后量子基于验证元的三方口令认证密钥交换协议

廉欢欢[1]，侯慧莹[1]，赵运磊[1,2]

（1. 复旦大学计算机科学技术学院，上海 200433；2. 西安电子科技大学综合业务网国家重点实验室，陕西 西安 710071）

摘 要：针对服务器直接以明文的方式存储口令，存在服务器泄露的风险，基于两方的基于格的口令认证密钥交换（PAKE）协议不适用于大规模通信系统的问题，提出了一种格上基于验证元的三方口令认证密钥交换协议。通过随机口令哈希方案生成验证元，并结合口令策略检查机制实现口令的检查，利用基于格的 CCA 安全公钥加密体制构造一个新的基于验证元的 3PAKE 协议，同时实现用户与服务器的双向认证。安全性和性能分析证明了所提协议在通信效率和安全度上都具有较好的优势。

关键词：三方密钥交换；口令认证；验证元；格；可证明安全

中图分类号：TP309

文献标志码：A

DOI: 10.11959/j.issn.1000−436x.2022062

Post-quantum verifier-based three-party password authenticated key exchange protocol

LIAN Huanhuan[1], HOU Huiying[1], ZHAO Yunlei[1,2]

1. College of Computer Science and Technology, Fudan University, Shanghai 200433, China

2. State Key Laboratory of Integrated Services Networks, Xidian University, Xi’an 710071, China

Abstract: In view of the fact that server stored the passwords directly in plaintext, there was a risk of server compromise, and two-party PAKE protocol was not suitable for large-scale communication systems, a three-party verifier-based password authenticated key exchange protocol from lattices was proposed. Hashing scheme and zero-knowledge password policy check were combined to realize the generation of verifier and the password checking. A novel verifier-based 3PAKE protocol was constructed by using CCA-secure public-key encryption from lattices, which realized mutual authentication. Security and performance analysis shows that the proposed protocol has better advantages in communication efficiency and security.

Keywords: three-party key exchange, password authentication, verifier, lattice, provable security

0 引言

基于口令的认证密钥交换（PAKE, password authenticated key exchange）协议使参与者之间使用低熵秘密在一个不安全的通信网络中相互认证并建立会话密钥，为后续通信建立安全的新信道。口

收稿日期：2021−08−16；**修回日期**：2021−12−17

通信作者：赵运磊，ylzhao@fudan.edu.cn

基金项目：国家自然科学基金资助项目（No.U1536205, No.61472084）；国家重点研发计划基金资助项目（No.2017YFB0802000）；上海市创新行动计划基金资助项目（No.16DZ1100200）；上海市科学技术发展基金资助项目（No.16JC1400801）；上海市科委技术标准基金资助项目（No.21DZ2200500）；山东省重点研发计划基金资助项目（No.2017CXG0701, No.2018CXGC0701）

Foundation Items: The National Natural Science Foundation of China (No.U1536205, No.61472084), The National Key Research and Development Program of China (No.2017YFB0802000), Shanghai Innovation Action Project (No.16DZ1100200), Shanghai Science and Technology Development Funds (No.16JC1400801), Technical Standard Project of Shanghai Scientific and Technological Committee (No.21DZ2200500), Shandong Provincial Key Research and Development Program (No.2017CXG0701, No.2018CXGC0701)

令认证简单易记、部署方便，而且可以摆脱复杂的公钥证书和密钥管理。因此，基于口令的认证密钥交换协议是目前应用比较广泛的协议。1992 年，Bellovin 和 Merritt[1]给出了第一个真正意义上的 PAKE 协议，并采用启发式的方法对提出的方案进行了形式化的安全性分析。PAKE 协议引起了广泛关注，众多密码学者提出了具有不同计算效率和安全性的 PAKE 协议[2-5]。

由于具有并行运算、计算速度快和抗已知量子攻击等特点，基于格难题的 PAKE 协议成为后量子时代密码领域的研究热点。2009 年，Katz 等[6]提出了第一个基于格上难题的口令认证密钥交换方案。该方案构造了基于格的近似平滑投射哈希（ASPH, approximate smooth projective Hash）函数，并将其用于 GL（Gennaro-Lindell）框架[7]，从而设计了 PAKE 协议，但该协议投射密钥依赖于密文，密钥长度过长使协议效率较低。2012 年，Ding 等[8]构造了一种较高效的格基 PAKE 协议，该协议利用文献[6]提出的加密体制和 ASPH 函数，在 Groce-Katz[9]框架下进行构造，并在标准模型下给出了安全性证明。Ding 等[10]采用 HMQV 变体设计了 2 种基于格的 PAKE 协议，分别是提供隐式认证的两轮方案和提供显式认证的三轮方案，并且都是在随机预言模型下构造的。Zhang 等[11]构造了可拆分的公钥加密体制，而且改进了 ASPH 函数，提出了一种仅需两轮的 PAKE 协议。之后不同构造的后量子 PAKE 协议相继被提出[12-13]。上述 PAKE 协议均是两方的密钥交换协议，需要每 2 个用户共享一个口令，在大量用户通信中口令管理非常复杂，因此不适用于大规模通信系统中。

三方 PAKE（3PAKE, three-party PAKE）协议中每个用户仅需和服务器共享一个口令，用户在服务器的帮助下建立共同的会话密钥，有效地解决了两方 PAKE 协议的局限性。叶茂等[14]利用文献[6]提出的公钥加密方案和相应的 ASPH 函数，在文献[9]的框架下，构造了一个基于带误差学习（LWE, learning with error）问题的 3PAKE 协议，满足大规模端到端的通信需求。2017 年，Xu 等[15]基于环上带误差学习（RLWE, ring-learning with error）问题构造了 3PAKE 协议并给出了安全性证明，但该协议存在效率较低的缺陷。王彩芬等[16]在格困难问题上提出了隐式认证和显式认证 2 种 3PAKE 协议，并且提供了用户匿名性。于金霞等[17]采用文献[11]提出的可拆分加密体制，构造了一个基于格的 3PAKE 协议。Jiang 等[18]于 2020 年提出一种新的 PAKE 框架，采用 ASPH 函数、密钥协调机制和认证码，基于 LWE 和 RLWE 困难问题构造 2 种 PAKE 协议。

