格上基于密文标准语言的可证明安全两轮口令认证密钥交换协议

尹安琪[1],曲彤洲[1],郭渊博[1],汪 定[2],陈 琳[1],李勇飞[1]
(1. 信息工程大学电子技术学院,河南郑州 450001; 2. 南开大学网络空间安全学院,天津 300350)

摘 要: 降低口令认证密钥交换(Password-based Authenticated Key Exchange,PAKE)协议的通信轮次和安全性假设是格上PAKE协议的重要优化方向.平滑投射哈希函数(Smooth Projective Hash Function,SPHF)是构造PAKE协议的重要数学工具,但现有的基于格的SPHF多不能在超多项式模数下应用.为此,本文提出了两种格上基于密文标准语言的SPHFs,在不增加通信开销和存储开销的前提下解决了上述问题.基于上述SPHFs,本文提出了一种基于格的可证明安全的两轮PAKE协议,该协议可以抵抗量子攻击,在不需要零知识证明和随机预言机的前提下,降低了协议通信轮次和安全性假设;本文还基于更加准确的标准安全模型对所提出的协议进行了严格的安全性证明.实验证明,本文提出的协议具有更优的通信轮次复杂度、计算开销、安全性假设和实际安全性.

关键词: 口令; 密钥交换; 平滑投射哈希函数; 可证明安全; 格; 抗量子
中图分类号: TN918.1 **文献标识码:** A **文章编号:** 0372-2112(2022)05-1140-10
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20210517

Provably Secure Two-Round PAKE Based on Ciphertext Standard Language over Lattices

YIN An-qi[1], QU Tong-zhou[1], GUO Yuan-bo[1], WANG Ding[2], CHEN Lin[1], LI Yong-fei[1]
(1. *College of Electronic Technology, Information Engineering University, Zhengzhou, Henan* 450001, *China*;
2. *College of Cyber Science, Nankai University, Tianjin* 300350, *China*)

Abstract: Reducing the communication round complexity and security assumptions are important directions of password-based authenticated key exchange(PAKE) protocol over lattices. Smooth projective Hash function(SPHF) is an important mathematical tool for constructing PAKE. But most of the existing lattice-based SPHFs cannot be applied under hyper-polynomial modulus. This paper proposes two SPHFs based on the standard language of ciphertext over lattices, which solves the above problem without increasing communication and storage overhead. Based on the proposed SPHFs, this paper proposes a provably secure two-round PAKE protocol over lattices, which can resist quantum attacks and reduce the communication round complexity and the security assumptions without random oracle and zero-knowledge proof. And this paper also provides a strict security proof for the proposed protocol based on a more accurate security model. Experiment results show that the protocol proposed has better communication round complexity, computational overhead, security assumptions and actual security.

Key words: password; key exchange; SPHF(Smooth Projective Hash Function); provably secure; lattice; post-quantum

1 引言

口令认证密钥交换协议可使只拥有低熵口令的协议参与方,在不安全的公开网络上,协商密码学安全的会话密钥[1]. 现有PAKE的研究多是基于传统困难问题的[2~5];Shor算法[6]证明,存在量子算法可以解密所有基于大整数分解或离散对数的公钥密码体制;因此上述PAKE无法抵抗量子攻击. 而基于格的密码体制可有效抵抗量子攻击;与其他抗量子密码体制(多变量公钥密

收稿日期:2021-04-21;修回日期:2021-09-26;责任编辑:覃怀银
基金项目:国家自然科学基金(No.61501515, No.61601515);信息保障技术重点实验室开放基金(No.KJ-15-108)

码体制、基于Hash函数的数字签名方案、基于编码的密码体制)相比,其在计算效率[7]、困难实例随机选取[8]等方面具有显著优势,但关于其的研究却相对有限.

平滑投射哈希函数是构造PAKE的重要数学工具.现有SPHF一般是基于非标准密文语言的,不能在超多项式模数下应用[9]. 文献[10]提出了一种支持一轮PAKE协议构造的适应性SPHF;但并不能解决上述问题. 文献[11]提出了一种基于密文标准语言的SPHF,但其所基于的公钥加密(Public Key Encryption,PKE)方案计算效率较低,且该文未对相关协议进行安全性证明.

KOY/GL[3,12]和JG/GK[4,5]架构是目前应用最广泛的PAKE协议架构,但都需要三轮通信. 文献[1]率先提出了格上的两轮PAKE;文献[13]则利用可拆分函数,提出了一种可实现互认证的两轮PAKE;但二者均是基于随机预言机的,且前者还需要基于可靠性模拟的非交互式零知识(Simulation-Sound Non-Interactive Zero- Knowledge,SS-NIZK)证明,计算效率较低. 文献[14]提出了第一个格上不需要SS-NIZK的两轮PAKE,但也是基于非标准密文语言的.

KOY/GL架构下的协议需要适应性选择密文攻击下不可区分(INDistinguishability under Adaptive Chosen-Ciphertext Attack,IND-CCA2)的安全性假设,这导致协议计算效率较低,存储开销较大. 文献[15]优化了PAKE的密文长度,但仍需要IND-CCA2的安全性假设. JG/GK架构下的协议在客户端仅需要选择明文攻击下不可区分(INDistinguishability under Chosen-Plaintext Attack,IND-CPA)的安全性假设,但此类协议仍需要三轮通信. 目前格上的二轮[1,16]、一轮[2,10]PAKE在服务器和客户端一般都需要IND-CCA2安全的加密方案. 虽然文献[14]提出的两轮协议降低了服务器端的安全性假设,但本文指出了其协议设计中的一个错误.

因此,研究格上基于密文标准语言的平滑投射哈希函数,是解决格上SPHF不能在超多项式模数下应用的有效方法;而降低格上PAKE协议的通信轮次和安全性假设则对降低PAKE协议的通信开销、计算开销以及提高协议的实际安全性具有重要意义. 为此,本文提出了两种格上基于密文标准语言的平滑投射哈希函数;并基于此提出了一种格上可证明安全的两轮口令认证密钥交换协议,协议降低了客户端所需的安全性假设;实验结果表明本文具有更优的计算开销、通信开销和安全性假设.

2　预备知识

2.1　符号说明

本文的符号定义见表1.

表1　符号定义

符号	含义
n	安全参数
$[\boldsymbol{A}\|\boldsymbol{B}]/[\boldsymbol{A}\|\|\boldsymbol{B}]$	矩阵$\boldsymbol{A}$和$\boldsymbol{B}$的横/纵向级联
negl(.)	可忽略函数
q	LWE困难问题模数
Ham(.,.)	汉明距离函数
$\boldsymbol{A}^{\mathrm{T}}$	矩阵$\boldsymbol{A}$的转置矩阵
$\leftarrow/\xleftarrow{r}$	取样/随机取样
$\|D\|$	集合D的大小
$d(.,.)$	距离函数(欧式距离)
$\|\|\boldsymbol{x}\|\|$	向量$\boldsymbol{x}$的欧几里得范数
$\perp$	非法标识
$\lceil a\rceil/\lfloor a\rfloor$	大于/小于a的最小/最大整数
$\lceil a\rfloor$	与a最接近的整数
$\langle .,. \rangle$	内积运算

2.2　格及格上的加密方案

格[1]　设$m\geq n$,基矩阵$\boldsymbol{B}\in\mathbb{R}^{m\times n}$($\boldsymbol{B}$的列向量线性无关),$m$维的格$\Lambda$格定义为

$$\Lambda_{\boldsymbol{B}}=\{\boldsymbol{Bs}|\boldsymbol{s}\in\mathbb{Z}^n\} \tag{1}$$

高斯离散分布[9]　设$s>0$,$\boldsymbol{y}\in\mathbb{R}^m$,定义$\mathbb{R}^m$上的高斯权重函数为

$$\rho_{s,\boldsymbol{y}}(\boldsymbol{x}):=\exp\left(-\pi\left|\boldsymbol{x}-\boldsymbol{y}\right|^2/s^2\right) \tag{2}$$

设参数为s,中心为$\boldsymbol{y}$,格Λ上的离散高斯分布为

$$D_{\Lambda,s,\boldsymbol{y}}(\boldsymbol{x})=\rho_{s,\boldsymbol{y}}(\boldsymbol{x})/\rho_{s,\boldsymbol{y}}(\Lambda) \tag{3}$$

其中$\boldsymbol{x}\in\Lambda$,$\rho_{s,\boldsymbol{y}}(\Lambda)=\sum_{\boldsymbol{x}\in\Lambda}\rho_{s,\boldsymbol{y}}(\boldsymbol{x})$,此外若$\boldsymbol{y}$为零向量,可将其省略.

带差错学习(Learning With Errors,LWE)问题[1]　令$q\geq 2$,另设χ是$\mathbb{Z}$上的离散高斯分布. LWE问题的定义为,给定多项式数目的取样,区分以下两个分布:(1)$(\boldsymbol{a},<\boldsymbol{a},\boldsymbol{s}>+x)$,$\boldsymbol{a}\xleftarrow{r}\mathbb{Z}_q^n$,$x\leftarrow\chi$,$\boldsymbol{s}\in\mathbb{Z}_q^n$是固定的随机均匀选择的秘密;(2)$(\boldsymbol{a},b)$,$\boldsymbol{a}\xleftarrow{r}\mathbb{Z}_q^n$,$b\xleftarrow{r}\mathbb{Z}_q$. LWE问题的困难性说明见文献[6].

格上的公钥加密方案　文献[17]中的GPV方案是格上经典的基于LWE问题的IND-CPA安全的PKE之一,设明文为$p_0\in\{0,1\}$,其定义如下.

