_{(b_1^1)},\ \hat{u}_{(\hat{b}_1,\ \cdots,\ \hat{b}_{g_0})},\ \cdots,\ \hat{u}_{(\hat{b}_1)},\ \hat{u})\\&=(\tilde{l}^1_{g_i+g_0},\ \cdots,\ \tilde{l}^1_0).\end{aligned}$$

另外通过验证算法 $\mathrm{Ver_{pp}}(\hat{u},\delta,w,1)$ 可以计算出路径:

$$\begin{aligned}l^0&=(v^0_{g_i}=w^0_{g_i+1},\ \cdots,\ v^0_1,\ \hat{v}_{g_0},\ \cdots,\ \hat{v}_1,\ \hat{v}_0=\hat{u})\\&=(l^0_{g_i+g_0},\ \cdots,\ l^0_0)\end{aligned}$$

和路径:

$$\begin{aligned}l^1&=(v^1_{g_i}=w^1_{g_i+1},\ \cdots,\ v^1_1,\ \hat{v}_{g_0},\ \cdots,\ \hat{v}_1,\ \hat{v}_0=\hat{u})\\&=(l^1_{g_i+g_0},\ \cdots,\ l^1_0).\end{aligned}$$

且由于 $\delta\in S$, 所以有如下不等式关系:

$$\mathrm{ptb}_l(w^0_{g_i+1})<\mathrm{ptb}_l(\delta)<\mathrm{ptb}_l(w^1_{g_i+1}).$$

当 $l^0=\tilde{l}^0$, 则 $\tilde{l}^1_{g_i+g_0}=\delta$, 对比路径 l^1 和 $\tilde{l}^1$, 有 $l^1_{g_i+g_0}\neq\delta$, 所以能找到一个最小整数 k 满足 $l^1_k\neq\tilde{l}^1_k$, 即找到了哈希值为 $l^1_{k\text{-}1}$ 的哈希碰撞; 同理, 若 $l^1=\tilde{l}^1$, 对比路径 l^0 和 $\tilde{l}^0$, 也能找到一个最小整数 k 满足 $l^0_k\neq\tilde{l}^0_k$, 并找到了哈希函数 $\mathrm{H_a}$ 的一对碰撞, 所以 $\Pr[\mathrm{C}\,|\,\mathrm{A}]=1$。

下面将讨论: 在事件 B 发生的前提下, 事件 C 发生的概率。

若 $\delta\notin S\wedge type=0$, 令 $i=\mathrm{deep}(\delta)-2$, 则算法 $\mathcal{B}$ 伪造的证据为 $w=((b_1,\ \cdots,\ b_{g_i+g_0}),\ (w_{g_i+g_0},\ \cdots,\ w_1))$, 然后通过累加器生成算法 $\mathrm{Acc_{pp}}(S)\to\hat{u}$ 可以获得树 T 中的一条路径:

$$\begin{aligned}\tilde{l}&=(u^i_{(b_1,\ \cdots,\ b_{g_i})},\ \cdots,\ u^i_{(b_1)},\ \hat{u}_{g_0},\ \cdots,\ \hat{u}_1,\ \hat{u})\\&=(\tilde{l}_{g_i+g_0},\ \cdots,\ \tilde{l}_0).\end{aligned}$$

另外, 通过验证算法 $\mathrm{Ver_{pp}}(\hat{u},\delta,w,0)$ 可以计算出路径:

$$l=(v_{g_i+g_0}=\delta,\ \cdots,\ v_0=\hat{u}).$$

由于 $\delta\notin S$, 故 $u_{(b_1,\ \cdots,\ b_{g_i+g_0})}\neq v_{g_i+g_0}$, 同时可以找到一个最小整数 k 使得 $\tilde{l}_k\neq v_k$, 即找到了哈希值为 v_{k-1} 的一对哈希碰撞, 所以 $\Pr[\mathrm{C}\,|\,\mathrm{B}]=1$。

综上可以得出事件 C 发生的概率:

$$\begin{aligned}\Pr[\mathrm{C}]&\geqslant\Pr[\mathrm{C}\,|\,\mathrm{A}]\cdot\Pr[\mathrm{A}]+\Pr[\mathrm{C}\,|\,\mathrm{B}]\cdot\Pr[\mathrm{B}]\\&\geqslant\Pr[\mathrm{A}]+\Pr[\mathrm{B}]\\&\geqslant 1/\mathrm{poly}(N).\end{aligned}$$

该结果与哈希函数 $\mathrm{H_a}$ 的抗碰撞性定义相矛盾。根据定理 6 可以得知当存在 PPT 算法打破哈希函数 $\mathrm{H_a}$ 的抗碰撞性, 也能找到一个 PPT 算法打破 $\text{Ring-SIS}^{\infty}_{q,2l,2}$ 问题的困难性假设。

综上所述, 若 $\text{Ring-SIS}^{\infty}_{q,2l,2}$ 问题对所有 PPT 算法是困难的, 则通用累加器 ACC 是安全的。

4 零知识证明

本节主要介绍 Ring-SIS 通用累加器相关的, 被累加值的零知识证明协议 Π。在此之前, 首先简要介绍协议中主要用到的一些技巧以及协议所证断言的变形过程。

4.1 秘密值之间特殊关系证明

若向量 $p\in\{0,1\}^k\subseteq R^k$, $t,v,f\in\{0,1\}^{k\cdot l}\subseteq R^{k\cdot l}$ 满足:

$$t=v-p*v+p*f,$$

则可以从等式组

$$\begin{cases}z_p=r_p+d\cdot p;\\ z_t=r_t+d\cdot t;\\ z_v=r_v+d\cdot v;\\ z_f=r_f+d\cdot f;\end{cases}$$