上述格上 PAKE 协议中的口令直接以明文的方式存储在服务器上，而随着口令控制访问的增加，对于不同的提供者，用户需要频繁使用相应的口令，因此若存在服务器攻击，将带来口令文件泄露的风险。为解决上述问题，Bellovin 和 Merritt[19]构造出第一个增强的加密密钥交换协议，协议中服务器只存储用来验证拥有正确口令的用户身份的验证元，不直接存储明文口令。在具体的系统中，验证元是对盐值和口令进行运算得到的变换值。即使服务器端口令文件泄露，攻击者也需要更多的时间进行离线攻击才能获得口令，因此此类协议有效地减少了服务器端口令文件泄露所造成的影响。Benhamouda 等[20]提出了 2 种有效的口令哈希方案，基于此构造了不同轮数的基于验证元的口令认证密钥交换（VPAKE, verifier-based PAKE）协议。杨晓燕等[21]提出了标准模型下三方的 VPAKE 协议，并给出了严格的安全性证明。2020 年，张启慧等[22]指出杨晓燕等[21]提出的协议难以抵抗离线字典攻击，未达到声称的安全性，并提出了改进的基于验证元的 3PAKE 协议。然而这些 VPAKE 协议都是基于经典计算机上困难的计算性问题，能够在多项式时间内解决。2021 年，舒琴等[23]提出了首个格上基于验证元的 3PAKE 协议，在通用可组合框架下，定义 3VPAKE 理想功能，构造一个新的理想格上的协议，但该协议消息传输量较多。

通过以上分析，基于格上难题的 PAKE 协议绝大多数都是针对“用户−服务器”场景设计的两方协议，两方协议在大规模用户端到端应用中，每 2 个用户通信需要共享一个口令，那么用户记忆的口令数量随着用户数量的增加呈线性增长，这给资源受限的客户端造成了较大的存储负担，因此不能满足大规模“用户−用户”的通信需求[14,24]。而基于格的三方协议中每个用户只需要存储一个口令，有效地减少了存储开销，并且能够抵抗已知量子攻击，更适用于大规模用户端到端应用场景，例如车联网中大规模车辆与车辆之间的通信、智能医疗网络中大量患者与医护人员之间的通信、移动设备获取网络服务等。此外，现有的格上 PAKE 协议存在服务器口令信息泄露带来的风险。因此，本文基于格上

难题构造一种基于验证元的三方口令认证密钥交换协议。本文主要贡献如下：1) 结合口令 Hash 方案和零知识口令策略检查机制在不泄露口令的前提下，实现验证元的生成和口令的认证；2) 利用基于 LWE 问题的选择密文攻击（CCA, chosen ciphertext attack）安全的公钥加密体制和近似平滑投射函数，基于目前高效的格上 2PAKE 框架[11]设计新的 3VPAKE 协议。该协议具有抵抗不可检测在线字典攻击和服务器信息泄露的优点，并且实现了双向认证的功能。

1　相关知识

1.1　格和困难问题

定义 1　格。给定线性空间 $\mathbb{R}^{n\times m}$ 上 m 个线性无关向量 $\boldsymbol{B}=(\boldsymbol{b}_1,\cdots,\boldsymbol{b}_m)$，这些向量的整系数线性组合表示为格 $\boldsymbol{\Lambda}\subset\mathbb{R}^n$，即 $\boldsymbol{\Lambda}=\mathcal{L}(\boldsymbol{B})=\sum_{i=1}^{m}x_i\boldsymbol{b}_i, x_i\in\mathbb{Z}$。

定义 2　LWE 分布[25]。对于任意正整数 $n,q\in\mathbb{Z}$，秘密向量 $\boldsymbol{s}\in\mathbb{Z}_q^n$ 和实数 $\alpha>0$，LWE 分布 $A_{s,\alpha}$ 的输出为 $(\boldsymbol{a},\boldsymbol{b}=\boldsymbol{a}^t\boldsymbol{s}+\boldsymbol{e} \bmod q)\in\mathbb{Z}_q^n\times\mathbb{Z}_q$，其中，$\boldsymbol{a}\leftarrow_r\mathbb{Z}_q^n$ 为随机均匀选取的向量，$\boldsymbol{e}\leftarrow_r D_{\mathbb{Z},\alpha q}$ 为容错向量。

定义 3　判定 LWE 问题。给定 m 个独立采样 $(\boldsymbol{a}_1,\boldsymbol{b}_1),\cdots,(\boldsymbol{a}_m,\boldsymbol{b}_m)$，其中 $\boldsymbol{a}_i\in\mathbb{Z}_q^n$，$\boldsymbol{b}_i\in\mathbb{Z}_q$，每个样本是从以下一个分布中选取的：1) LWE 分布 $A_{s,\alpha}$；2) $\mathbb{Z}_q^n\times\mathbb{Z}_q$ 均匀分布。任意 PPT 算法能够区分样本取自哪种分布的优势是可忽略的。

1.2　公钥加密体制

在本文协议中，带标签公钥加密体制 $\mathcal{PKE}$= (KeyGen, Enc, Dec) 包含以下 3 种算法。1) 密钥生成算法 $(\mathrm{pk},\mathrm{sk})\leftarrow\mathrm{KeyGen}(1^\kappa)$：输入公共参数，输出公私钥对。2) 加密算法 $c\leftarrow\mathrm{Enc}(\mathrm{pk},\mathrm{label},\mathrm{pw},r)$：输入公钥和明文，并输入标签和随机数，输出密文。3) 解密算法 $\mathrm{pw}\leftarrow\mathrm{Dec}(\mathrm{sk},\mathrm{label},c)$：输入私钥、密文和相应的标签，输出相应的明文。

正确性。对于公私钥对(pk, sk)、安全参数 κ 和随机选取的 r，密文 c 在标签 $\mathrm{label}\in\{0,1\}^*$ 上得到明文 pw，$\Pr[\mathrm{Dec}(\mathrm{sk},\mathrm{label},c)=\mathrm{pw}\mid (\mathrm{pk},\mathrm{sk})\leftarrow\mathrm{KeyGen}(1^\kappa)\wedge c\leftarrow\mathrm{Enc}(\mathrm{pk},\mathrm{label},\mathrm{pw})]\geqslant 1-\mathrm{negl}(\kappa)$。