GPV.KeyGen(params):私钥$\mathrm{sk}_{\mathrm{GPV}}=\boldsymbol{s}\xleftarrow{r}\mathbb{Z}_q^n$,公钥$\mathrm{pk}_{\mathrm{GPV}}=\boldsymbol{B}_{\mathrm{GPV}}=\boldsymbol{As}+\boldsymbol{x}\in\mathbb{Z}_q^n$,其中$x_i\xleftarrow{r}\chi(i\in[m],\chi=\bar{\psi}_\alpha)$.

GPV.Enc($\boldsymbol{B}_{\mathrm{GPV}},p_0;\boldsymbol{s}$):$\boldsymbol{y}\xleftarrow{r}D_{\mathbb{Z}^m,\boldsymbol{y}}$,密文$(\boldsymbol{u},c_0)=(\boldsymbol{B}_{\mathrm{GPV}}{}^{\mathrm{T}}\boldsymbol{y},\boldsymbol{B}_{\mathrm{GPV}}{}^{\mathrm{T}}\boldsymbol{y}+p_0\cdot\lfloor q/2\rfloor)\in\mathbb{Z}_q^{n+1}$.

GPV.Dec($\boldsymbol{s},(\boldsymbol{u},c_0)$):计算$p_0'=c_0-\boldsymbol{s}^{\mathrm{T}}\boldsymbol{u}$,若$\mathrm{Dis}(p_0',0)<\mathrm{Dis}(p_0',\lfloor q/2\rfloor)\bmod q$,输出0;否则,输出1.

文献[9]中的KV方案是格上最广泛应用的IND-CCA2安全的PKE之一. 设 $\boldsymbol{B}\in\mathbb{Z}_q^{m\times n}$, $\boldsymbol{U}=(\boldsymbol{u}_0,\boldsymbol{u}_1,\cdots,\boldsymbol{u}_l)\in(\mathbb{Z}_q^{m\times n})^{l+1}$, $\boldsymbol{x}\leftarrow\bar{\psi}_b^m\in\mathbb{Z}_q^l$, $\boldsymbol{s}\xleftarrow{r}\mathbb{Z}_q^n$, 公钥 $\mathrm{pk}_{\mathrm{KV}}=\boldsymbol{B}_{\mathrm{KV}}=[\boldsymbol{B}|\boldsymbol{U}]\in\mathbb{Z}_q^{m\times n}$, 标签 $\mathrm{label}=\boldsymbol{U}$, 明文 $\boldsymbol{p}=(p_1,p_2,\cdots,p_l)\in\mathbb{Z}_q^l$. KV方案定义如下.

KV.KeyGen(params):

$(\boldsymbol{B},\boldsymbol{T})\leftarrow\mathrm{TrapSamp}(1^n,q,m)$[18], 输出 $\boldsymbol{B}\in\mathbb{Z}_q^{m\times n}$, $\boldsymbol{T}\in\mathbb{Z}_q^{m\times m}$.

$\mathrm{KV.Enc}^{\mathrm{label}}(\boldsymbol{B}_{\mathrm{KV}},\boldsymbol{p};\boldsymbol{s})$: 为加密明文 $\boldsymbol{p}=(p_1,p_2,\cdots,p_l)\in\mathbb{Z}_q^l$, 随机选择一个错误向量 $\boldsymbol{x}\leftarrow\bar{\psi}_b^m\in\mathbb{Z}_q^l$, 计算并输出密文

$$\boldsymbol{y}=\boldsymbol{B}_{\mathrm{KV}}^{\mathrm{T}}\begin{pmatrix}\boldsymbol{s}^{\mathrm{T}}\\1\\\boldsymbol{p}^{\mathrm{T}}\end{pmatrix}+\boldsymbol{x}\in\mathbb{Z}_q^m \tag{4}$$

$\mathrm{KV.Dec}^{\mathrm{label}}(\boldsymbol{T},\boldsymbol{c})$: $(\boldsymbol{s},1,\boldsymbol{p})\leftarrow\mathrm{BBDsolve}(\boldsymbol{T},\boldsymbol{c},\boldsymbol{U})$[17], 输出 $\boldsymbol{p}$.

为简化说明, 下文以IND-CPA安全的方案为例构造格上的SPHF, 基于IND-CCA2安全的方案构造SPHF的方法与其相同.

2.3 平滑投射哈希函数

近似平滑投射哈希函数(Approximate Smooth Projective Hash Function, ASPHF)可通过抽样算法定义, 抽样算法输出 $(K,\ l,\ H=\{\mathrm{Hash}(\mathrm{params},\ \mathrm{hk},\ \boldsymbol{W})\}^l,\ S,\ \mathrm{HashKG}:\ \mathrm{hk}\xleftarrow{r}K,\ \mathrm{ProjKG}:\ K\rightarrow S)$; 其中 K、S 为哈希、投射密钥空间, HashKG为 K 上的哈希密钥生成函数; ProjKG为 K 到 S 上的投射密钥生成函数[2]. ASPHF满足:

(1) 存在高效算法可以计算HashKG、ProjKG和 $\mathrm{Hash}(\mathrm{params},\mathrm{hk},\boldsymbol{W})$;

(2) 近似正确性: 对于 $\forall\mathrm{hk}\in K$, $\forall\boldsymbol{W}\in L$, $\mathrm{ph}\leftarrow\mathrm{ProjKG}(\mathrm{params},\mathrm{hk},\boldsymbol{W})$ 以及可忽略函数 $\mathrm{negl}(\cdot)$, 若存在投射哈希函数ProjHash使式(5)成立, 称该ASPHF为 $\varepsilon-$ASPHF.

$$\Pr\left(\mathrm{Ham}\begin{pmatrix}\mathrm{Hash}(\mathrm{params},\mathrm{hk},\boldsymbol{W}),\\\mathrm{ProjHash}(\mathrm{ph},\boldsymbol{W},\boldsymbol{w})\end{pmatrix}\geq\varepsilon\right)=\mathrm{negl}(n) \tag{5}$$

(3) 平滑性: 对于 $\forall\mathrm{hk}\xleftarrow{r}K$, $\forall\boldsymbol{W}'\in X-L$, $\mathrm{ph}\leftarrow\mathrm{ProjKG}(\mathrm{params},\mathrm{hk},\boldsymbol{W}')$ 和 $v\xleftarrow{r}Y$ (Y 为Hash函数的值域), 式(6)中的两个分布在统计上不可区分.

$$\begin{cases}\begin{pmatrix}\mathrm{ph}\leftarrow\mathrm{ProjKG}(\mathrm{params},\mathrm{hk},\boldsymbol{W}'),\\h\leftarrow\mathrm{Hash}(\mathrm{params},\mathrm{hk},\boldsymbol{W}')\end{pmatrix}&(1)\\(\mathrm{ph}\leftarrow\mathrm{ProjKG}(\mathrm{params},\mathrm{hk},\boldsymbol{W}'),v)&(2)\end{cases} \tag{6}$$

现介绍证明本文SPHF性质所需的定理.

定理1[11] 对于 $\boldsymbol{A}\in\mathbb{Z}_q^{m\times n}$, $\boldsymbol{c}\in\mathbb{Z}_q^m$, $\boldsymbol{p}\in\mathbb{Z}_q^n$, 其中 m 可以表示成 n 的多项式函数. 对于概率舍入函数 $R:\mathbb{Z}_q\rightarrow\{0,1\}$, 满足对于 $\forall x\in\mathbb{Z}_q$,

$$p_r(x)=\Pr\big(\mathrm{R}(x)=1\big)=\sum\nolimits_j\hat{p}_r\cdot e_{j/q}(x) \tag{7}$$

其中 $j\in J$, $J\subset\mathbb{Z}_q$, $\hat{p}_r\in\mathbb{C}$. 对于 $\varepsilon=\mathrm{negl}(n)$, 令 $s\geq\eta_\varepsilon(\Lambda_{\boldsymbol{A}})$. 假设

$$\forall j\in J-\{0\},s\cdot d(j\boldsymbol{c},\Lambda_{\boldsymbol{A}})>q\sqrt{m} \tag{8}$$

令 $p(\boldsymbol{c})=\Pr(R(\boldsymbol{h}_i^{\mathrm{T}}\cdot\boldsymbol{c})=1|\boldsymbol{u}_i=\boldsymbol{B}^{\mathrm{T}}\cdot\boldsymbol{h}_i)$, 在 $\boldsymbol{u}_i=\boldsymbol{B}^{\mathrm{T}}\cdot\boldsymbol{h}$ 的条件下, $p(\boldsymbol{c})$ 由舍入函数 R 的随机性, 以及 $\boldsymbol{h}_i$ 的分布决定. 那么

$$\left|p(\boldsymbol{c})-\hat{r}_0\right|\leq(2+O(\varepsilon))|J|C^m+O(\varepsilon) \tag{9}$$

其中 $C=(2\pi\mathrm{e})^{1/2}\cdot\mathrm{e}^{-\pi}<1$. 此外, 现有文献通常直接称ASPHF为SPHF[10,13,14], 下文也如此.

3 PAKE协议的安全模型

通信模型[19] 假设通信在不安全的公开网络上进行, 参与方包括客户 $u_1,u_2,\cdots\in U$, 攻击者(或敌手) $\mathrm{ad}_1,\mathrm{ad}_2,\cdots\in B$ 和可信服务器 $s_1,s_2,\cdots\in E$.