中获得如下关于 d 的线性关系式:

$$\begin{aligned}&d\cdot z_v-z_p*z_v+z_p*z_f-d\cdot z_t\\&=d^2\cdot(v-p*v+p*f-t)+r_p*r_f-r_p*r_v\\&\quad+d\cdot(r_v-p*r_v-r_p*v+p*r_f+r_p*f-r_t)\\&=d\cdot(r_v-p*r_v-r_p*v+p*r_f+r_p*f-r_t)\\&\quad+r_p*r_f-r_p*r_v,\end{aligned}$$

其中 $d\in\mathcal{C}$, $r_p\in R_q^k$, $r_t,r_v,r_f\in R_q^{k\cdot l}$。

为了证明秘密值 p,t,v,f 之间形如:

$$t = v - p * v + p * f$$

的关系, 证明者先选择随机向量 $r_t, r_v, r_f \in R_q^{k\cdot l}$, $r_p \in R_q^k$, 令

$$h_1 = r_v - p * r_v - r_p * v + p * r_f + r_p * f - r_t,$$
$$h_2 = r_p * r_f - r_p * r_v,$$

然后计算 $c_1 = \mathrm{Com}(h_1)$ 与 $c_2 = \mathrm{Com}(h_2)$, 并将其先发送给验证者。证明者在收到挑战值 $d \xleftarrow{\$} \mathcal{C}$ 后, 再计算并回复消息 z_b, z_f, z_t 给验证者, 最后由验证者验证:

$$\mathrm{Com}(d \cdot z_v - z_p * z_v + z_p * z_f - d \cdot z_t) = d \cdot c_1 - c_2$$

是否成立。

4.2 范围证明

为了证明秘密满足一些特殊的范围限制, 通常会采用范围限制转换技术来实现, 范围限制转换技术即将秘密值满足范围限制这一条件转换成秘密值满足特定关系等式的条件。比如, 为了证明秘密值 $\vec{s} \in \{0,1\}^k$, 则只要证明等式 $\vec{s} \circ \vec{s} - \vec{s} = 0$ 是否成立即可, 另外, 集合 $\{0,1\}^k$ 可以看做是 R^k 的真子集。

当 $d \in \mathcal{C}$, $\vec{r} \xleftarrow{\$} R_q^k$, 同时 $\vec{s} \circ \vec{s} - \vec{s} = 0$, 可以从

$$\begin{cases} \vec{z} = \vec{r} + d \cdot \vec{s}; \\ \vec{z} \circ \vec{z} = \vec{r} \circ \vec{r} + d^2 \cdot \vec{s}^2 + 2d \cdot \vec{r} \circ \vec{s}; \\ d \cdot \vec{z} = d \cdot \vec{r} + d^2 \cdot \vec{s}; \end{cases}$$

中, 推导出

$$\begin{aligned} &\vec{z} \circ \vec{z} - d \cdot \vec{z} \\ &= d^2 \cdot (\vec{s} \circ \vec{s} - \vec{s}) + d \cdot (2\vec{r} \circ \vec{s} - \vec{r}) + \vec{r} \circ \vec{r} \\ &= d \cdot (2\vec{r} \circ \vec{s} - \vec{r}) + \vec{r} \circ \vec{r}, \end{aligned}$$

最终得到了一个关于 d 的线性关系式。

为了证明 $\vec{s} \in \{0,1\}^k$, 证明者可以先取随机向量 $\vec{r} \xleftarrow{\$} R_q^k$, 然后计算承诺 $c_1 = \mathrm{Com}(2\vec{r} \circ \vec{s} - \vec{r})$ 与承诺 $c_2 = \mathrm{Com}(\vec{r} \circ \vec{r})$, 并将 c_1, c_2 一起发送给验证者, 证明者在收到挑战值 $d \xleftarrow{\$} \mathcal{C}$ 后, 计算消息 $\vec{z} = \vec{r} + d \cdot \vec{s}$, 并将 $\vec{z}$ 发送给验证者, 最后由验证者验证等式:

$$\mathrm{Com}(\vec{z}^2 - d \cdot \vec{z}) = d \cdot c_1 + c_2$$

是否成立即可。

4.3 被累加值的零知识证明

本小节将详细介绍如何使用秘密值的特殊关系证明以及范围证明等技巧来共同实现 Ring-SIS 通用累加器所对应的, 被累加值的零知识证明系统。该零知识系统具体可以表述成如下语言:

$$R_{\mathrm{acc}} = \left\{ \begin{array}{l} (a \in R_q^{2l}, u \in R_q^l);\ \delta \in R_q^l, w \in \{0,1\}^L \times R_q^{L\cdot l} : \\ \mathrm{Ver_a}(\hat{u}, \delta, w, 0) = 1 \end{array} \right\},$$

其中 $l = \lceil \log q \rceil$, $L = g_i + g_0$。

根据 3.2 小节中的成员关系验证算法的要求, 当 $h_{g+r} = \delta$, 对 $\forall j \in [L]$, 有下列等式成立:

$$\begin{cases} h_{j-1} \times G = \overline{b}_j \cdot \mathrm{H_a}(h_j, w_j) + b_j \cdot \mathrm{H_a}(w_j, h_j); \\ \hat{u} \times G = \overline{b}_1 \cdot \mathrm{H_a}(h_1, w_1) + b_1 \cdot \mathrm{H_a}(w_1, h_1); \end{cases}$$

上述等式也得能等价于如下形式:

$$\begin{cases} h_{j-1} \times G = \sum_{i=1}^{l} a_i (\overline{b}_j h_j^i + b_j w_j^i) + \sum_{i=1}^{l} a_{l+i} (\overline{b}_j w_j^i + b_j h_j^i); \\ \hat{u} \times G = \sum_{i=1}^{l} a_i (\overline{b}_1 h_1^i + b_1 w_1^i) + \sum_{i=1}^{l} a_i (\overline{b}_1 w_1^i + b_1 h_1^i). \end{cases}$$