1.3　近似平滑投射哈希函数

平滑投射哈希函数最早由 Cramer 和 Shoup[26] 提出，后来由密码学家们在 PAKE 协议的研究中相继进行了改进[5-6]。通俗地讲，给定集合 X 和语言 L，L 表示 X 的某个子集。对于词语 $c\in L$，$\mathrm{hk}\leftarrow K$ 表示哈希密钥，hp 表示投射密钥，近似平滑投射哈希函数由 4 种算法组成：1) 采样一个哈希密钥 $\mathrm{hk}\leftarrow_r K$；2) 对于 $\mathrm{hk}\in K$，计算投射密钥 $\mathrm{hp}=\mathrm{Proj}(\mathrm{hk})$；3) 对于 $c\in X$，$\mathrm{hk}\leftarrow_r K$，计算哈希值 $\mathrm{Hash}(\mathrm{hk},L,c)$；4) 利用投射密钥 hp，对于 $c\in L$ 以及相应的证据 w，计算哈希值 $\mathrm{ProjHash}(\mathrm{hp},L,c,w)$。

ϵ-approximate 平滑投射哈希函数满足近似正确性和平滑性 2 个性质。1) 近似正确性：对于合法的词语 $c\in L$ 和相应的证据 w，满足 $\Pr[\mathrm{Ham}(\mathrm{Hash}(\mathrm{hk},L,c),\mathrm{ProjHash}(\mathrm{hp},L,c,w))\geqslant\epsilon]=\mathrm{negl}(\kappa)$，其中 $\epsilon\in\left(0,\frac{1}{2}\right)$ 为较小的实数。2) 平滑性：对于 $c\in X\setminus L$，即使知道 hp，$\mathrm{Hash}(\mathrm{hk},L,c)$ 和均匀随机选取的输出是统计不可区分的。在本文协议设计中，投射密钥不依赖于密文，只依赖于哈希密钥。

1.4　随机口令哈希方案和口令策略检查

随机口令哈希方案[20,27-28]可用于生成验证元，本文利用文献[28]设计的格上困难问题的口令哈希方案，该方案表示为 $\mathcal{H}=(\mathrm{PSetup},\mathrm{PPreSalt},\mathrm{PPreHash},\mathrm{PSalt},\mathrm{PHash})$，定义如下。

$\mathrm{PSetup}(\kappa)$：安全参数为 κ，生成公共参数 $\mathrm{pp}=(n,q,m,\mathbb{S}_P,\mathbb{S}_H,\boldsymbol{A},\boldsymbol{B})$。

$\mathrm{PPreSalt}(\mathrm{pp})$：输入公共参数 pp，采样 $\chi\leftarrow\mathcal{S}_{n_{\max}}$，输出 $s_P=\chi$。

$\mathrm{PPreHash}(\mathrm{pp},\mathrm{pw},s_P)$：输出预哈希值 P，计算 $\mathrm{encode}(pw)\in\{0,1\}^{8t}$ 得到 $\boldsymbol{e}\in\{0,1\}^{8n_{\max}}$，利用 $T_{\chi,8}$ 得到 $\boldsymbol{e}'=T_{\chi,8}(\boldsymbol{e})\in\{0,1\}^{8n_{\max}}$，其中 T 为置换运算[26]，并输出 $P=\boldsymbol{e}'$。

$\mathrm{PSalt}(\mathrm{pp})$：采样 $\boldsymbol{r}_0\leftarrow\{0,1\}^m$，输出 $s_H=\boldsymbol{r}_0$。

$\mathrm{PHash}(\mathrm{pp},P,s_P,s_H)$：输出哈希值 $\boldsymbol{h}$，形成 $\boldsymbol{e}_0=(\mathrm{bin}(\chi(1)-1)\|\cdots\|\mathrm{bin}(\chi(n_{\max})-1))\in\{0,1\}^{n_{\max}\lceil\log n_{\max}\rceil}$，$\boldsymbol{x}=\boldsymbol{e}_0\|\boldsymbol{e}'$，并输出 $\boldsymbol{h}=\boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{r}_0\in\mathbb{Z}_q^n$。

该方案中 PHash 算法通过输入预哈希值 P、预盐值 s_P 和盐值 s_H 计算哈希值，这里 P 作为输入可以对口令实现隐藏；s_P 和 s_H 增加了随机性，可以防止离线字典攻击，提高破解口令的困难性。

口令策略检查（PPC, password policy check）[27-28] 是客户端与服务器之间交互的协议，是使用户向服务器注册口令并证明口令符合服务器的策略。通过

输入口令 Hash 方案 $\mathcal{H}$ 的公共参数和口令策略 $f=((k_1,k_2,k_3,k_4),n_{\min},n_{\max})$，即口令的长度在 $n_{\min}$ 和 $n_{\max}$ 之间并且包含 k_1 个数字、k_2 个符号、k_3 个小写字母和 k_4 个大写字母，当且仅当 $f(\mathrm{pw})=\mathrm{true}$ 时，服务器端接受用户所选取的口令的哈希值。

2 安全模型

Bellare 和 Pointcheval 在文献[29]提出的模型上进行了改进，本节在改进模型[20]的基础上给出了 3VPAKE 的安全性分析模型。

1) 用户和口令

协议参与者包含用户和可信服务器，令 $U\in\mathcal{U}$ 为协议用户的集合，$S\in\mathcal{S}$ 为可信服务器。每个用户 $U\in\mathcal{U}$ 拥有一个口令 pw_C，将其映射为 π_C。每个服务器 S 拥有关于 π_C 的哈希值 $h=<h_U=\mathrm{Hash}(\mathrm{pp},P,s_P,s_H)>_{U\in\mathcal{U}}$ 和盐值，其中假设每一个口令都是从字典集合 $\mathcal{D}$ 中独立均匀选取的。

2) 协议执行

敌手 $\mathcal{A}$ 通过下述几种询问预言机与所有实例进行交互。

$\mathrm{Execute}(U_1,i_1,U_2,i_2,S,j)$。这个询问模型化敌手的被动攻击，敌手偷听实例 $\prod_{U_1}^{i_1}$、$\prod_{U_2}^{i_2}$ 与 $\prod_S^j$ 之间诚实的协议执行，并将消息返回给 $\mathcal{A}$。

$\mathrm{Send}(U_1,i_1,\mathrm{msg})$。这个询问模型化了敌手的主动攻击，敌手拦截并修改发送给实例 $\prod_{U_1}^{i_1}$ 的消息 msg，将用户实例 $\prod_{U_1}^{i_1}$ 对消息的回应返回给 $\mathcal{A}$。

$\mathrm{Send}(U_2,i_2,\mathrm{msg})$。这个询问模型化了敌手的主动攻击，敌手拦截并修改发送给实例 $\prod_{U_2}^{i_2}$ 的消息 msg，将用户实例 $\prod_{U_2}^{i_2}$ 对消息的回应返回给 $\mathcal{A}$。