设各参与方可与其他参与方(并发地)执行多次协议; 称一次协议的执行为一个实例(如 $\Pi_{u_1}^i$ 表示客户 u_1 的第 i 个实例). 以实例 $\Pi_{u_1}^i$ 为例, 其维持本地状态变量 $(\mathrm{sid}_{u_1}^i,\mathrm{pid}_{u_1}^i,\mathrm{sk}_{u_1}^i,\mathrm{acc}_{u_1}^i,\mathrm{term}_{u_1}^i)$, $\mathrm{sid}_{u_1}^i$ 为会话标识, $\mathrm{pid}_{u_1}^i$ 表示客体 u_1 所认为的与其共同参与协议执行的另一参与方的标识; $\mathrm{sk}_{u_1}^i$ 表示客户 u_1 所得到的会话密钥; $\mathrm{acc}_{u_1}^i$ 是标识实例 $\Pi_{u_1}^i$ 是否被客户 u_1 接受的布尔变量, $\mathrm{acc}_{u_1}^i=1$ 表示接受; $\mathrm{term}_{u_1}^i$ 是标识实例 $\Pi_{u_1}^i$ 是否被客户 u_1 拒绝的布尔变量, 拒绝用 $\mathrm{term}_{u_1}^i=1$ 标识. 下文中伙伴关系、正确性、新鲜性的定义与BRP模型[19]保持一致.

攻击者模型 攻击者 ad_1 可在协议执行过程中进行消息的窃听、拦截、注入、篡改等. 根据BPR模型[19], 本文用 $\mathrm{Execute}(u_1,\ i,\ s_1,\ j)$ 预言机、$\mathrm{Send}(u_1,\ i,\ \mathrm{msg})$ 预言机、$\mathrm{Reveal}(u_1,\ i)$ 预言机以及 $\mathrm{Test}(u_1,\ i)$ 预言机建模实例, 并用预言机询问建模攻击者的攻击能力. 各预言机的具体定义也与BRP模型[19]保持一致.

攻击者优势[11] 协议的安全性是通过一系列Game定义的: 攻击者可以执行一系列上述预言机询问(但只允许执行一次Test询问); 所有Game执行完毕后, 攻击者输出猜测比特 b'; 若 $b'=b$, 称攻击者攻击成功. 本文用Success表示攻击成功事件, ad_1 在攻击协议 Π 时的攻击优势 $\mathrm{Adv}_{\Pi,\mathrm{ad}_1}$ 为

$$\mathrm{Adv}_{\Pi,\mathrm{ad}_1}(n)=2\Pr(\mathrm{Success})-1 \tag{10}$$

定义1(安全协议) 设安全参数为 n, 口令空间为 D, 服从CDF-Zipf分布[20], 另设概率多项式时间(Probabilistic Polynomial Time, PPT)攻击者最多可执行 $Q(n)$ 次在线攻击, 称PAKE协议 Π 是安全的, 如果式(11)成立.

$$\mathrm{Adv}_{\Pi,\mathrm{ad}_1}(n)\le C'\cdot Q(n)^{s'}+\mathrm{negl}(n) \tag{11}$$

其中 $C'=0.062239$, $s'=0.155478$.

本文假设口令服从CDF-Zipf分布. 根据文献[21], CDF-Zipf分布下敌手优势的准确性,至少是均匀分布下敌手优势准确性的3~4倍. 因此,本文安全协议模型至少比均匀随机模型精确3~4倍.

4　基于密文标准语言的SPHF

4.1　基于KV密文标准语言的SPHF

为解决现有SPHF不能在超多项式模数下应用的问题,本文提出了一种基于KV密文标准语言(standard languages)的SPHF,记作KV-S-SPHF. 下面首先定义本文中基于KV密文的标准语言

$$\begin{aligned}&L_0=\left\{(\boldsymbol{p},\boldsymbol{c},u)\mid\exists \boldsymbol{s},\boldsymbol{e}_{\mathrm{KV}},\boldsymbol{c}\leftarrow\mathrm{Enc}\left(\mathrm{pk}_{\mathrm{KV}},\boldsymbol{p},u;\ \boldsymbol{s},\boldsymbol{e}_{\mathrm{KV}}\right)\right\},\\&L=\left\{(\boldsymbol{p},\boldsymbol{c},u)\mid\mathrm{Dec}\left(\mathrm{sk}_{\mathrm{KV}},\boldsymbol{c},u\right)=\boldsymbol{p}\right\}\end{aligned} \tag{12}$$

为构造基于KV密文标准语言的SPHF(记作KV-S-SPHF),舍入函数应该满足:除1次谐波外的j(j为奇数)次谐波的权重尽量为0. 本文选择的舍入函数及其波形分别如式(13)和图1所示.

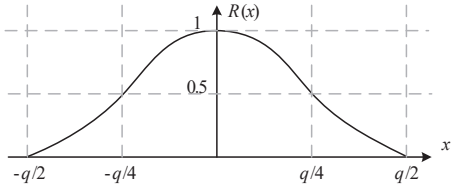

图1　舍入函数波形图

$$R(x)=1/2+1/2\cos(2\pi x/q) \tag{13}$$

本文构建的基于KV密文标准语言的SPHF(记作KV-S-SPHF)如下:

(1) hk←KV.HashKG(params):随机输入$(\boldsymbol{h}_1,\cdots,\boldsymbol{h}_k)\leftarrow(\mathbb{Z}_q^m)^k$;输出哈希密钥hk: $=(\boldsymbol{h}_1,\cdots,\boldsymbol{h}_k)\in(\mathbb{Z}_q^m)^k$.

(2) ph←KV.ProjKG(params, hk$=(\boldsymbol{h}_1,\cdots,\boldsymbol{h}_k)$, $\mathrm{pk}_{\mathrm{KV}}=\boldsymbol{B}_{\mathrm{KV}}$):输入hk$=(\boldsymbol{h}_1,\cdots,\boldsymbol{h}_k)\in(\mathbb{Z}_q^m)^k$,公钥$\boldsymbol{B}_{\mathrm{KV}}$;输出ph: $=(\boldsymbol{u}_1,\cdots,\boldsymbol{u}_k)=\alpha(\boldsymbol{h}_1,\cdots,\boldsymbol{h}_k)$,其中投射密钥$\boldsymbol{u}_i=\boldsymbol{B}_{\mathrm{KV}}^{\mathrm{T}}\cdot\boldsymbol{h}_i(1\le i\le k)$.

(3) $\boldsymbol{H}$←KV.Hash(hk$=(\boldsymbol{h}_1,\cdots,\boldsymbol{h}_k)$, $\boldsymbol{W}_{\mathrm{KV}}$: $=$(label, $\boldsymbol{c},\boldsymbol{p}$)):输入hk$=(\boldsymbol{h}_1,\cdots,\boldsymbol{h}_k)\in(\mathbb{Z}_q^m)^k$,单词$\boldsymbol{W}_{\mathrm{KV}}$: $=$(label, $\boldsymbol{c},\boldsymbol{p}$),其中$\boldsymbol{p}\in\mathbb{Z}_q^l$, $\boldsymbol{c}\in\mathbb{Z}_q^m$;哈希函数的计算方式如式(14)所示,其中$R(x)=1/2+1/2\cos(2\pi x/q)$;输出$\boldsymbol{H}=(b_1,\cdots,b_k)$.

$$\begin{cases}z_i=\mathrm{Hash}(\mathrm{hk}=\boldsymbol{h}_i,\boldsymbol{W}_{\mathrm{KV}}:=(\mathrm{label},\ \boldsymbol{c},\ \boldsymbol{p})\\\quad=\boldsymbol{h}_i^{\mathrm{T}}\cdot\left[\boldsymbol{c}-\boldsymbol{U}\cdot\begin{pmatrix}1\\\boldsymbol{p}\end{pmatrix}\right](\mathrm{mod}\ q)\in\mathbb{Z}_q\\b_i=R(z_i)\in\{0,1\}\end{cases} \tag{14}$$

(4) $\boldsymbol{P}\leftarrow$ KV.ProjHash(ph $=(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k)$, $\boldsymbol{W}_{\mathrm{KV}}=$(label, $\boldsymbol{c},\boldsymbol{p}$); $\boldsymbol{w}=\boldsymbol{e}_{\mathrm{KV}}$):输入ph$=(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k)$, $\boldsymbol{W}_{\mathrm{KV}}=$(label, $\boldsymbol{c},\boldsymbol{p}$)和$\boldsymbol{e}_{\mathrm{KV}}$;投射函数如式(15)所示;输出$\boldsymbol{P}=(b_1',\cdots,b_k')$.

$$\begin{cases}z'_i=\mathrm{ProjHash}(\mathrm{ph}=(\boldsymbol{u}_1,\cdots,\boldsymbol{u}_k),\\\quad\boldsymbol{W}_{\mathrm{KV}}:=(\mathrm{label},\boldsymbol{c},\boldsymbol{p});\boldsymbol{w}=\boldsymbol{e}_{\mathrm{KV}})\\\quad=\boldsymbol{u}_i^{\mathrm{T}}\cdot\boldsymbol{e}_{\mathrm{KV}}(\mathrm{mod}\ q)\in\mathbb{Z}_q\\b'_i=R(z'_i)\in\{0,1\}\end{cases} \tag{15}$$

下面证明KV-S-SPHF满足正确性与平滑性.