为了更加简洁地描述上述验证等式, 我们可以定义如下参数符号:

$$a^1 = (a_1 \quad \dots \quad a_l);\ a^2 = (a_{l+1} \quad \dots \quad a_{2l});$$
$$G = (2^{l-1} \quad \dots \quad 1)^T;\ u = (0 \quad \dots \quad 0 \quad \hat{u} \times \mathrm{G})^T;$$
$$\mathrm{A} = \begin{pmatrix} a^1 & 0 & \dots & 0 \\ 0 & a^1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a^1 \end{pmatrix}, \mathrm{B} = \begin{pmatrix} a^2 & 0 & \dots & 0 \\ 0 & a^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a^2 \end{pmatrix} \in R_q^{L \times (L\cdot l)};$$
$$\mathrm{C} = \begin{pmatrix} \mathrm{G} & 0 & \dots & 0 \\ 0 & \mathrm{G} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \mathrm{G} \end{pmatrix} \in R_q^{L \times (L\cdot l)};\ \mathrm{I} = \begin{pmatrix} x^{N-1} \\ \dots \\ x \\ 1 \end{pmatrix} \in R_q^{N \times 1};$$

令矩阵 $\boldsymbol{t} \in \{0,1\}^{(L\cdot l)\times N}$, $\boldsymbol{y} \in \{0,1\}^{(L\cdot l)\times N}$, $\boldsymbol{f} \in \{0,1\}^{(L\cdot l)\times N}$, $\boldsymbol{v} \in \{0,1\}^{(L\cdot l+l)\times N}$ 来分别表示多项式向量的系数矩阵, 则 $\boldsymbol{t}, \boldsymbol{y}, \boldsymbol{v}, \boldsymbol{f}$ 需要分别满足下述关系:

$$\boldsymbol{t} \times \mathrm{I} = \begin{pmatrix} (\overline{b}_L h_L^i + b_L w_L^i)_{i\in[l]}^T \\ \dots \\ (\overline{b}_1 h_1^i + b_1 w_1^i)_{i\in[l]}^T \end{pmatrix};\ \boldsymbol{y} \times \mathrm{I} = \begin{pmatrix} (\overline{b}_L w_L^i + b_L h_L^i)_{i\in[l]}^T \\ \dots \\ (\overline{b}_1 w_1^i + b_1 h_1^i)_{i\in[l]}^T \end{pmatrix};$$
$$\boldsymbol{v} \times \mathrm{I} = \begin{pmatrix} (h_L^i)_{i\in[l]}^T \\ \dots \\ (h_0^i)_{i\in[l]}^T \end{pmatrix};\ \boldsymbol{f} \times \mathrm{I} = \begin{pmatrix} (w_L^i)_{i\in[l]}^T \\ \dots \\ (w_1^i)_{i\in[l]}^T \end{pmatrix};$$

最后令矩阵 $\boldsymbol{v}_1 = \boldsymbol{v}[1 : L \cdot l,\ :] \in \{0,1\}^{(L\cdot l)\times N}$ 来表示矩阵 $\boldsymbol{v}$ 中第 1 行至第 $L \cdot l$ 行的所有元素, 矩阵 $\boldsymbol{v}_2 = \boldsymbol{v}[l+1 : (L+1) \cdot l,\ :] \in \{0,1\}^{(L\cdot l)\times N}$ 表示矩阵 $\boldsymbol{v}$ 的第 l 行至第 $L \cdot l + l$ 行所有元素, $p = b \in \{0,1\}^L$。从验证等式:

$$\begin{cases} h_{j-1}\times G=\sum_{i=1}^{l}a_i(\bar{b}_j h_j^i+b_j w_j^i)+\sum_{i=1}^{l}a_{l+i}(\bar{b}_j w_j^i+b_j h_j^i) \\ \hat{u}\times G=\sum_{i=1}^{l}a_i(\bar{b}_1 h_1^i+b_1 w_1^i)+\sum_{i=1}^{l}a_i(\bar{b}_1 w_1^i+b_1 h_1^i) \end{cases},$$

可以得到简洁版的成员关系验证等式:

$$\begin{cases} t=\bar{p}*v_1+p*f; \\ y=\bar{p}*f+p*v_1; \\ u=\mathrm{A}\times(t\times\mathrm{I})+\mathrm{B}\times(y\times\mathrm{I})+\mathrm{C}\times(v_2\times\mathrm{I}). \end{cases}$$

下面将构造被累加值的零知识证明协议 Π 来零知识地证明秘密值满足上述成员关系验证等式。

协议 Π:

公共输入: $\mathrm{A},\mathrm{B}\in R_q^{L\times(L\cdot l)}$, $\mathrm{C}\in\mathbb{Z}_q^{L\times(L\cdot l)}$, $\mathrm{I}\in R_q^{N\times 1}$, $\mathrm{E}\in R_q^{(6L\cdot l\cdot N+l\cdot N+L+1)\times k}$, $u\in R_q^{L\times 1}$;

输入证据: $\boldsymbol{p}=\{0,1\}^L$, $\boldsymbol{t}=\{0,1\}^{(L\cdot l)\times N}$, $\boldsymbol{y}=\{0,1\}^{(L\cdot l)\times N}$, $\boldsymbol{f}=\{0,1\}^{(L\cdot l)\times N}$, $\boldsymbol{v}=\{0,1\}^{((L+1)\cdot l)\times N}$;