$\mathrm{Send}(S,j,\mathrm{msg})$。这个询问模型化了敌手的主动攻击，敌手拦截并修改发送给服务器 $\prod_S^j$ 的消息 msg，将服务器 $\prod_S^j$ 对消息的回应返回给 $\mathcal{A}$。

$\mathrm{Reveal}(U,i)$。这个询问将生成的会话密钥 sk_U^i 返回给敌手。

$\mathrm{Corrupt}(U)$。这个询问是敌手对用户的腐化建立模型，它的输出为用户的口令和用户的会话内部状态。

$\mathrm{Corrupt}(S)$。这个询问是敌手对服务器的腐化建立模型，它的输出为服务器的口令哈希值和服务器的会话内部状态。

$\mathrm{Test}(U,i)$。这个询问模型选择一个随机比特 $b\leftarrow_r\{0,1\}$，如果 b=1，则返回用户实例 $\prod_U^i$ 的会话密钥 sk_U^i；否则返回均匀选取的随机值。在攻击期间，敌手只能访问一次 Test。

3) 伙伴关系和新鲜性

① sk_U^i 表示 $\prod_U^i$ 的会话密钥，sid_U^i 表示 $\prod_U^i$ 的会话标识，pid_U^i 表示 $\prod_U^i$ 交互对等方标识，acc_U^i 表示对实例是否接受（不接受即拒绝），term_U^i 表示会话是否结束。

② 如果 $\mathrm{sid}_U^{i_1}=\mathrm{sid}_U^{i_2}\neq\perp$、$\mathrm{pid}_{U_1}^{i_1}=\prod_{U_2}^{i_2}$ 和 $\mathrm{pid}_{U_2}^{i_2}=\prod_{U_1}^{i_1}$，实例 $\prod_{U_1}^{i_1}$ 和 $\prod_{U_2}^{i_2}$ 互为匹配会话。如果实例 $\prod_{U_1}^{i_1}$ 和 $\prod_{U_2}^{i_2}$ 互为匹配会话，$\mathrm{acc}_{U_1}^{i_1}=\mathrm{acc}_{U_2}^{i_2}=1$ 且 $\mathrm{sid}_{U_1}^{i_1}=\mathrm{sid}_{U_2}^{i_2}\neq\perp$，那么称协议是正确的。

③如果会话 sid 已接受，且敌手未对会话 sid 或者匹配会话进行过 Reveal 询问，且会话 sid 接受之前，攻击者未对用户和服务器进行过 Corrupt 询问，则会话 sid 是新鲜的。

定义 4 语义安全性。协议运行中，敌手 $\mathcal{A}$ 可以任意顺序进行多次 Execute、$\mathrm{Send}(U_1,i_1,\mathrm{msg})$、$\mathrm{Send}(U_2,i_2,\mathrm{msg})$、$\mathrm{Send}(S,j,\mathrm{msg})$、Reveal 和 Corrupt 询问，但对诚实的新鲜实例只进行一次 Test 询问，输出 b'，b 为从 Test 预言机中选择的比特，若 $b'=b$，则表示敌手成功。敌手 $\mathcal{A}$ 的优势表示为 $\mathrm{Adv}_{\Pi,\mathcal{A}}(\kappa)=2\Pr[\mathrm{Succ}]-1$。

如果敌手 $\mathcal{A}$ 的优势满足 $\mathrm{Adv}_{\Pi,\mathcal{A}}(\kappa)\leqslant\frac{Q(\kappa)}{|\mathcal{D}|}+\mathrm{negl}(\kappa)$，则称 VPAKE 协议是安全的。其中，$Q(\kappa)$ 表示敌手可进行在线攻击的次数，$\mathcal{D}$ 表示字典集合。

3 基于格的三方 VPAKE 协议

3.1 模块构造

本节主要描述本协议中的零知识口令策略检查（ZKPPC, zero-knowledge PPC）机制和近似平滑投射哈希函数。

1) 零知识口令策略检查机制

本文利用文献[28]提出的机制使证明者在不泄露口令的情况下向验证者证明自己知道口令。本文协议中，用户在零知识环境中使服务器相信自己拥有的口令，可以通过哈希得到给定的哈希值，并且该口令满足口令策略 $f=((k_1,k_2,k_3,k_4),n_{\min},n_{\max})$。具体说明如下。

令 $k_{\text{all}} = n_{\min} - (k_1 + k_2 + k_3 + k_4)$，输入口令哈希中的公共参数 $\boldsymbol{A}$、$\boldsymbol{B}$ 和哈希值 $\boldsymbol{h}$，以及其他信息 $\Delta = (\delta_{D,1}, \cdots, \delta_{D,k_1}, \cdots, \delta_{U,k_4}, \delta_{\text{all},1}, \cdots, \delta_{\text{all},k_{\text{all}}}) \in [n_{\max}]^{n_{\min}}$，信息 Δ 包含内部向量 $\boldsymbol{P} = \boldsymbol{e}'$ 的位置，以及对口令中数字、符号、字母和其他信息的编码的位置。由于这些模块的原始位置被秘密值 χ 保护，即使 Δ 被泄露给服务器也不会损害用户端。用户端的证据包含向量 $\boldsymbol{x} = (\boldsymbol{e}_0 \,\|\, \boldsymbol{e}')$ 和 $\boldsymbol{r}_0 \in \{0,1\}^m$，其中 $\boldsymbol{e}'$ 的形式为 $(x_1, \cdots, x_{n_{\max}})$，而且 $\boldsymbol{e}_0 = (\text{bin}(\chi(1)-1) \| \cdots \| \text{bin}(\chi(n_{\max})-1))$。服务器检查如果满足 $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{r}_0 = \boldsymbol{h} \bmod q$ 和 $f(\text{pw}) = \text{true}$，则用户是可信的。本文验证元是通过口令哈希方案计算生成的，在服务器不知道口令的情况下，ZKPPC 协议确保用户的口令符合服务器端的口令策略，并且与用户注册阶段结束时通信的验证元信息相关。