证明　(1) 正确性证明

对于$\forall\boldsymbol{W}_{\mathrm{KV}}\in L$,满足

$$\begin{aligned}\boldsymbol{p}_1&=\Pr(\boldsymbol{H}=\boldsymbol{P})\\&=\Pr\begin{pmatrix}\mathrm{KV.Hash}\left((\boldsymbol{h}_1,\cdots,\boldsymbol{h}_k),(\mathrm{label},\boldsymbol{c},\boldsymbol{p})\right)=\\\mathrm{KV.ProjHash}\left((\boldsymbol{u}_1,\cdots,\boldsymbol{u}_k),(\mathrm{label},\boldsymbol{c},\boldsymbol{p});\boldsymbol{e}_{\mathrm{KV}}\right)\end{pmatrix}\\&>\frac{1}{2}+\Delta\end{aligned} \tag{16}$$

其中,Δ是不可忽略的小数. 因为在构建KV-S-SPHF中使用了舍入函数,只要保证式(17)成立即可.

$$\begin{aligned}\boldsymbol{p}_1&=\Pr\left(R\left(\boldsymbol{h}_i^{\mathrm{T}}\left[\boldsymbol{c}-\boldsymbol{U}\begin{pmatrix}1\\\boldsymbol{p}\end{pmatrix}\right]\right)=R\left(\boldsymbol{u}_i^{\mathrm{T}}\cdot\boldsymbol{e}_{\mathrm{KV}}\right)\right)\\&>\frac{1}{2}+\Delta\end{aligned} \tag{17}$$

进一步,根据式(13)和式(17),可得式(18).

$$\begin{aligned}p_1&=\frac{l}{q}\sum_{x\in\mathbb{Z}_q}(R(x)R(x+\boldsymbol{h}_i^{\mathrm{T}}\cdot\boldsymbol{c})+(1-R(x))(1-R(x+\boldsymbol{h}_i^{\mathrm{T}}\cdot\boldsymbol{c})))+\mathrm{negl}(n)\\&=\frac{1}{2}+\frac{1}{q}\sum_{x\in\mathbb{Z}_q}\frac{1}{2}\cos\left(2\pi\frac{x}{q}\right)\cos\left(2\pi\frac{x+\boldsymbol{h}_i^{\mathrm{T}}\cdot\boldsymbol{c}}{q}\right)+\mathrm{negl}(n)\end{aligned} \tag{18}$$

因为$t\cdot s\cdot m=o(q)$,所以有$\boldsymbol{h}_i^{\mathrm{T}}\boldsymbol{c}=o(q)$($\boldsymbol{c}$是高斯取样的误差)在统计上成立. 余弦函数是Lipschitz连续函数,因此可通过积分来近似求和:

$$\begin{aligned}\boldsymbol{p}_1&=\frac{1}{2}+\frac{1}{2}\int_0^1\cos^2(2\pi x)\,\mathrm{d}x+o(1)\\&=\frac{3}{4}+o(1)\end{aligned} \tag{19}$$

根据式(16)和式(19)可知,KV-S-SPHF满足近似正确性,根据文献[11]中的正确性放大技术,我们可以得到一个具有统计正确性的KV-S-SPHF.

(2) 平滑性证明

本文引入新舍入函数后,只要保证对于$\forall\boldsymbol{W}_{\mathrm{KV}}\in X-L$,式(20)中的$p_2$满足$\left|p_2-1/2\right|\le\mathrm{negl}(n)$,那么就可以保证KV-S-SPHF的平滑性.

$$p_2=\Pr\left(R(z_i(\bmod q))=1\,\middle|\,\boldsymbol{u}_i=\boldsymbol{B}^{\mathrm{T}}\cdot\boldsymbol{h}_i\right)$$
$$=\Pr\left(R\left(\boldsymbol{h}_i^{\mathrm{T}}\cdot\left[\boldsymbol{c}-\boldsymbol{U}\cdot\begin{pmatrix}1\\\boldsymbol{p}\end{pmatrix}\right](\bmod q)\right)=1\,\middle|\,\boldsymbol{u}_i=\boldsymbol{B}^{\mathrm{T}}\cdot\boldsymbol{h}_i\right)\tag{20}$$

令 $p_R(x)=\Pr(R(x(\bmod q))=1)$,那么 $p_R(x)$ 是一个以 q 为周期的周期信号. 现对 $p_R(z_i)$ 进行插值处理,得到式(21).

$$p_R(z_i)=\sum_{j\in\mathbb{Z}_q}\hat{p}_{r',j}\cdot e_{j/q}(z_i)\tag{21}$$

其中 $\hat{p}_{r'}\in\mathbb{C}$ 是 $p_{r'}:\mathbb{Z}\to[0,1]$ 的傅里叶系数. 令 $s\geq\eta_\varepsilon(\Lambda_A^\perp)$,并假设 $\forall j\in J-\{0\}$

$$s\times d\left(j\left[\boldsymbol{c}-\boldsymbol{U}\begin{pmatrix}1\\\boldsymbol{p}\end{pmatrix}\right],\Lambda_A\right)>q\sqrt{m}\tag{22}$$

根据式(13),有式(23).

$$\begin{cases}\hat{p}_{r,0}=1/2\\\hat{p}_{r,\pm1}=1/2\\\hat{p}_{r,j}=0\ (j\neq0,\pm1)\end{cases}\tag{23}$$

根据定理1及式(19)~(23),对于 $\forall\boldsymbol{W}_{\mathrm{KV}}\in X-L$, $C=(2\pi\mathrm{e})^{1/2}\cdot\mathrm{e}^{-\pi}<1$,有

$$\left|2p_2-1\right|\leq2(6+O(\varepsilon))C^m+O(\varepsilon)\leq\mathrm{negl}(n)\tag{24}$$

综上,KV-S-SPHF是平滑投射哈希函数.

证毕.

4.2 基于GPV密文标准语言的SPHF

本小节研究基于GPV密文标准语言的SPHF,记作GVP-S-SPHF. GPV方案的公共矩阵为 $\boldsymbol{A}$,公钥为 $\mathrm{pk}_{\mathrm{GPV}}=\boldsymbol{B}_{\mathrm{GPV}}=\boldsymbol{As}+\boldsymbol{x}\in\mathbb{Z}_q^m$,具体方案如下:

(1) $\mathrm{hk}\leftarrow$GPV.HashKG(params):随机输入 $(\boldsymbol{h}_1,\cdots,\boldsymbol{h}_k)\leftarrow(\mathbb{Z}_q^m)^k$;输出哈希密钥 $\mathrm{hk}:=(\boldsymbol{h}_1,\cdots,\boldsymbol{h}_k)\in(\mathbb{Z}_q^m)^k$.

(2) $\mathrm{ph}\leftarrow$GPV.ProjKG(params, $\mathrm{hk}=(\boldsymbol{h}_1,\cdots,\boldsymbol{h}_k)$, $\mathrm{pk}_{\mathrm{KV}}=\boldsymbol{B}_{\mathrm{GPV}}$):输入 $\mathrm{hk}=(\boldsymbol{h}_1,\cdots,\boldsymbol{h}_k)\in(\mathbb{Z}_q^m)^k$, $\mathrm{pk}_{\mathrm{KV}}=\boldsymbol{B}_{\mathrm{GPV}}$;输出 $\mathrm{ph}:=(\boldsymbol{u}_1,\cdots,\boldsymbol{u}_k)=\alpha(\boldsymbol{h}_1,\cdots,\boldsymbol{h}_k)$,其中投射密钥 $\boldsymbol{u}_i=\boldsymbol{B}_{\mathrm{GPV}}^{\mathrm{T}}\cdot\boldsymbol{h}_i(1\leq i\leq k)$.

(3) $\boldsymbol{H}\leftarrow$GPV.Hash($\mathrm{hk}=(\boldsymbol{h}_1,\cdots,\boldsymbol{h}_k)$, $\boldsymbol{W}_{\mathrm{GPV}}:=(\boldsymbol{c},\boldsymbol{p})$):输入 $\mathrm{hk}=(\boldsymbol{h}_1,\cdots,\boldsymbol{h}_k)\in(\mathbb{Z}_q^m)^k$,单词 $\boldsymbol{W}_{\mathrm{GPV}}=(\boldsymbol{c},\boldsymbol{p})$,其中 $\boldsymbol{c}=(c_1,\cdots,c_k)\in\mathbb{Z}_q^k$;哈希函数的计算方式如式(25)所示,其中 $R(x)=1/2+1/2\cos(2\pi x/q)$;输出 $\boldsymbol{H}=(b_1,\cdots,b_k)$.

$$\begin{cases}z_i=\mathrm{Hash}(\mathrm{hk}=\boldsymbol{h}_i,\boldsymbol{W}_{\mathrm{GPV}}:=(c_i,p_i))\\\quad=\left\|\boldsymbol{h}_i^{\mathrm{T}}\cdot[c_i-(\lfloor q/2\rfloor\cdot p_i)]\right\|\\\quad(\bmod q)\in\mathbb{Z}_q\\b_i=R(z_i)\in\{0,1\}\end{cases}\tag{25}$$

(4) $\boldsymbol{P}\leftarrow$GPV.ProjHash($\mathrm{ph}=(\boldsymbol{u}_1,\cdots,\boldsymbol{u}_k)$, $\boldsymbol{W}_{\mathrm{GPV}}=(\boldsymbol{c},\boldsymbol{p})$; $\boldsymbol{w}=\boldsymbol{e}_{\mathrm{GPV}}$):输入投射密钥 $\mathrm{ph}=(\boldsymbol{u}_1,\cdots,\boldsymbol{u}_k)$, $\boldsymbol{W}_{\mathrm{GPV}}=(\boldsymbol{c},\boldsymbol{p})$ 和 $\boldsymbol{w}=\boldsymbol{e}_{\mathrm{GPV}}$;其中 $\boldsymbol{p}=(p_1,\cdots,p_k)$, $\boldsymbol{c}=(c_1,\cdots,c_k)\in\mathbb{Z}_q^k$;投射函数的计算方式如式(26)所示;输出 $\boldsymbol{P}=(b_1',\cdots,b_k')$.

$$\begin{cases}z'_i=\mathrm{ProjHash}(\mathrm{ph}=(\boldsymbol{u}_1,\cdots,\boldsymbol{u}_k),\\\quad\boldsymbol{W}_{\mathrm{GPV}}:=(\boldsymbol{c},\boldsymbol{p});\ \boldsymbol{w}=\boldsymbol{e}_{\mathrm{GPV}})\\\quad=\left\|\boldsymbol{u}_i\cdot\boldsymbol{e}_{\mathrm{GPV}}\right\|(\bmod q)\in\mathbb{Z}_q\\b'_i=R(z'_i)\in\{0,1\}\end{cases}\tag{26}$$

GPV-S-SPHF的正确性与平滑性证明过程与KV-S-SPHF类似,不再赘述.