证明目标: 存在向量 $\boldsymbol{p}=\{0,1\}^L$, $\boldsymbol{t}=\{0,1\}^{(L\cdot l)\times N}$, $\boldsymbol{y}=\{0,1\}^{(L\cdot l)\times N}$, $\boldsymbol{f}=\{0,1\}^{(L\cdot l)\times N}$ 与 $\boldsymbol{v}=\{0,1\}^{((L+1)\cdot l)\times N}$ 满足下述关系:

$$\begin{cases} \boldsymbol{t}=v_1-\boldsymbol{p}*v_1+\boldsymbol{p}*\boldsymbol{f}; \\ \boldsymbol{y}=\boldsymbol{f}-\boldsymbol{p}*\boldsymbol{f}+\boldsymbol{p}*v_1; \\ \mathrm{A}\times(\boldsymbol{t}\times\mathrm{I})+\mathrm{B}\times(\boldsymbol{y}\times\mathrm{I})+\mathrm{C}\times(v_2\times\mathrm{I})=u; \end{cases}$$

其中 $\boldsymbol{v}_1=\boldsymbol{v}[1:L\cdot l,\ :]$, $\boldsymbol{v}_2=v[l+1:(L+1)\cdot l,\ :]$。

生成证明:

(1) 证明者选择以下随机数:

$$r_p\xleftarrow{\$}R_q^L;$$
$$r_v\xleftarrow{\$}R_q^{(L+1)\cdot l\times N};$$
$$r_f,\ r_t,\ r_y\xleftarrow{\$}R_q^{(L\cdot l)\times N};$$
$$\rho_{c_1},\ \rho_{c_2}\in\mathcal{S}_{\infty,\beta}^k;$$
$$\rho\in\mathcal{N}_\sigma^k.$$

定义如下参数:

$$r_{v1}=r_v[1:L\cdot l,\ :];$$
$$r_{v2}=r_v[l+1:(L+1)\cdot l,\ :];$$
$$h_1=r_{v1}-p*r_{v1}-r_p*v_1+p*r_f+r_p*f-r_t;$$
$$h_2=r_p*r_f-r_p*r_{v1};$$
$$h_3=r_f-p*r_f-r_p*f+p*r_{v1}+r_p*v_1-r_y;$$
$$h_4=-h_2;$$
$$\vec{r}=\mathrm{mtc}_5(r_b,r_v,r_w,r_t,r_y)\in R_q^{4L\cdot l\cdot N+l\cdot N+L};$$
$$\vec{s}=\mathrm{mtc}_5(b,v,w,t,y)\in\{0,\ 1\}^{4L\cdot l\cdot N+l\cdot N+L};$$
$$h_5=2\vec{r}\circ\vec{s}-\vec{r};$$
$$h_6=\vec{r}\circ\vec{r};$$

计算

$$c_1=\mathrm{Com}_\mathrm{E}(\mathrm{mtc}_3(h_1,h_3,h_5);\rho_{c_1});$$
$$c_2=\mathrm{Com}_\mathrm{E}(\mathrm{mtc}_3(h_2,h_4,h_6);\rho_{c_2});$$
$$P=\mathrm{A}\times(r_t\times\mathrm{I})+\mathrm{B}\times(r_y\times\mathrm{I})+\mathrm{C}\times(r_{v2}\times\mathrm{I});$$
$$P_1=\mathrm{E}\times\rho;$$

最后证明者将消息

$$a=(c_1,c_2,P,P_1)$$

发送给验证者。

(2) 验证者选择随机挑战 $d\xleftarrow{\$}\mathcal{C}$ 发送给证明者。

(3) 证明者检查挑战 d 是否有效, 若有效则计算如下消息:

$$z_p=r_p+d\cdot p;$$
$$z_v=r_v+d\cdot v;$$
$$z_f=r_f+d\cdot f;$$
$$z_t=r_t+d\cdot t;$$
$$z_y=r_y+d\cdot y;$$
$$\tilde{\rho}=\rho+\rho_{c_2}+d\cdot\rho_{c_1};$$

对计算的 $\tilde{\rho}$ 进行拒绝抽样, 即以概率

$$P_{abort}=1-\frac{\mathcal{N}_\sigma^k(\tilde{\rho})}{\mathrm{M}\cdot\mathcal{N}_{\rho_{c_2}+d\cdot\rho_{c_1}}^k(\tilde{\rho})}$$

判断协议是否中止。若协议未中断, 则证明者回复消息 $z=(z_p,\ z_v,\ z_f,\ z_t,\ z_y,\ \tilde{\rho})$。

(4) 验证者令

$$\vec{z}=\mathrm{mtc}_5(z_p,z_v,z_f,z_t,z_y)\in R_q^{4L\cdot l\cdot N+l\cdot N+L};$$
$$z_{v1}=z[1:L\cdot l,\ :];$$
$$z_{v2}=z[l+1:(L+1)\cdot l,\ :];$$
$$e_1=d\cdot z_{v1}-z_p*z_{v1}+z_p*z_f-d\cdot z_t;$$
$$e_2=d\cdot z_f-z_p*z_f+z_p*z_{v1}-d\cdot z_y;$$
$$e_3=\vec{z}\circ\vec{z}-d\cdot\vec{z};$$

并检查下述条件:

$$\begin{cases} \|\tilde{\rho}\|\leqslant 2\sigma\sqrt{k\cdot N}; \\ \mathrm{Com}_\mathrm{E}(\mathrm{mtc}_3(e_1,e_2,e_3);\tilde{\rho})=P_1+d\cdot c_1+c_2; \\ \mathrm{A}\times(z_t\times\mathrm{I})+\mathrm{B}\times(z_y\times\mathrm{I})+\mathrm{C}\times(z_{v2}\times\mathrm{I})=P+d\cdot u; \end{cases}$$