2) ASPH 函数

本节根据 CCA 公钥加密机制[6]给出了相应的适用于本文 VPAKE 协议的 ASPH 函数。

令 $\mathcal{PKE}$= (KeyGen, Enc, Dec) 为基于格的 CCA 安全公钥加密体制，存在 $n_1, n_2 \in \mathbb{Z}$，q 为素数，$n_2 = n_1 + n + 1$，$m = O(n \log q) \in \mathbb{Z}$，包含 3 种算法。①密钥生成算法：输入安全参数 κ，计算 $(\boldsymbol{R}, \boldsymbol{V}) \leftarrow \text{TrapGen}(1^{n_2}, 1^m, q)$ 和 $\text{crs} \leftarrow \text{CRSGen}(1^\kappa)$，输出公私钥对 $(\boldsymbol{R}, \boldsymbol{V})$。②加密算法 Enc(pk, m', label)：给出公钥 $\boldsymbol{R}$ 和明文 $\boldsymbol{m}'$，随机选取 $\boldsymbol{s} \in \mathbb{Z}_q^{n_1}$ 和 $\tilde{\boldsymbol{e}} \leftarrow {}_r D_{Z^m, \alpha q}$，得到密文 $\boldsymbol{c} = \boldsymbol{R}\begin{pmatrix} \boldsymbol{s} \\ 1 \\ \boldsymbol{m}' \end{pmatrix} + \tilde{\boldsymbol{e}} (\bmod q)$ 和 $\pi' \leftarrow \text{Prove}(\text{crs}, \boldsymbol{R}, \boldsymbol{c}, \boldsymbol{s}, \text{label})$。③解密算法 Dec(sk, c, label)：给出私钥 $\boldsymbol{R}$、π' 和 $\boldsymbol{c}$，如果 $\text{Verify}(\text{crs}, \boldsymbol{R}, \boldsymbol{c}, \pi', \text{label}) = 0$，则返回 $\perp$，否则计算 $\begin{pmatrix} \boldsymbol{s} \\ 1 \\ \boldsymbol{m}' \end{pmatrix} \leftarrow \text{Solve}(\boldsymbol{R}, \boldsymbol{V}, \boldsymbol{c})$，返回 $\boldsymbol{m}'$。

ASPH 函数。X 表示一个集合，L 被定义为语言且满足 $L \subset X$。

$$X = \{(\text{label}, c) \mid \text{label} \in \{0,1\}^*, c \in C_{\text{pk}}\}$$

$$L_H = \{(\text{label}, c) \mid \exists r, c = \text{Enc}(\text{pk}, \text{label}, H; r)\}$$

$$L_{s,H} = \{(\text{label}, c) \mid \exists \pi, \exists r, c = \text{Enc}(\text{pk}, \text{label}, \pi; r) \wedge H = \text{PHash}(\text{pp}, \text{PPrehash}(\text{pp}, \pi, s_P), s_P, s_H)\}$$

其中，$s = (s_P, s_H)$，$\epsilon \in (0, \frac{1}{2})$。

对于语言 L_H，近似平滑投射哈希函数主要按照 1.3 节所述的内容进行展开。对于语言 $L_{s,H}$，所构造的 ASPH 函数如下。

哈希密钥生成函数 HKGen(K)。在高斯分布中采样哈希密钥，其表示为 $\text{hk} = (\boldsymbol{g}_1, \cdots, \boldsymbol{g}_\ell)$，其中 $\boldsymbol{g}_i \sim D_{Z^m, \gamma}$，令 $\boldsymbol{R} = (\boldsymbol{B}' \| \boldsymbol{U}) \in \mathbb{Z}_q^{m \times n_2}$，使 $\boldsymbol{B}' = \mathbb{Z}_q^{m \times n_1}$，$\boldsymbol{U} = \mathbb{Z}_q^{m \times (n+1)}$。

投射密钥生成函数 PKGen(hk)。投射密钥 hp=Proj($\boldsymbol{g}_1, \cdots, \boldsymbol{g}_\ell$)=($\boldsymbol{u}_1, \cdots, \boldsymbol{u}_\ell$)，计算出 $\boldsymbol{u}_i = \boldsymbol{B}'^{\mathrm{T}} \boldsymbol{g}_i$，$1 \leqslant i \leqslant \ell$。

哈希函数 Hash(hk, $L_{s,H}$, c)。给出哈希密钥 $\text{hk} = (\boldsymbol{g}_1, \cdots, \boldsymbol{g}_\ell)$ 和 $(\text{label}, \boldsymbol{y}) \in X$，盐值 $\boldsymbol{r}_0$，验证元 $\boldsymbol{h}$，其中 $\boldsymbol{y}$ 为密文。首先，计算 $z_i = \boldsymbol{g}_i^{\mathrm{T}} \left[\boldsymbol{y} - \boldsymbol{U} \begin{pmatrix} 1 \\ \boldsymbol{h} - \boldsymbol{B}\boldsymbol{r}_0 \end{pmatrix} \right] \in \mathbb{Z}_q$；其次，每个 z_i 在 $\left[-\frac{q-1}{2}, \frac{q-1}{2} \right]$ 内，如果 z_i=0，令 $b_i \leftarrow {}_r\{0,1\}$，否则令

$$b_i = \begin{cases} 0, & z_i < 0 \\ 1, & z_i > 0 \end{cases}$$

投射哈希函数 ProjHash(hp, $L_{s_H,H}$, $\boldsymbol{c}$, $\boldsymbol{w}$, $\boldsymbol{s}$)。对于 $\tilde{\boldsymbol{e}} \leftarrow {}_r D_{Z^m, \alpha q}$，给出 hp=$(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_\ell) \in S$，$(\text{label}, \boldsymbol{y}, w) \in L_{s_H,H}$ 和 $\boldsymbol{s} \in \mathbb{Z}_q^{n_1}$，其中 w 和 $\boldsymbol{s}$ 为证据，$\boldsymbol{y}$ 为密文，计算 $\boldsymbol{e} \leftarrow \text{encode}(w)$，对 $\boldsymbol{e}$ 进行置换运算得到 $\boldsymbol{e}'$，令 $P = \boldsymbol{e}'$，计算 $\boldsymbol{e}_0 = (\text{bin}(\chi(1)-1) \| \cdots \| \text{bin}(\chi(n_{\max})-1))$ 和 $\boldsymbol{x} = \boldsymbol{e}_0 \| \boldsymbol{e}'$，使密文 $\boldsymbol{y} = \boldsymbol{B}'\boldsymbol{s} + \boldsymbol{U}\begin{pmatrix} 1 \\ \boldsymbol{A}\boldsymbol{x} \end{pmatrix} + \tilde{\boldsymbol{e}} (\bmod q)$，计算 $z_i' = \boldsymbol{u}_i^{\mathrm{T}} \boldsymbol{s}$。每个 z_i' 在 $\left[-\frac{q-1}{2}, \frac{q-1}{2} \right]$ 范围，如果 b_i'=0，令 $b_i' \leftarrow {}_r\{0,1\}$，否则令