5 格上可证明安全的两轮PAKE

5.1 两轮协议及正确性分析

基于本文提出的两种平滑投射哈希函数,本节提出了一种格上基于标准安全模型的不需要SS-NIZK的二轮PAKE,如算法1所示.

密码原语 (1) KV方案及KV-S-SPHF;(2) GPV方案及GPV-S-SPHF.

初始化阶段 建立KV和GPV方案的公钥,即公共参考序列(Common Reference String,CRS).

协议执行过程 假设协议在客户 u_1 和服务器 s_1 之间执行,u_1 和 s_1 共享口令 pw_{u_1}.

u_1 选择 $\boldsymbol{e}_{\mathrm{GPV}}\leftarrow D_{\mathbb{Z}^m,\boldsymbol{y}}$,哈希密钥 $\mathrm{hk}_{u_1}\xleftarrow{\mathrm{r}}$GPV.HashKG(params);计算投射密钥 $\mathrm{ph}_{u_1}\leftarrow$GPV.ProjKG(params, hk_{u_1}, $\mathrm{pk}_{\mathrm{GPV}}$);将己方标签设置为 $\mathrm{label}=u_1\|s_1\|\mathrm{ph}_{u_1}$;并计算己方密文 $(\boldsymbol{u},\boldsymbol{c}_{u_1})\leftarrow$GPV.Enc($\mathrm{pk}_{\mathrm{GPV}}$, pw_{u_1}, label; $\boldsymbol{e}_{\mathrm{GPV}}$);最后向 s_1 发送 $u_1\|\boldsymbol{c}_{u_1}\|\mathrm{ph}_{u_1}$.

s_1 收到消息 $u_1\|\boldsymbol{c}_{u_1}\|\mathrm{ph}_{u_1}$ 后,选择 $\boldsymbol{e}_{\mathrm{KV}}\xleftarrow{\mathrm{r}}(\mathbb{Z}_q)^n$,哈希密钥 $\mathrm{hk}_{s_1}\xleftarrow{\mathrm{r}}$KV.HashKG(params);计算投射密钥 $\mathrm{ph}_{s_1}\leftarrow$KV.ProjKG(params, hk_{s_1}, $\mathrm{pk}_{\mathrm{KV}}$);设置客户方标签为 $\mathrm{label}=u_1\|s_1\|\mathrm{ph}_{u_1}$,己方标签为 $\mathrm{label}'=u_1\|s_1\|\mathrm{ph}_{u_1}\|\boldsymbol{c}_{u_1}\|\mathrm{ph}_{s_1}$;并计算己方密文 $\boldsymbol{c}_{s_1}\leftarrow$KV.Enc$^{\mathrm{label}'}$($\mathrm{pk}_{\mathrm{KV}}$, pw_{s_1,u_1}; $\boldsymbol{e}_{\mathrm{KV}}$);最后向 u_1 发送 $s_1\|\boldsymbol{c}_{s_1}\|\mathrm{ph}_{s_1}$.

u_1 收到 $s_1\|\boldsymbol{c}_{s_1}\|\mathrm{ph}_{s_1}$ 后,计算 $\boldsymbol{H}_{u_1}\leftarrow$KV.Hash($\mathrm{hk}_{u_1}$, $\boldsymbol{W}_{\mathrm{KV}}=(\mathrm{label}',\boldsymbol{c}_{s_1},\mathrm{pw}_{u_1})$) 和 $\boldsymbol{P}_{u_1}\leftarrow$GVP.ProjHash($\mathrm{ph}_{s_1}$, $(\boldsymbol{c}_{u_1},\mathrm{pw}_{u_1})$; $\boldsymbol{e}_{\mathrm{GPV}}$);最后计算会话密钥 $\boldsymbol{K}_{u_1}\leftarrow\boldsymbol{H}_{u_1}\cdot\boldsymbol{P}_{u_1}$. s_1 计算会话密钥的方式与 u_1 类似,但无需等待 u_1 收到 $s_1\|\boldsymbol{c}_{s_1}\|\mathrm{ph}_{s_1}$.

正确性分析 根据式(27)以及SPHF的平滑性(见式(6)),本文协议满足正确性.

算法 1　格上基于标准安全模型的不需要 SS-NIZK 的二轮 PAKE 协议

客户 $u_1(\mathrm{pw}_{u_1})$	CRS: $\mathrm{pk}_{\mathrm{KV}}, \mathrm{pk}_{\mathrm{GPV}}$	服务器 $s_1(\mathrm{pw}_{s_1,u_1})$
(1) $\mathrm{hk}_{u_1} \xleftarrow{r} \mathrm{GPV.HashKG}(\mathrm{params}), \boldsymbol{e}_{\mathrm{GPV}} \leftarrow D_{\mathbb{Z}^m,y}$		
(2) $\mathrm{ph}_{u_1} \leftarrow \mathrm{GPV.ProjKG}(\mathrm{params}, \mathrm{hk}_{u_1}, \mathrm{pk}_{\mathrm{GPV}})$		
(3) $\mathrm{label} = u_1 \Vert s_1 \Vert \mathrm{ph}_{u_1}$		
(4) $(\boldsymbol{u}, \boldsymbol{c}_{u_1}) \leftarrow \mathrm{GPV.Enc}(\mathrm{pk}_{\mathrm{GPV}}, \mathrm{pw}_{u_1}, \mathrm{label}; \boldsymbol{e}_{\mathrm{GPV}})$		
	$u_1 \Vert \boldsymbol{c}_{u_1} \Vert \mathrm{ph}_{u_1} \xrightarrow{}$	
		(1) $\mathrm{hk}_{s_1} \xleftarrow{r} \mathrm{KV.HashKG}(\mathrm{params}), \boldsymbol{e}_{\mathrm{KV}} \xleftarrow{r} (\mathbb{Z}_q)^n$
		(2) $\mathrm{ph}_{s_1} \leftarrow \mathrm{KV.ProjKG}(\mathrm{params}, \mathrm{hk}_{s_1}, \mathrm{pk}_{\mathrm{KV}})$
		(3) $\mathrm{label} = u_1 \Vert s_1 \Vert \mathrm{ph}_{u_1}, \mathrm{label}' = u_1 \Vert s_1 \Vert \mathrm{ph}_{u_1} \Vert \boldsymbol{c}_{u_1} \Vert \mathrm{ph}_{s_1}$
		(4) $\boldsymbol{c}_{s_1} \leftarrow \mathrm{KV.Enc}^{\mathrm{label}'}(\mathrm{pk}_{\mathrm{KV}}, \mathrm{pw}_{s_1,u_1}; \boldsymbol{e}_{\mathrm{KV}})$
	$s_1 \Vert c_{s1} \Vert ph_{s1} \xleftarrow{}$	
(5) $\mathrm{label}' = u_1 \Vert s_1 \Vert \mathrm{ph}_{u_1} \Vert \boldsymbol{c}_{u_1} \Vert \mathrm{ph}_{s_1}$		(5) $\boldsymbol{H}_{s_1} \leftarrow \mathrm{GPV.Hash}(\mathrm{hk}_{s_1}, \boldsymbol{W}_{\mathrm{GPV}} := (\boldsymbol{c}_{u_1}, \mathrm{pw}_{s_1,u_1}))$
(6) $\boldsymbol{H}_{u_1} \leftarrow \mathrm{KV.Hash}(\mathrm{hk}_{u_1}, \boldsymbol{W}_{\mathrm{KV}} := (\mathrm{label}', \boldsymbol{c}_{s_1}, \mathrm{pw}_{u_1}))$		(6) $\boldsymbol{P}_{s_1} \leftarrow \mathrm{KV.ProjHash}(\mathrm{ph}_{u_1}, \boldsymbol{W}_{\mathrm{KV}} := (\mathrm{label}', \boldsymbol{c}_{s_1}, \mathrm{pw}_{s_1,u_1}); \boldsymbol{e}_{\mathrm{KV}})$
(7) $\boldsymbol{P}_{u_1} \leftarrow \mathrm{GVP.ProjHash}(\mathrm{ph}_{s_1}, \boldsymbol{W}_{\mathrm{GPV}} := (\boldsymbol{c}_{u_1}, \mathrm{pw}_{u_1}); \boldsymbol{e}_{\mathrm{GPV}})$		(7) $\boldsymbol{K}_{s_1} \leftarrow \boldsymbol{H}_{s_1} \cdot \boldsymbol{P}_{s_1}$
(8) $\boldsymbol{K}_{u_1} \leftarrow \boldsymbol{H}_{u_1} \cdot \boldsymbol{P}_{u_1}$		
(9) 删除除 pw_{u_1} 和 $\boldsymbol{K}_{u_1}$ 外所有存储信息		(8) 删除除 pw_{s_1} 和 $\boldsymbol{K}_{s_1}$ 外所有存储信息

$$\begin{aligned}\boldsymbol{K}_{u_1} &= \boldsymbol{H}_{u_1} \cdot \boldsymbol{P}_{u_1} \\ &= \mathrm{KV.Hash}(\mathrm{hk}_{u_1}, \boldsymbol{W}_{\mathrm{KV}} := (\mathrm{label}', \boldsymbol{c}_{s_1}, \mathrm{pw}_{u_1})) \cdot \\ &\mathrm{GVP.ProjHash}(\mathrm{ph}_{s_1}, \boldsymbol{W}_{\mathrm{GPV}} := (\boldsymbol{c}_{u_1}, \mathrm{pw}_{u_1}); \boldsymbol{e}_{\mathrm{GPV}}) \\ &= \mathrm{KV.ProjHash}(\mathrm{ph}_{u_1}, \boldsymbol{W}_{\mathrm{KV}} := (\mathrm{label}', \boldsymbol{c}_{s_1}, \mathrm{pw}_{s_1,u_1}); \\ &\boldsymbol{e}_{\mathrm{KV}}) \cdot \mathrm{GPV.Hash}(\mathrm{hk}_{s_1}, \boldsymbol{W}_{\mathrm{GPV}} := (\boldsymbol{c}_{u_1}, \mathrm{pw}_{s_1,u_1})) \\ &= \boldsymbol{P}_{s_1} \cdot \boldsymbol{H}_{s_1} = \boldsymbol{H}_{s_1} \cdot \boldsymbol{P}_{s_1} = \boldsymbol{K}_{s_1}\end{aligned} \tag{27}$$

5.2　安全性证明

本文通过证明定理 2,来证明本文 PAKE 协议是一个安全的 PAKE 协议.