是否都成立。

4.4 安全性分析

完整性: 当协议不中止, 诚实证明者可以正确回复所有来自验证者的挑战 d, 由定理 2 可知, 当 $\sigma=\alpha\cdot T$, 可以获得 $\mathrm{M}=\exp(12/\alpha+1/(2\alpha^2))$, 协议中断的概率 $\mathrm{P_{abort}}\leqslant 1-\frac{1-2^{-100}}{\mathrm{M}}$。另外根据定理 1 可知 $\Pr[\|\tilde{\rho}\|\leqslant 2\sigma\sqrt{k\cdot N}]\geqslant 1-2^{-\frac{k\cdot N}{2}}$, 因此诚实证明者

使得诚实验证者相信的概率趋近于$\frac{1}{\mathrm{M}}$。

3-特殊合理性: 当恶意证明者能以$\frac{1}{|\mathcal{C}|}$的概率猜中挑战值, 或者能以不超过τ的概率伪造承诺并通过验证, 则合理性错误$error_s$最多为$\frac{1}{|\mathcal{C}|}+\tau$。将概率$\tau$降至$\frac{1}{|\mathcal{C}|}$, 则协议的合理性错误为$\frac{2}{|\mathcal{C}|}$。

假设恶意证明者至少能以概率$\frac{2}{|\mathcal{C}|}+\varepsilon$欺骗验证者, 其中$\varepsilon$为可忽略函数, 则恶意证明者至少能正确回答 3 条挑战。提取器可以根据 3 条不同挑战的有效副本$tr_1=(a,d,z)$, $tr_2=(a,d',z')$和$tr_3=(a,d'',z'')$, 计算并输出秘密:

$$\hat{p}=\frac{2\cdot z_p''-z_p-z_p'}{2\cdot d''-d-d'};\ \hat{f}=\frac{z_f''-z_f-z_f'}{2\cdot d''-d-d'};$$
$$\hat{v}=\frac{z_v''-z_v-z_v'}{2\cdot d''-d-d'};\ \hat{t}=\frac{z_t''-z_t-z_t'}{2\cdot d''-d-d'};$$
$$\hat{y}=\frac{z_y''-z_y-z_y'}{2\cdot d''-d-d'}.$$

接下来证明提取出的秘密具有合法有效性。令

$$\hat{\vec{r}}=\mathrm{mtc}_5(\hat{r}_p,\hat{r}_v,\hat{r}_f,\hat{r}_t,\hat{r}_y);$$
$$\hat{\vec{s}}=\mathrm{mtc}_5(\hat{p},\hat{v},\hat{f},\hat{t},\hat{y});$$
$$\begin{cases}\hat{v}_1=\hat{v}[1:(g-1)\cdot l,\ :];\\ \hat{v}_2=\hat{v}[l+1:(g+1)\cdot l,\ :];\end{cases}$$
$$\begin{cases}\hat{r}_{v1}=\hat{r}_v[1:(g-1)\cdot l,\ :];\\ \hat{r}_{v2}=\hat{r}_v[l+1:(g+1)\cdot l,\ :];\end{cases}$$

$$\begin{cases}z_p=\hat{r}_p+d\cdot\hat{p};\\ z_p'=\hat{r}_p+d'\cdot\hat{p};\\ z_p''=\hat{r}_p+d''\cdot\hat{p};\end{cases}\quad\begin{cases}z_{v1}=\hat{r}_{v1}+d\cdot\hat{v}_1;\\ z_{v1}'=\hat{r}_{v1}+d'\cdot\hat{v}_1;\\ z_{v1}''=\hat{r}_{v1}+d''\cdot\hat{v}_1;\end{cases}$$

$$\begin{cases}z_f=\hat{r}_f+d\cdot\hat{f};\\ z_f'=\hat{r}_f+d'\cdot\hat{f};\\ z_f''=\hat{r}_f+d''\cdot\hat{f};\end{cases}\quad\begin{cases}z_t=\hat{r}_t+d\cdot\hat{t};\\ z_t'=\hat{r}_t+d'\cdot\hat{t};\\ z_t''=\hat{r}_t+d''\cdot\hat{t};\end{cases}$$

$$\begin{cases}z_y=\hat{r}_y+d\cdot\hat{y};\\ z_y'=\hat{r}_y+d'\cdot\hat{y};\\ z_y''=\hat{r}_y+d''\cdot\hat{y};\end{cases}\quad\begin{cases}\vec{z}=\vec{r}+d\cdot\hat{\vec{s}};\\ \vec{z}'=\hat{\vec{r}}+d'\cdot\hat{\vec{s}};\\ \vec{z}''=\hat{\vec{r}}+d''\cdot\hat{\vec{s}};\end{cases}$$

$$\begin{cases}e_1=d\cdot z_{v1}-z_p*z_{v1}+z_p*z_f-d\cdot z_t;\\ e_1'=d'\cdot z_{v1}'-z_p'*z_{v1}'+z_p'*z_f'-d'\cdot z_t';\\ e_1''=d''\cdot z_{v1}''-z_p''*z_{v1}''+z_p''*z_f''-d''\cdot z_t'';\end{cases}$$

$$\begin{cases}e_2=d\cdot z_f-z_p*z_f+z_p*z_{v1}-d\cdot z_y;\\ e_2'=d'\cdot z_f'-z_p'*z_f'+z_p'*z_{v1}'-d'\cdot z_y';\\ e_2''=d''\cdot z_f''-z_p''*z_f''+z_p''*z_{v1}''-d''\cdot z_y'';\end{cases}$$

$$\begin{cases}e_3=\vec{z}\circ\vec{z}-d\cdot\vec{z};\\ e_3'=\vec{z}'\circ\vec{z}'-d'\cdot\vec{z}';\\ e_3''=\vec{z}''\circ\vec{z}''-d''\cdot\vec{z}'';\end{cases}$$