$$b_i' = \begin{cases} 0, & z_i' < 0 \\ 1, & z_i' > 0 \end{cases}$$

近似正确性。对于 $(\text{label}, \boldsymbol{y}) \in L_{s,H}$，其中 $\boldsymbol{y}$ 为密文且 $\boldsymbol{y} = \boldsymbol{B}'\boldsymbol{s} + \boldsymbol{U}\begin{pmatrix} 1 \\ \boldsymbol{A}\boldsymbol{x} \end{pmatrix} + \tilde{\boldsymbol{e}} (\bmod q)$。对于每个 $i \in [\ell]$，$z_i = \boldsymbol{g}_i^{\mathrm{T}} \left(\boldsymbol{y} - \boldsymbol{U}\begin{pmatrix} 1 \\ \boldsymbol{h} - \boldsymbol{B}\boldsymbol{r}_0 \end{pmatrix} \right) = \boldsymbol{g}_i^{\mathrm{T}} (\boldsymbol{B}'\boldsymbol{s} + \tilde{\boldsymbol{e}}) = \boldsymbol{u}_i^{\mathrm{T}} \boldsymbol{s} + \boldsymbol{g}_i^{\mathrm{T}} \tilde{\boldsymbol{e}}$，则存在 $|z_i - z_i'| = |\boldsymbol{g}_i^{\mathrm{T}} \tilde{\boldsymbol{e}}| \leqslant \gamma\sqrt{m}\alpha q\sqrt{m} \leqslant \frac{\epsilon}{2}\frac{q}{4}$ 的概率是不可忽略的。每个 $\boldsymbol{u}_i$ 在 $\mathbb{Z}_q^{n_1}$ 上是统计均匀的，那么 $z_i' = \boldsymbol{u}_i^{\mathrm{T}} \boldsymbol{s}$ 也是均匀随机的，因此通过计算得到

$\Pr[b_i \neq b_i'] \leqslant \frac{\epsilon}{2}$ 。根据切诺夫界可知 $\Pr[\mathrm{Ham}(\mathrm{Hash}(\mathrm{hk}, L, c), \mathrm{ProjHash}(\mathrm{hp}, L, c, w, r)) \leqslant \epsilon\ell]$ 是不可忽略的。

平滑性。对于 $(\mathrm{label}, \boldsymbol{y}) \in X \setminus L_{s,H}$，2 个分布 $\{\mathrm{hp}, \mathrm{Hash}(\mathrm{hk}, L, c)\}$ 与 $\{\mathrm{hp}, \rho \leftarrow \{0,1\}^*\}$ 在统计上是不可区分的。

3.2 方案构造

本节给出了一个新的格上基于验证元的 3PAKE 协议，该协议利用了文献[6, 11]中 PAKE 的设计思想以及文献[28]给出的验证元的生成方式和验证技术。

本文协议的系统模型由以下实体组成：用户集合 U 和可信服务器 S。整体系统模型如图 1 所示。用户集合中每个客户持有各自的口令，与想要通信的用户之间协商共享的会话密钥，为后续通信建立安全信道。假设服务器端是可信的，存储用户集合各自口令对应的哈希值和盐值，计算出临时秘密值，帮助拥有不同口令的 2 个用户建立共同的密钥。首先，服务器端对用户端口令进行认证，确认用户拥有的口令满足口令策略，并且与哈希值相关，再接受相应的哈希值；其次，2 个用户通过服务器传输信息并计算出临时秘密值；最后，2 个用户交互协商出共享的会话密钥。

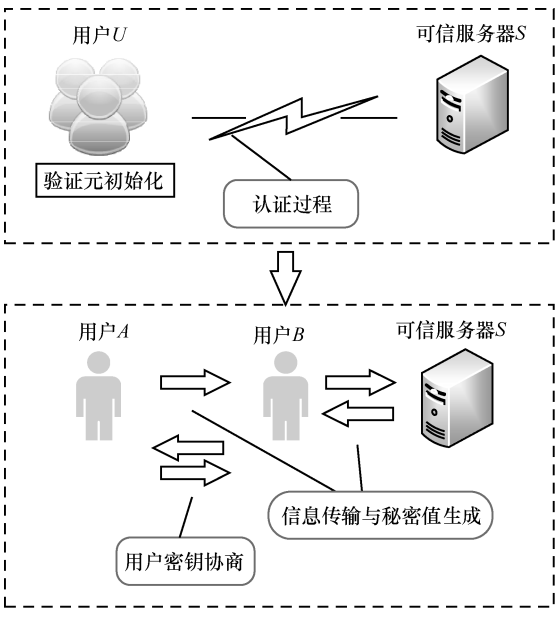

图 1 整体系统模型

设 κ 表示安全参数，函数 $H_1 : \{0,1\}^* \to \{0,1\}^\kappa$ 表示单向抗碰撞哈希函数；$\mathrm{ECC} : \{0,1\}^\ell \to \{0,1\}^\kappa$ 表示可以纠错 2ϵ 小部分的编码算法，$\mathrm{ECC}^{-1} : \{0,1\}^\kappa \to \{0,1\}^\ell$ 表示解码算法；$F = \{F_{\mathrm{mk}} : \mathrm{mk} \in \{0,1\}^\ell\}_{\ell \in \mathbb{N}}$ 表示伪随机函数，协议中的符号说明如表 1 所示。3VPAKE 协议具体描述如图 2 所示。

表 1 符号说明

符号	说明
hk	哈希密钥
hp	投射密钥
label	标签
H_{SA}, H_{SB}	存储于服务器上的 A 和 B 的口令哈希值
s_{P_1}, s_{P_2}	预盐值
s_{H_1}, s_{H_2}	盐值
mk	伪随机函数密钥
$\mathrm{sk}_{AB}, \mathrm{sk}_{BA}$	由用户 A 和 B 生成的会话密钥

初始化。$\mathcal{PKE}$ 体制公钥由可信第三方生成，表示为 pk。假设 U 表示协议用户的集合，A 和 B 表示共同建立会话密钥的 2 个用户，S 表示可信服务器。A 的口令为 pw_A，B 的口令 pw_B，分别被映射为 π_A 和 π_B。S 拥有 A 和 B 口令的验证值，分别为 (H_{SA}, s_{H_1}) 和 (H_{SB}, s_{H_2})，P_U 和 s_P 由用户端拥有，这里哈希值 $H_S = H_U$。A 输入口令 π_A 和身份，口令私有保存，身份发送给服务器 S，S 从数据库中得到该用户的信息对 (H_{SA}, s_{H_1})。B 进行类似的操作。