定理 2　如果 (1) KV 方案是 IND-CCA2 安全的 PKE,且 KV-S-SPHF 是 SPHF;(2) GPV 方案是 IND-CPA 安全的 PKE,且 GPV-S-SPHF 是 SPHF;那么算法 1 中的 PAKE 是一个安全的 PAKE.

假设 PPT 攻击者 (ad_1) 对算法 1 所示的协议进行攻击,本文通过 Game0, Game1, Game2, …, 来评估攻击者的优势,其中 Game0 与真实攻击一致. 在 Game$i$ 中,用 $\mathrm{Success}_i$ 表示攻击成功事件,ad_1 的优势为 $\mathrm{Adv}_{\mathrm{ad}_1,i} = 2\Pr[\mathrm{Success}_{\mathrm{Game}i}] - 1$. 为方便表述,令 $\mathrm{msg1} = u_1 \Vert \boldsymbol{c}_{u_1} \Vert \mathrm{ph}_{u_1}$, $\mathrm{msg2} = s_1 \Vert \boldsymbol{c}_{s_1} \Vert \mathrm{ph}_{s_1}$. 注意模拟器已知参与双方的私有输入,每次 Game 只对其前一个 Game 进行修改.

Game1　修改 Execute 预言机,若 $\mathrm{pw}_{u_1} = \mathrm{pw}_{s_1,u_1}$,将服务器与用户会话密钥的计算方式替换为 $\mathrm{KV.Hash}(\mathrm{hk}_{u_1}, \boldsymbol{W} = (\mathrm{label}', \boldsymbol{c}_{s_1}, \mathrm{pw}_{u_1})) \cdot \mathrm{GPV.Hash}(\mathrm{hk}_{s_1}, \boldsymbol{W} = (\boldsymbol{c}_{u_1}, \mathrm{pw}_{s_1,u_1}))$. 下证

$$\left| \mathrm{Adv}_{\mathrm{ad}_1,1}(n) - \mathrm{Adv}_{\mathrm{ad}_1,0}(n) \right| \le \mathrm{negl}(n) \tag{28}$$

证明　根据 SPHF 的正确性,式 (28) 成立.

证毕.

Game2　修改 Execute 预言机,用服从 Zipf 分布的非法口令 pw'_{u_1} 来生成 $\boldsymbol{c}_{u_1}$. 下证

$$\left| \mathrm{Adv}_{\mathrm{ad}_1,2}(n) - \mathrm{Adv}_{\mathrm{ad}_1,1}(n) \right| \le \mathrm{negl}(n) \tag{29}$$

证明　构建一个 PPT 攻击者 ad_2 攻击 GPV 方案:给定公钥 $\mathrm{pk}_{\mathrm{GPV}}$, ad_2 将 $(\mathrm{pw}_{u_1}, \mathrm{pw}'_{u_1})$ 发送给其自己的“挑战”预言机;当收到挑战密文 $\boldsymbol{c}'_{u_1}$ 后,ad_2 用 $\boldsymbol{c}'_{u_1}$ 构建 msg1,并发送给 ad_1.

令 $\mathrm{Event}_{\mathrm{GPV}}^{\mathrm{pw}_u}(\mathrm{ad}_2)$ 表示事件:ad_2 获得的挑战密文是由真实口令生成的,且 ad_2 输出 1. 令 $\mathrm{Event}_{\mathrm{GPV}}^{\mathrm{pw}'_u}(\mathrm{ad}_2)$ 表示事件:ad_2 获得的挑战密文是由非法口令生成的,且 ad_2 输出 1. 如果 ad_1 攻击成功 (即 $b' = b$),那么 ad_2 输出 1;否则 ad_2 输出 0. 因此“ad_2 获得的挑战密文是由真实口令产生的且输出 1”的概率,与 ad_1 攻击成功的概率是相同的 ($\Pr\left(\mathrm{Event}_{\mathrm{GPV}}^{\mathrm{pw}_u}(\mathrm{ad}_2) = 1\right) = \Pr\left(\mathrm{Success}_{\mathrm{Game1}}\right)$). 同理 $\Pr\left(\mathrm{Event}_{\mathrm{GPV}}^{\mathrm{pw}'_u}(\mathrm{ad}_2) = 1\right) = \Pr\left(\mathrm{Success}_{\mathrm{Game2}}\right)$. 令 ad_2 攻击

GPV的优势为$\mathrm{Adv}_{\mathrm{ad}_2,\mathrm{GPV}}^{\mathrm{IND-CPA}}(n)$,那么

$$\begin{aligned}&\left|\mathrm{Adv}_{\mathrm{ad}_1,2}(n)-\mathrm{Adv}_{\mathrm{ad}_1,1}(n)\right|\\&=2\left|\Pr\left(\mathrm{Success}_{\mathrm{Game2}}\right)-\Pr\left(\mathrm{Success}_{\mathrm{Game1}}\right)\right|\\&=2\left|\Pr\left(\mathrm{Event}_{\mathrm{GPV}^1}^{\mathrm{pw}_u}(\mathrm{ad}_2)=1\right)-\Pr\left(\mathrm{Event}_{\mathrm{GPV}^1}^{\mathrm{pw}'_u}(\mathrm{ad}_2)=1\right)\right|\\&=2\mathrm{Adv}_{\mathrm{ad}_2,\mathrm{GPV}}^{\mathrm{IND-CPA}}(n)\le\mathrm{negl}(n)\end{aligned}\tag{30}$$

根据GPV方案的IND-CPA安全性,式(29)成立.

证毕.

Game3 修改Execute预言机,若双方持有相同的口令,用服从Zipf分布的非法口令$\mathrm{pw}'_{s_1,u_1}=\mathrm{pw}'_{u_1}$来生成$\boldsymbol{c}_{s1}$. 下证

$$\left|\mathrm{Adv}_{\mathrm{ad}_1,3}(n)-\mathrm{Adv}_{\mathrm{ad}_1,2}(n)\right|\le\mathrm{negl}(n)\tag{31}$$

该证明同Game2类似,不再赘述.

Game4 修改Execute预言机,若双方持有相同的口令,将双方的会话密钥替换为相互独立的随机数. 下证

$$\left|\mathrm{Adv}_{\mathrm{ad}_1,4}(n)-\mathrm{Adv}_{\mathrm{ad}_1,3}(n)\right|\le\mathrm{negl}(n)\tag{32}$$

证明 因为双方都使用非法密钥计算哈希值,所以$(\mathrm{label}',\boldsymbol{c}_{s_1},\mathrm{pw}_{u_1})\notin L_{\mathrm{KV}}$且$(\boldsymbol{c}_{u_1},\mathrm{pw}_{s_1,u_1})\notin L_{\mathrm{GPV}}$. 根据KV-S-SPHF和GPV-S-SPHF的平滑性,易知式(32)成立.

证毕.

Game5 修改Execute预言机,若$\mathrm{pw}_{s_1,u_1}\neq\mathrm{pw}_{u_1}$,那么将协议参与双方的会话密钥替换为相互独立的随机数. 下证

$$\left|\mathrm{Adv}_{\mathrm{ad}_1,5}(n)-\mathrm{Adv}_{\mathrm{ad}_1,4}(n)\right|\le\mathrm{negl}(n)\tag{33}$$

证明 根据KV-S-SPHF和GPV-S-SPHF的平滑性,易知式(33)成立.

证毕.

Game6 修改Execute,若$\mathrm{pw}_{s_1,u_1}\neq\mathrm{pw}_{u_1}$,那么将$\boldsymbol{c}_{s_1}$和$\boldsymbol{c}_{u_1}$分别替换为相互独立的非法口令$\mathrm{pw}'_{s_1,u_1}$和$\mathrm{pw}'_{u_1}$(满足Zipf分布)的加密值. 下证

$$\left|\mathrm{Adv}_{\mathrm{ad}_1,6}(n)-\mathrm{Adv}_{\mathrm{ad}_1,5}(n)\right|\le\mathrm{negl}(n)\tag{34}$$

该证明与Game2类似,不再赘述.

在Game6中,对于Execute预言机询问而言,所有的会话密钥以及传输消息都是随机的,与真实口令无关.

下文开始修改Send预言机,并将Send预言机分为以下三种:(1) $\mathrm{Send}_0(u_i,\ \mathrm{Start})$:攻击者$\mathrm{ad}_1$初始化$u_i$与$s_j$之间的协议;模拟器向$\mathrm{ad}_1$返回msg1. (2) $\mathrm{Send}_1(s_j,\ \mathrm{msg1})$:$\mathrm{ad}_1$向$s_j$发送msg1,返回给攻击者msg2,并设置$s_j$的会话密钥$\boldsymbol{K}_{s_j}$. (3) $\mathrm{Send}_2(u_i,\ \mathrm{msg2})$:$\mathrm{ad}_1$向$u_i$发送msg2;该预言机不向$\mathrm{ad}_1$返回任何消息,只是设置$u_i$的会话密钥$\boldsymbol{K}_{u_i}$.