对于验证公式:

$$\begin{cases}①\ \mathrm{Com}_{\mathrm{E}}(\mathrm{mtc}_3(e_1,e_2,e_3);\tilde{\rho})=P_1+d\cdot c_1+c_2;\\ ②\ \mathrm{Com}_{\mathrm{E}}(\mathrm{mtc}_3(e_1',e_2',e_3');\tilde{\rho}')=P_1+d'\cdot c_1+c_2;\\ ③\ \mathrm{Com}_{\mathrm{E}}(\mathrm{mtc}_3(e_1'',e_2'',e_3'');\tilde{\rho}'')=P_1+d''\cdot c_1+c_2;\end{cases}$$

可以推导出:

④=②−③:

$$\mathrm{Com}_{\mathrm{E}}(\mathrm{mtc}_3(e_1',e_2',e_3')-\mathrm{mtc}_3(e_1'',e_2'',e_3'');\tilde{\rho}'-\tilde{\rho}'')=(d'-d'')\cdot c_1.$$

⑤=③−①:

$$\mathrm{Com}_{\mathrm{E}}(\mathrm{mtc}_3(e_1'',e_2'',e_3'')-\mathrm{mtc}_3(e_1,e_2,e_3);\tilde{\rho}''-\tilde{\rho})=(d''-d)\cdot c_1.$$

由$(d''-d)\cdot④-(d'-d'')\cdot⑤$可以再次推导出:

⑥:

$$(d''-d)\cdot\begin{pmatrix}0\\ \mathrm{mtc}_3(e_1',e_2',e_3')-\mathrm{mtc}_3(e_1'',e_2'',e_3'')\end{pmatrix}=(d'-d'')\cdot\begin{pmatrix}0\\ \mathrm{mtc}_3(e_1'',e_2'',e_3'')-\mathrm{mtc}_3(e_1,e_2,e_3)\end{pmatrix},$$

将$\hat{p},\hat{f},\hat{v},\hat{t},\hat{y}$代入到等式⑥, 最终可以得到:

$$(d'-d)\cdot\begin{pmatrix}0\\ \mathrm{mtc}(\hat{v}_1-\hat{p}*\hat{v}_1+\hat{p}*\hat{f}-\hat{t})\\ \mathrm{mtc}(\hat{f}-\hat{p}*\hat{f}+\hat{p}*\hat{v}_1-\hat{y})\\ \mathrm{mtc}(\hat{\vec{s}}\circ\hat{\vec{s}}-\hat{\vec{s}})\end{pmatrix}=0.$$

由于$d'-d\neq 0$, 且根据定理 4 可知, $d'-d$在R_q中是可逆的, 所以可以得出结论:

$$\begin{cases}\hat{v}_1-\hat{p}*\hat{v}_1+\hat{p}*\hat{f}-\hat{t}=0;\\ \hat{f}-\hat{p}*\hat{f}+\hat{p}*\hat{v}_1-\hat{y}=0;\\ \hat{\vec{s}}\circ\hat{\vec{s}}-\hat{\vec{s}}=0.\end{cases}$$

另外根据验证等式:

$$\begin{cases}\mathrm{A}\times(z_t\times\mathrm{I})+\mathrm{B}\times(z_y\times\mathrm{I})+\mathrm{C}\times(z_{v2}\times\mathrm{I})=P+d\cdot u;\\ \mathrm{A}\times(z_t'\times\mathrm{I})+\mathrm{B}\times(z_y'\times\mathrm{I})+\mathrm{C}\times(z_{v2}'\times\mathrm{I})=P+d'\cdot u;\\ \mathrm{A}\times(z_t''\times\mathrm{I})+\mathrm{B}\times(z_y''\times\mathrm{I})+\mathrm{C}\times(z_{v2}''\times\mathrm{I})=P+d''\cdot u;\end{cases}$$

可以推导出

$$\mathrm{A}\times(\hat{t}\times\mathrm{I})+\mathrm{B}\times(\hat{y}\times\mathrm{I})+\mathrm{C}\times(\hat{v}_2\times\mathrm{I})=u.$$

诚实验证者零知识性:

模拟器Sim:

(1) 随机选择

$$c_1',c_2'\xleftarrow{\$}R_q^{6L\cdot l\cdot N+l\cdot N+L+1};$$
$$\rho'\xleftarrow{\$}\mathcal{N}_\sigma^k;$$
$$P'\xleftarrow{\$}R_q^{L\times 1}$$
$$d'\xleftarrow{\$}\mathcal{C};$$

(2) 计算 $P_1' = \mathrm{E} \times \rho'$;

(3) 以概率 $P_{\text{abort}} = 1 - 1/M$ 输出副本:

$$tr_1' = \begin{pmatrix} c_1', c_2', P', P_1'; \\ d'; \\ \perp \end{pmatrix},$$

并中止程序, 否则继续步骤 4;

(4) 随机选择

$$z_p \xleftarrow{\$} R_q^{L\times 1};\ z_v \xleftarrow{\$} R_q^{(L+1)\cdot l\times N};$$
$$z_f, z_t, z_y \xleftarrow{\$} R_q^{(L\cdot l)\times N};$$
$$\tilde{\rho}' \xleftarrow{\$} \mathcal{N}_\sigma^k;$$

(5) 令

$$\vec{z}' = \mathrm{mtc}_5(z_p', z_v', z_f', z_t', z_y');$$
$$z_{v1}' = z_v'[1:L\cdot l,\ :];\ z_{v2}' = z_v'[l+1:(L+1)\cdot l,\ :];$$
$$e_1' = d'\cdot z_{v1}' - z_p' * z_{v1}' + z_p' * z_f' - d'\cdot z_t';$$
$$e_2' = d'\cdot z_f' - z_p' * z_f' + z_p' * z_v^1 - d'\cdot z_y';$$
$$e_3' = \vec{z}' \circ \vec{z}' - d'\cdot \vec{z}';$$