协议执行。1) 用户 A 首先随机选取 $r_A \leftarrow_r \{0,1\}^*$，哈希密钥 $\mathrm{hk}_A \leftarrow_r K$，计算投射密钥 $\mathrm{hp}_A = \mathrm{Proj}(\mathrm{hk}_A)$，并计算 $\mathrm{label}_A := A \| B \| S \| \mathrm{hp}_A$ 和 $\mathrm{label}'_A := H_1(\mathrm{lable}_A)$，然后计算密文 $c_A := \mathrm{Enc}(\mathrm{pk}, \mathrm{label}_A, \pi_A; r_A)$。用户 A 向用户 B 发送消息 $M_1 = (\mathrm{hp}_A, c_A, \mathrm{label}'_A, A)$。

2) 用户 B 收到用户 A 发送的消息后，选取随机值 $r_B \leftarrow_r \{0,1\}^*$，哈希密钥 $\mathrm{hk}_B \leftarrow_r K$，计算投射密钥 $\mathrm{hp}_B = \mathrm{Proj}(\mathrm{hk}_B)$，然后计算 $\mathrm{label}_B := A \| B \| S \| \mathrm{hp}_B$、$\mathrm{label}'_B := H_1(\mathrm{lable}_B)$ 和密文 $c_B := \mathrm{Enc}(\mathrm{pk}, \mathrm{label}_B, \pi_B; r_B)$。将消息 $M_2 = (M_1, \mathrm{hp}_B, c_B, \mathrm{label}'_B, B)$ 发送给服务器 S。

3) 服务器收到用户 B 发送的消息之后，通过 $\mathrm{label}_A := A \| B \| S \| \mathrm{hp}_A$ 验证 label'_A 是否等于 $H_1(\mathrm{lable}_A)$。选取随机值 $r_{SA} \leftarrow_r \{0,1\}^*$，哈希密钥 $\mathrm{hk}_{SA} \leftarrow_r K$，并计算投射密钥 $\mathrm{hp}_{SA} = \mathrm{Proj}(\mathrm{hk}_{SA})$、$\mathrm{label}_{SA} := A \| B \| S \| \mathrm{hp}_{SA}$ 和密文 c_{SA}。服务器 S 以相同

$A(\pi_A)$　　$B(\pi_B)$　　$S(H_{SA}, H_{SB}, s_{H_1}, s_{H_2})$

$r_A \leftarrow_r \{0,1\}^*, \mathrm{hk}_A \leftarrow_r K, \mathrm{hp}_A = \mathrm{Proj}(\mathrm{hk}_A)$

$\mathrm{label}_A := A \| B \| S \| \mathrm{hp}_A, \mathrm{label}'_A := H_1(\mathrm{lable}_A)$

$c_A := \mathrm{Enc}(\mathrm{pk}, \mathrm{label}_A, \pi_A; r_A)$

$\xrightarrow{M_1=(\mathrm{hp}_A, c_A, \mathrm{label}'_A, A)}$

$r_B \leftarrow_r \{0,1\}^*, \mathrm{hk}_B \leftarrow_r K, \mathrm{hp}_B = \mathrm{Proj}(\mathrm{hk}_B)$

$\mathrm{label}_B := A \| B \| S \| \mathrm{hp}_B, \mathrm{label}'_B := H_1(\mathrm{lable}_B)$

$c_B := \mathrm{Enc}(\mathrm{pk}, \mathrm{label}_B, \pi_B; r_B)$

$\xrightarrow{M_2=(M_1, \mathrm{hp}_B, c_B, \mathrm{label}'_B, B)}$

$r_{SA} \leftarrow_r \{0,1\}^*, \mathrm{hk}_{SA} \leftarrow_r K, \mathrm{hp}_{SA} = \mathrm{Proj}(\mathrm{hk}_{SA})$

$r_{SB} \leftarrow_r \{0,1\}^*, \mathrm{hk}_{SB} \leftarrow_r K, \mathrm{hp}_{SB} = \mathrm{Proj}(\mathrm{hk}_{SB})$

$\mathrm{label}_{SA} := A \| B \| S \| \mathrm{hp}_{SA}, c_{SA} := \mathrm{Enc}(\mathrm{pk}, \mathrm{label}_{SA}, \mathrm{H}_{SA}; r_{SA})$

$\mathrm{label}_{SB} := A \| B \| S \| \mathrm{hp}_{SB}, c_{SB} := \mathrm{Enc}(\mathrm{pk}, \mathrm{label}_{SB}, \mathrm{H}_{SB}; r_{SB})$

$\mathrm{mk}_{SA} \leftarrow_r \{0,1\}^\ell, \mathrm{mk}_{SB} \leftarrow_r \{0,1\}^\ell$

$m_A = F_{\mathrm{mk}_{SA}}(A \| S) \oplus F_{\mathrm{mk}_{SB}}(B \| S)$

$m_B = F_{\mathrm{mk}_{SB}}(A \| S) \oplus F_{\mathrm{mk}_{SA}}(B \| S)$

$\mathrm{tk}_{SA} = \mathrm{ProjHash}(\mathrm{hp}_A, L_{H_{SA}}, c_{SA}, r_S) \oplus \mathrm{Hash}(\mathrm{hk}_{SA}, L_{(s_1, H_{SA})}, c_A)$

$\mathrm{tk}_{SB} = \mathrm{ProjHash}(\mathrm{hp}_B, L_{H_{SB}}, c_{SB}, r_S) \oplus \mathrm{Hash}(\mathrm{hk}_{SB}, L_{(s_2, H_{SB})}, c_B)$

$\Delta_{SA} = \mathrm{tk}_{SA} \oplus \mathrm{ECC}(\mathrm{mk}_{SA}), \Delta_{SB} = \mathrm{tk}_{SB} \oplus \mathrm{ECC}(\mathrm{mk}_{SB})$

$g_1 = H_1(A, B, S, \Delta_{SA}, \mathrm{ECC}^{-1}(\mathrm{tk}_{SA}), m_A)$

$g_2 = H_1(A, B, S, \Delta_{SB}, \mathrm{ECC}^{-1}(\mathrm{tk}_{SB}), m_B)$

$\xleftarrow{M_3=(M_4=(s_{H_1}, \mathrm{hp}_S, c_{SA}, \Delta_{SA}, g_1, m_A), s_{H_2}, \mathrm{hp}_S, c_{SB}, \Delta_{SB}, g_2, m_B)}$