本文用$\mathrm{sk}_{\mathrm{KV}}$表示KV方案的私钥,用$\mathrm{sk}_{\mathrm{GPV}}$表示GPV方案的私钥,注意协议执行本身并不需要私钥.

Game7 如果Execute产生的某个msg1与Send_0预言机产生的某msg1发生了碰撞,那么直接宣布ad_1攻击成功. 该变化只会增加ad_1的优势,因此$\mathrm{Adv}_{\mathrm{ad}_1,6}(n)\le\mathrm{Adv}_{\mathrm{ad}_1,7}(n)$.

Game8 修改Send_1,若msg1是由预言机产生的,那么Game8与Game7保持一致. 否则设置$\mathrm{label}=u_1\|s_1\|\mathrm{ph}_{u_1}$,并用$\mathrm{sk}_{\mathrm{GPV}}$对$\boldsymbol{c}_{u_1}$解密得到$\mathrm{pw}_{\mathrm{ad}}$,此时可能出现以下两种情况:(1) 若$\mathrm{pw}_{\mathrm{ad}}=\mathrm{pw}_{u_1}$,宣布$\mathrm{ad}_1$攻击成功并中止Game;(2) 若$\mathrm{pw}_{\mathrm{ad}}\neq\mathrm{pw}_{u_1}$,随机设置$\boldsymbol{H}_{s_1}$以及$\boldsymbol{P}_{u_1}$. 下证

$$\mathrm{Adv}_{\mathrm{ad}_1,7}(n)\le\mathrm{Adv}_{\mathrm{ad}_1,8}(n)+\mathrm{negl}(n)\tag{35}$$

证明 考虑msg1不是由预言机产生的情况:情况(1)只会增加ad_1的优势;根据SPHF的平滑性,情况(2)变化前后,ad_1所观察到分布是统计上不可区分的,所以式(35)成立.

证毕.

Game9 修改Send_2,令mgs1是由预言机产生的,若msg2也是由预言机产生的,那么可能会出现以下两种情况:(1) 若u_1和s_1持有相同的口令,令$\boldsymbol{K}_{u_1}=\boldsymbol{K}_{s_1}$;(2) 否则,为$u_1$随机选择一个会话密钥$\boldsymbol{K}_{u_1}$.

若msg2不是由预言机产生的,设置$\mathrm{label}'=u_1\|s_1\|\mathrm{ph}_{u_1}\|\boldsymbol{c}_{u_1}\|\mathrm{ph}_{s_1}$,并根据$\mathrm{sk}_{\mathrm{KV}}$对$\boldsymbol{c}_{s_1}$进行解密,此时可能出现另外两种情况:(3) 如果$\mathrm{pw}_{\mathrm{ad}}=\mathrm{pw}_{s_1,u_1}$,直接宣布$\mathrm{ad}_1$成功并中止Game;(4) 如果$\mathrm{pw}_{\mathrm{ad}}\neq\mathrm{pw}_{s_1,u_1}$,将$\boldsymbol{H}_{s_1}$以及$\boldsymbol{P}_{u_1}$设置为随机数. 下证

$$\mathrm{Adv}_{\mathrm{ad}_1,8}(n)\le\mathrm{Adv}_{\mathrm{ad}_1,9}(n)+\mathrm{negl}(n)\tag{36}$$

证明 如果msg1和msg2是由预言机产生的:上述情况(1)与Game8保持一致;对于上述情况(2),根据SPHF的平滑性,该变化只会带来可忽略的攻击者优势的变化;若msg2不是预言机产生的:上述情况(3)只会增加ad_1优势;根据KV-S-SPHF的平滑性,上述情况(4)只会带来可忽略的优势变化. 综上,式(36)成立.

证毕.

Game10 修改Send_0预言机,用非法口令pw'_{u_1}生成$\boldsymbol{c}_{u_1}$. 下证

$$\left|\mathrm{Adv}_{\mathrm{ad}_1,10}(n)-\mathrm{Adv}_{\mathrm{ad}_1,9}(n)\right|\le \mathrm{negl}(n) \tag{37}$$

证明方式同Game2,不再赘述.

Game11　修改 Send_1 预言机,用非法口令 pw'_{s_1,u_1} 生成 $\boldsymbol{c}_{s_1}$. 下证

$$\left|\mathrm{Adv}_{\mathrm{ad}_1,11}(n)-\mathrm{Adv}_{\mathrm{ad}_1,10}(n)\right|\le \mathrm{negl}(n) \tag{38}$$

该证明方法与Game2类似,但 ad_2 要检测 ad_1 是否构造了合法的msg1和msg2. 因为 ad_2 只能通其自己的"挑战"预言机判断msg2的合法性,所以该证明依赖于KV方案的IND-CCA2安全性.

下证 $\mathrm{Adv}_{\mathrm{ad}_1,0}(n)\le C'\cdot Q(n)^{s'}+\mathrm{negl}(n)$.

证明　记攻击者 ad_1 猜测出正确口令为事件Guess,攻击者 ad_1 未能猜测出正确口令为事件NGuess,则

$$\begin{aligned}&\Pr\left(\mathrm{Success}_{\mathrm{Game11}}\right)\\&=\Pr\left(\mathrm{Success}_{\mathrm{Game11}}|\ \mathrm{NGuess}\right)\cdot\Pr(\mathrm{NGuess})\\&\quad+\Pr\left(\mathrm{Success}_{\mathrm{Game11}}|\ \mathrm{Guess}\right)\cdot\Pr(\mathrm{Guess})\\&\le\Pr\left(\mathrm{Success}_{\mathrm{Game11}}|\ \mathrm{NGuess}\right)\\&\quad+\Pr\left(\mathrm{Success}_{\mathrm{Game11}}|\ \mathrm{Guess}\right)\cdot\Pr(\mathrm{Guess})\\&=\Pr\left(\mathrm{Success}_{\mathrm{Game11}}|\ \mathrm{NGuess}\right)\\&\quad+\left(1-\Pr\left(\mathrm{Success}_{\mathrm{Game11}}|\ \mathrm{NGuess}\right)\right)\cdot\Pr(\mathrm{Guess})\end{aligned} \tag{39}$$

在Game11中,所有的口令都已经被替换为服从Zipf分布的随机数,所以

$$\Pr[\mathrm{Success}_{\mathrm{Game11}}|\mathrm{Guess}]\le C'\cdot Q(n)^{s'}+\mathrm{negl}(n) \tag{40}$$

若攻击者没有猜测出正确口令,那么其只能通过比特猜测取胜,又在Game11中,会话密钥已经替换为随机数,所以

$$\Pr[\mathrm{Success}_{\mathrm{Game}_{11}}|\mathrm{NGuess}]=1/2\pm\mathrm{negl}(n) \tag{41}$$

根据式(10)、式(39)、式(40)和式(41)可知,

$$\begin{aligned}\mathrm{Adv}_{\mathrm{ad}_1,11}(n)&=2\Pr[\mathrm{Success}_{\mathrm{Game}_{11}}]-1\\&\le 2(\Pr[\mathrm{Success}_{\mathrm{Game11}}|\mathrm{NGuess}]+(1-\\&\quad\Pr[\mathrm{Success}_{\mathrm{Game11}}|\mathrm{NGuess}])\cdot\Pr[\mathrm{Guess}])\\&\le C'\cdot Q(n)^{s'}+\mathrm{negl}(n)\end{aligned} \tag{42}$$

又根据式(28)~(38),及式(42)可知

$$\begin{aligned}\mathrm{Adv}_{\mathrm{ad}_1,0}(n)&\le\mathrm{Adv}_{\mathrm{ad}_1,11}(n)+\mathrm{negl}(n)\\&\le C'\cdot Q(n)^{s'}+\mathrm{negl}(n)\end{aligned} \tag{43}$$

综上,定理2成立.

证毕.

6　协议仿真与性能评估

6.1　协议仿真与效率评估

本文在Intel(R)Core(TM)i5-4590平台上对本文提出的协议以及其他相关协议进行仿真. 目标平台为单CPU(四核),操作系统为Windows7,内存为8 GB,主频为3.3 GHz. 本文用python语言实现各种密码原语,其执行时间如表2所示,其中Enc代表加密算法.

表2　密码原语执行时间

操作	执行时间	操作	执行时间
MP. Enc	0.310152048399	MP. SPHF	0.00109302669866
Reg. Enc	0.0433926372716	Reg. SPHF	0.00593389340693
KV. Enc	0.873595026517	KV. SPHF	0.444697060174
SPKE. Enc	13.985888249	SPKE. SPHF	0.00112316393037
GPV. Enc	0.1392346325	GPV. SPHF	0.0097478549383

表3给出了不同协议的计算开销. 根据表3,本文协议在客户端具有最优的执行效率,这主要是因为本文协议的客户端加密算法以及GPV. SPHF都具有较高的计算效率,且不需要零知识证明. 在服务器端,本文协议的密码原语与K-PAKE-1方案相同,但本文协议在不增加计算开销(表3中K-PAKE-1的计算开销不包括签名验签的时间开销)的同时,解决了K-PAKE-1不能在超多项式模数下应用的问题.

表3　协议效率对比

协议名称	客户端	服务器端
Z-PAKE[1]	13.9870245578	13.987024017
B-PAKE[11]	14.6049738586	14.6049722413
L-PAKE-1[14]	0.316085944235	0.0444856664001
L-PAKE-2[10]	0.311245077527	0.311245077527
K-PAKE-1[9]	1.02864704845	1.02864758728
本文方案	0.299152334473	1.16812224452

6.2　协议通信与存储开销评估

为评估协议的通信与存储开销,本文假设 $l=k=128$ bit, $n=256$ bit, $m=6400$ bit, $\log q=12$. 表4总结了不同协议的通信、存储复杂度及开销.