(6) 计算

$$P_1' = \mathrm{Com}_{\mathrm{E}}(\mathrm{mtc}_3(e_1', e_2', e_3');\ \tilde{\rho}') - d'\cdot c_1' - c_2';$$
$$P' = \mathrm{A}\times(z_t' \times \mathrm{I}) + \mathrm{B}\times(z_y' \times \mathrm{I}) + \mathrm{C}\times(z_{v2}' \times \mathrm{I}) - d'\cdot \mathrm{u};$$

(7) 以概率 $P_{\text{abort}} = 1 - 1/\mathrm{M}$ 输出副本:

$$tr_2' = \begin{pmatrix} c_1', c_2', P', P_1'; \\ d'; \\ z_p', z_v', z_f', z_t', z_y', \tilde{\rho}' \end{pmatrix}.$$

下面将论证模拟器 Sim 输出副本的分布与协议 Π 输出副本的分布不可区分。

在模拟器 Sim 被中止的情况下, 由于承诺方案的隐藏性, 所以 (c_1, c_2) 的分布与 (c_1', c_2') 的分布不可区分; P_1 与 P_1' 都来自于相同分布; P 的分布由来自于均匀随机分布的 r_t, r_y, r_v 共同决定, 所以 P 与 P' 也是不可区分的。故而在被中止情况下, 副本 tr_1' 的分布与协议 Π 输出副本的分布是不可区分的。

在模拟器 Sim 不被中止的情况下, 承诺值 (c_1, c_2) 的分布与 (c_1', c_2') 的分布不可区分依赖于承诺方案 Com 的隐藏性; 由于协议 Π 在不中止的情况下, 有效副本需要满足 $\|\tilde{\rho}\| \leqslant 2\sigma\sqrt{k\cdot N}$ 的条件, 又因为 $\tilde{\rho} = \rho + \rho_{c_2} + d\cdot\rho_{c_1}$, 故而 $\|\rho\| \leqslant 2\sigma\sqrt{k\cdot N}$。因此, 当 $\mathrm{DKS}^2_{(6L\cdot l\cdot N + l\cdot N + L + 1), k, 2\sigma\sqrt{k\cdot N}}$ 问题对于任意算法是困难的, P_1 的分布与 $R_q^{6L\cdot l\cdot N + l\cdot N + L + 1}$ 上均匀随机分布不可区分, 所以 P_1 的分布与 P_1' 的分布也是不可区分的, 具体可参考文献[18]定理 6 中的证明; 对于 P 的分布与 P' 的分布, 由于 $r_t, r_y, r_v, z_t', z_y', z_v'$ 都是均匀随机抽取的, 所以 P 与 P' 的分布也是均匀随机的; 对于 $(z_p, z_v, z_f, z_t, z_y)$ 的分布, 又由于 r_p, r_v, r_f, r_t, r_y 都分别取自均匀随机分布, 所以 $(z_p, z_v, z_f, z_t, z_y)$ 的分布与 $(z_p', z_v', z_f', z_t', z_y')$ 的分布也是不可区分的; 最后根据定理 2 可知, 由拒绝抽样所得的 $\tilde{\rho}$ 的分布与 $\mathcal{N}_\sigma^k$ 是统计不可区分的。

故综上, 模拟器 Sim 输出副本的分布与协议 Π 输出副本的分布是不可区分的。

4.5 拓展

关于基于 Ring-SIS 的通用累加器所对应的非被累加值的零知识证明系统, 具体可以表述成如下语言:

$$R_{\text{acc}} = \left\{ \begin{array}{l} (a \in R_q^{2l}, u \in R_q^l);\ \delta \in R_q^l, w \in \{0,1\}^{2L} \times R_q^{(2L+2)\cdot l} : \\ \mathrm{Ver}_{\mathrm{a}}(\hat{u}, \delta, w, 1) = 1 \end{array} \right\},$$

其中 $l = \lceil \log q \rceil$, $L = g_i + g_0$。

根据 3.2 小节中的非成员关系验证算法, 非被累加值的零知识证明需要证明者证明非被累加值所在子树中存在两个相邻叶子节点, 且非被累加值的大小在这两个叶子节点取值之间, 为了实现以上过程, 不仅需要被累加值的零知识证明, 还需要大整数范围证明以及大整数加法关系证明共同实现。由于基于 Schnorr-like 框架的大整数间加法关系的零知识证明还未取得相关进展, 故暂时未能实现对于合理性错误可忽略的非被累加值的零知识证明, 我们未来也将继续探究该问题。

5 总结

近年来, 各类抗量子的密码学工具逐渐走入人们的视野, 其最大的亮点主要在于更强的安全性, 但影响着其实用性的关键主要还是在于其效率的考量。本课题主要是从抗量子攻击通用累加器的效率角度, 以及相应知识的零知识协议的效率角度进行进行研究, 提出了计算效率和更新效率远优于以往方案的 Ring-SIS 通用累加器, 同时还提出了单轮次执行合理性错误可忽略的, 被累加值的零知识协议方案, 其直接解决了以往方案需要重复执行多次来降低合理性错误的弊端, 为基于格上困难性假设的通用累加器实用性相关的研究, 进行了近一步的探索。

参考文献

[1] Benaloh J, de Mare M. One-Way Accumulators: A Decentralized Alternative to Digital Signatures[M]. Advances in Cryptology — EUROCRYPT '93. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994: 274-285.

[2] Barić N, Pfitzmann B. Collision-Free Accumulators and Fail-Stop Signature Schemes without Trees[M]. Advances in Cryptology — EUROCRYPT '97. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997: 480-494.

[3] Camenisch J, Lysyanskaya A. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials[M]. Advances in Cryptology — CRYPTO 2002. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002: 61-76.

[4] Nguyen L. Accumulators from Bilinear Pairings and Applications [C] *Topics in Cryptology - CT-RSA 2005*, 2005.

[5] I. Damgård and N. Triandopoulos, Supporting Non-membership Proofs with Bilinear-map Accumulators, IACR Cryptology ePrint Archive, http://eprint.iacr.org/2008/538, 2008.

[6] Li J T, Li N H, Xue R. Universal Accumulators with Efficient Nonmembership Proofs [C]. *Applied Cryptography and Network Security*, 2007.

[7] Camacho P, Hevia A, Kiwi M, et al. Strong Accumulators from Collision-Resistant Hashing[J]. *International Journal of Information Security*, 2012, 11(5): 349-363.

[8] Lipmaa H. Secure Accumulators from Euclidean Rings without Trusted Setup[M]. Applied Cryptography and Network Security. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012: 224-240.

[9] Libert B, Ling S, Nguyen K, et al. Zero-Knowledge Arguments for Lattice-Based Accumulators: Logarithmic-Size Ring Signatures and Group Signatures without Trapdoors[C] *Advances in Cryptology - EUROCRYPT*, 2016.

[10] M. Ajtai, Generating Hard Instances of Lattice Problems[C]. *Acm Symposium on the Theory of Computing*, 1996.

[11] Stern J. A New Paradigm for Public Key Identification[C]. *IEEE Transactions on Information Theory*, 1996, 1757-1768.

[12] Jain A, Krenn S, Pietrzak K, et al. Commitments and Efficient Zero-Knowledge Proofs from Learning Parity with Noise[M]. Advances in Cryptology – ASIACRYPT 2012. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012: 663-680.

[13] Ling S, Nguyen K, Wang H X, et al. Lattice-Based Group Signatures: Achieving Full Dynamicity with Ease[C]. *Applied Cryptography and Network Security*, 2017.

[14] Libert B, Ling S, Nguyen K, et al. Lattice-Based Zero-Knowledge Arguments for Integer Relations[C]. *Advances in Cryptology*, 2018.

[15] Yu Z X, Au M H, Yang R P, et al. Lattice-Based Universal Accumulator with Nonmembership Arguments[C]. *Information Security and Privacy*, 2018

[16] Benhamouda F, Krenn S, Lyubashevsky V, et al. Efficient Zero-Knowledge Proofs for Commitments from Learning with Errors over Rings[M]. Computer Security—ESORICS 2015. Cham: Springer International Publishing, 2015: 305-325.

[17] C. Baum, I. Damgård, S. Oechsner and C. Peikert, "Efficient Commitments and Zero-Knowledge Protocols from Ring-SIS with Applications to Lattice-based Threshold Cryptosystems," IACR Cryptology ePrint Archive, http://eprint.iacr.org/2016/997, 2016.

[18] Baum C, Damgård I, Lyubashevsky V, et al. More Efficient Commitments from Structured Lattice Assumptions[M]. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018: 368-385.

[19] J. Bootle, V. Lyubashevsky and G. Seiler, "Algebraic Techniques for Short(er) Exact Lattice-Based Zero-Knowledge Proofs", IACR Cryptology ePrint Archive, http://eprint.iacr.org/2019/642, 2019.

[20] R. P. Yang, M. H. Au, Z. F. Zhang, Q. L. Xu and W. Whyte, Efficient Lattice-Based Zero-Knowledge Arguments with Standard Soundness: construction and applications, IACR Cryptology ePrint Archive, http://eprint.iacr.org/2019/747, 2019.

[21] Micciancio D. Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions[J]. *Computational Complexity*, 2007, 16(4): 365-411.

[22] Kawachi A, Tanaka K, Xagawa K. Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems[M]. Advances in Cryptology - ASIACRYPT 2008. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008: 372-389.

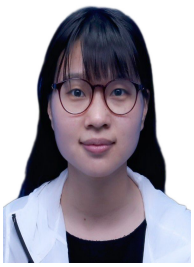
[23] Lyubashevsky V. Lattice Signatures without Trapdoors[M]. Advances in Cryptology – EUROCRYPT 2012. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012: 738-755.

[24] Lyubashevsky V, Seiler G. Short, Invertible Elements in Partially Splitting Cyclotomic Rings and Applications to Lattice-Based Zero-Knowledge Proofs[M]. Advances in Cryptology – EUROCRYPT 2018. Cham: Springer International Publishing, 2018: 204-224.

[25] Lyubashevsky V, Micciancio D. Generalized Compact Knapsacks are Collision Resistant[M]. Automata, Languages and Programming. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006: 144-155.

[26] Peikert C, Rosen A. Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices [C]. Theory of *Cryptography Conference*, 2006.

谭子欣 于 2017 年在湘潭大学计算机科学与技术专业获得工学学士学位。现在中国科学院信息工程研究所信息安全国家重点实验室网络空间安全专业攻读硕士学位。研究领域为安全协议与零知识证明。研究兴趣包括: 集合成员关系证明/非集合成系关系证明。Email: criss_tan_xtu@163.com



马丽 于 2017 年在河南大学计算机科学与技术专业获得工学学士学位。现在中国科学院信息工程研究所信息安全国家重点实验室计算机技术专业攻读硕士学位。研究领域为密码理论与技术。研究兴趣包括: 基于机器学习的密码设计和分析。Email: mali@iie.ac.cn

邓燚 于2008年在中科院软件所获得博士学位。现仍中科院信息工程研究所信息安全国家重点实验室研究员。研究领域为密码学与安全协议。研究兴趣包括零知识证明、安全规约方法、密码协议相关的复杂度研究以及这些技术在密码货币和区块链中的应用。Email: deng@iie.ac.cn