本文协议在客户端具有最低的存储开销,这得益于本文的公共参考序列(CRS)及密文长度较短. 在服务器端,本文协议与K-PAKE-1协议选用了相同的加密算法,但其存储开销约是本文协议的1.17倍,这主要是因为本文协议在客户端的密文以及投射密钥的长度更低. 在通信开销方面,K-PAKE-1协议约是本文协议的2.98倍,这是因为本文协议不需要传递签名公钥及签名,且客户端的密文长度低. 此外,本文纠正L-PAKE-1协议设计中的一个错误:在客户端计算 $\boldsymbol{h}_{\mathrm{C}}$ 时采用Reg.Hash算法,在服务器端计算 $\boldsymbol{h}_{\mathrm{S}}$ 时采用MP.Hash算法,才能保证协议的正确性.

表4 通信与存储开销对比

协议名称		Z-PAKE[1]	B-PAKE[11]	L-PAKE-1[14]	L-PAKE-2[10]	K-PAKE-1[9]	本文方案
通信复杂度		$O(2(4m-2n_1+kn_1)\log q+k)$	$O((8m+2kn-4n_1)\log q)$	$O(2(m+kn)\log q)$	$O(2(m+kn)\log q)$	$O((3mn+2kn)\log q+k)$	$O(((mn+1)+k(n+1))\log q)$
通信开销		1001600	1394688	940032	940032	59769984	20055552
存储复杂度	客户	$O((2mn+k(m+2n_1)+8m-3n_1+n_2+1)\log q+5k)$	$O((2mn+k(m+2n)+9m+3n-4n_1)\log q+k)$	$O((mn+k(m+2n)+2m+3n+1)\log q+3k)$	$O((mn+k(m+2n)+3m+2n)\log q+3k)$	$O((mn(3k+2n+6)+n(2k+1))\log q+5k)$	$O((mn+2m+2n+1+(m+n+1)k)\log q+3k)$
	服务器	$O((2mn+k(m+2n_1)+8m-3n_1+n_2+1)\log q+5k)$	$O((2mn+k(m+2n)+9m+3n-4n_1)\log q+k)$	$O((mn+k(m+2n)+3m+3n+1)\log q+3k)$	$O((mn+k(m+2n)+3m+2n)\log q+3k)$	$O((mn(3k+2n+6)+n(2k+1))\log q+5k)$	$O((mn(2k+2n+4)+3n+1+(n+1)k)\log q+3k)$
存储开销	客户	50157184	50633088	30440844	30514560	17734832768	30046092
	服务器	50157184	50633088	30517644	30514560	17734832768	15178541964

7 总结

本文提出了一种格上基于KV密文标准语言和一种基于GPV密文标准语言的平滑投射哈希函数,解决了基于格的平滑投射哈希函数不能在超多项式模数下应用的问题;且所提出的SPHFs还可以应用在零知识证明、不经意传输和证据加密等领域. 在此基础上,本文提出了一种格上可证明安全的两轮PAKE协议,可抵抗量子攻击;不需要基于SS-NIZK,具有较高的计算效率;降低了客户端所需的安全性假设,提高了协议的实际安全性;协议只需要两轮通信,具有更优的通信轮次复杂度,这可以提高协议效率、简化协议设计和安全分析过程. 最后,本文基于更准确的标准安全模型对所提出的协议进行了严格的安全性证明,安全性证明不需要随机预言机(基于随机预言机设计协议可能会导致PAKE遭受离线口令猜测攻击). 实验证明,本文协议具有较高的计算效率;且在不增加通信开销和存储开销的前提下解决了协议不能在超多项式模数下应用的问题.

参考文献

[1] ZHANG J, YU Y. Two-round PAKE from approximate SPH and instantiations from lattices[C]//TAKAGI T. Advances in Cryptology-ASIACRYPT 2017. Cham, Germany: Springer, 2017: 37-67.

[2] KATZ J, VAIKUNTANATHAN V. Round-optimal password-based authenticated key exchange[C]//ISHAI Y. Theory of Cryptography Conference. Berlin: Springer, 2011: 293-310.

[3] KATZ J, OSTROVSKY R, YUNG M. Efficient password-authenticated key exchange using human-memorable passwords[C]//PFITZMANN B. International Conference on the Theory and Applications of Cryptographic Techniques. Berlin, Germany: Springer, 2001: 475-494.

[4] JIANG S, GONG G. Password based key exchange with mutual authentication[C]//HANDSCHUH H. International Workshop on Selected Areas in Cryptography. Berlin, Germany: Springer, 2004: 267-279.

[5] GROCE A, KATZ J. A new framework for efficient password-based authenticated key exchange[C]//AL-SHAER E S. Proceedings of the 17th ACM Conference on Computer and Communications Security. Chicago, USA: ACM, 2010: 516-525.

[6] REGEV O. On lattices, learning with errors, random linear codes, and cryptography[J]. Proceedings of the Annual ACM Symposium on Theory of Computing, 2009, 56(6): 84-93.

[7] ATENIESE G, FELICI G, MANCINI L V, et al. Hacking smart machines with smarter ones: how to extract meaningful data from machine learning classifiers[J]. International Journal of Security & Networks, 2015, 10(3): 137-150.

[8] BALUJA S, FISCHER I. Learning to attack: adversarial transformation networks[C]//MCILRAITH S. Thirty-Second AAAI Conference on Artificial Intelligence. California, USA: AAAI Press, 2018: 2687-2695.

[9] KATZ J, VAIKUNTANATHAN V. Smooth projective hashing and password-based authenticated key exchange from lattices[C]//MITSURU M. International Conference on the Theory and Application of Cryptology and Information Security. Berlin, Germany: Springer, 2009: 636-652.

[10] LI Z, WANG D. Achieving one-round password-based

authenticated key exchange over lattices[J]. IEEE Transactions on Services Computing, 2019, 2019(8): 1-14.

[11] BENHAMOUDA F, BLAZY O, LÉO D, et al. Hash proof systems over lattices revisited[C]//ABDALLA M. IACR International Workshop on Public Key Cryptography. Cham, Germany: Springer, 2018: 644-674.

[12] GENNARO R, LINDELL Y. A framework for password-based authenticated key exchange[J]. ACM Transactions on Information & System Security, 2006, 9(2): 181-234.

[13] YIN A, GUO Y, SONG Y, et al. Two-round password-based authenticated key exchange from lattices[J]. Wireless Communications and Mobile Computing, 2020, 2020 (17): 1-13.

[14] LI Z, WANG D. Two-round PAKE protocol over lattices without NIZK[C]//GUO F. International Conference on Information Security and Cryptology. Cham, Germany: Springer, 2018: 138-159.

[15] ZHANG J, YU Y, FAN S, et al. Improved lattice-based CCA2-secure PKE in the standard model[J]. Science China Information Sciences, 2020, 63(8): 22-28.

[16] 于金霞, 廉欢欢, 汤永利, 等. 格上基于口令的三方认证密钥交换协议[J]. 通信学报, 2018, 39(11): 91-101.
YU Jin-xia, LIAN Huan-huan, TANG Yong-li, et al. Password-based three-party authenticated key exchange protocol from lattices[J]. Journal on Communications, 2018, 39 (11): 91-101. (in Chinese)

[17] GENTRY C, PEIKERT C, VAIKUNTANATHAN V. Trapdoors for hard lattices and new cryptographic constructions[C]//LADNER R. Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing. New York, USA: ACM, 2008: 197-206.

[18] ALWEN J, PEIKERT C. Generating shorter bases for hard random lattices[J]. Theory of Computing Systems, 2011, 48(3): 535-553.

[19] BELLARE M, POINTCHEVAL D, ROGAWAY P. Authenticated key exchange secure against dictionary attacks [C]//PRENEEL B. International Conference on the Theory and Applications of Cryptographic Techniques. Berlin, Germany: Springer, 2000: 139-155.

[20] WANG D, CHENG H, WANG P, et al. Zipf's law in passwords[J]. IEEE Transactions on Information Forensics and Security, 2017, 12(11): 2776-2791.

[21] WANG D, WANG P. On the implications of zipf's law in passwords[C]//ASKOXYLAKIS I. European Symposium on Research in Computer Security. Cham, Germany: Springer, 2016: 111-131.

作者简介

尹安琪　女,1995年生于山东临沂. 现为信息工程大学电子技术学院博士研究生. 主要研究方向为格密码理论及格上的口令认证密钥交换协议.
E-mail: yinanqi0222@foxmail.com

曲彤洲　男,1994年生于辽宁铁岭.现为信息工程大学电子技术学院博士研究生.主要研究方向为粗粒度可重构密码逻辑阵列和密码计算.
E-mail: qutongzhou@outlook.com

郭渊博(通讯作者)　男,1975年生于陕西周至. 现为信息工程大学电子技术学院教授、博士生导师. 主要研究方向为网络防御、数据挖掘、机器学习和人工智能安全等.
E-mail: guo_yuanbo@126.com

汪　定　男,1985年生于湖北十堰. 现为南开大学网络空间安全学院教授、博士生导师. 主要研究方向为公钥密码学、系统安全、人工智能等.
E-mail: wangding@nankai.edu.cn

陈　琳　女,1975年生于河南开封. 现为信息工程大学电子技术学院副教授. 主要研究方向为安全专用芯片设计.
E-mail: chenlin916@163.com

李勇飞　男,1998年生于河南郑州,现为信息工程大学电子技术学院硕士研究生,主要研究方向为网络安全知识图谱.
E-mail: leekgfly@foxmail.com