$P_B \leftarrow \mathrm{PPrehash}(\mathrm{param}, \pi_B, \mathrm{s}_{P_2})$

$H_B \leftarrow \mathrm{PHash}(\mathrm{param}, s_{P_2}, s_{H_2}, P_B)$

$\mathrm{tk}_{SB} = \mathrm{Hash}(\mathrm{hk}_B, L_{H_B}, c_{SB}) \oplus \mathrm{ProjHash}(\mathrm{hp}_S, L_{(s_2, H_B)}, c_B, \pi_B, r_B)$

abort if $g_2 \neq H_1(A, B, S, \Delta_{SB}, \mathrm{ECC}^{-1}(\mathrm{tk}_{SB}), m_B)$

else $\mathrm{mk}_{SB} = \mathrm{ECC}^{-1}(\mathrm{tk}_{SB} \oplus \Delta_{SB})$

$\xleftarrow{M_4=(s_{H_1}, \mathrm{hp}_S, c_{SA}, \Delta_{SA}, g_1, m_A)}$

$P_A \leftarrow \mathrm{PPrehash}(\mathrm{param}, \pi_A, s_{P_1})$

$H_A \leftarrow \mathrm{PHash}(\mathrm{param}, s_{P_1}, s_{H_1}, P_A)$

$\mathrm{tk}_{SA} = \mathrm{Hash}(\mathrm{hk}_A, L_{H_A}, c_{SA}) \oplus \mathrm{ProjHash}(\mathrm{hp}_S, L_{(s_1, H_A)}, c_A, \pi_A, r_A)$

abort if $g_1 \neq H_1(A, B, S, \Delta_{SA}, \mathrm{ECC}^{-1}(\mathrm{tk}_{SA}), m_A)$

else $\mathrm{mk}_{SA} = \mathrm{ECC}^{-1}(\mathrm{tk}_{SA} \oplus \Delta_{SA}), g = m_A \oplus F_{\mathrm{mk}_{SA}}(A \| S)$

$\mathrm{sk}_{AB} = F_{\mathrm{mk}_{SA}}(A \| S) \oplus F_{\mathrm{mk}_{SA}}(B \| S) \oplus m_A$

$\xrightarrow{M_5=(g)}$　abort if $g \neq F_{\mathrm{mk}_{SB}}(B \| S)$

$\mathrm{sk}_{BA} = F_{\mathrm{mk}_{SB}}(A \| S) \oplus F_{\mathrm{mk}_{SB}}(B \| S) \oplus m_B$

图 2　3VPAKE 协议具体描述

的方式选取 r_{SB}、hk_{SB}，并计算 label_{SB}、c_{SB}；选取随机值 $\mathrm{mk}_{SA} \leftarrow_r \{0,1\}^\ell$，$\mathrm{mk}_{SB} \leftarrow_r \{0,1\}^\ell$，计算出 m_A 和 m_B，利用平滑投射哈希函数计算 tk_{SA} 和 tk_{SB} 的值。然后，利用编码算法得到 $\Delta_{SA} = \mathrm{tk}_{SA} \oplus \mathrm{ECC}(\mathrm{mk}_{SA})$ 和 $\Delta_{SB} = \mathrm{tk}_{SB} \oplus \mathrm{ECC}(\mathrm{mk}_{SB})$，之后计算 $\mathrm{ECC}^{-1}(\mathrm{tk}_{SA})$ 和 $\mathrm{ECC}^{-1}(\mathrm{tk}_{SB})$，如果解码为空，则终止；否则计算 $g_1 = H_1(A, B, S, \Delta_{SA}, \mathrm{ECC}^{-1}(\mathrm{tk}_{SA}), m_A)$ 和 $g_2 = H_1(A, B, S, \Delta_{SB}, \mathrm{ECC}^{-1}(\mathrm{tk}_{SB}), m_B)$。$S$ 向 B 发送消息 $M_3 = (M_4 = (s_{H_1}, \mathrm{hp}_S, c_{SA}, \Delta_{SA}, g_1, m_A), s_{H_2}, \mathrm{hp}_S, c_{SB}, \Delta_{SB}, g_2, m_B)$。

4) 用户 B 收到消息后，由 $\mathrm{PPrehash}(\mathrm{param}, \pi_B, s_{P_2})$ 算法生成预哈希值 P_B，再由 $\mathrm{PHash}(\mathrm{param}, s_{P_2}, s_{H_2}, P_B)$ 算法生成哈希值 H_B。利用计算出来的值

和接收到的消息计算 tk_{SB}。用户 B 利用 tk_{SB} 和已知的 Δ_{SB} 和 m_B 计算 $H_1(A,B,S,\Delta_{SB},\mathrm{tk}_{SB},m_B)$，并验证 g_2 与 $H_1(A,B,S,\Delta_{SB},\mathrm{ECC}^{-1}(\mathrm{tk}_{SB}),m_B)$ 的值是否相等，如果不相等，则算法终止；否则进行译码算法得到 $\mathrm{mk}_{SB}=\mathrm{ECC}^{-1}(\mathrm{tk}_{SB}\oplus\Delta_{SB})$。最后，$B$ 将消息 $M_4=(s_{H_1},\ \mathrm{hp}_S,c_{SA},\Delta_{SA},g_1,m_A)$ 发送给用户 A。

5) 用户 A 收到用户 B 发送的消息后，由 $\mathrm{PPrehash}(\mathrm{param},\pi_A,s_{P_1})$ 算法生成预哈希值 P_A，并由 $\mathrm{PHash}(\mathrm{param},s_{P_1},s_{H_1},P_A)$ 算法生成哈希值 H_A。然后计算 tk_{SA} 的值。由计算得出的 tk_{SA} 和已知的 Δ_{SA}，m_A 计算 $H_1(A,B,S,\Delta_{SA},\mathrm{tk}_{SA},m_A)$，并验证 g_1 与 $H_1(A,B,S,\Delta_{SA},\mathrm{ECC}^{-1}(\mathrm{tk}_{SA}),m_A)$ 的值是否相等，如果不相等，则算法终止；否则译码算法计算 $\mathrm{mk}_{SA}=\mathrm{ECC}^{-1}(\mathrm{tk}_{SA}\oplus\Delta_{SA})$。计算 $g=m_A\oplus F_{\mathrm{mk}_{SA}}(A\|S)$，$\mathrm{sk}_{AB}=F_{\mathrm{mk}_{SA}}(A\|S)\oplus F_{\mathrm{mk}_{SA}}(B\|S)\oplus m_A$。最后，$A$ 将消息 $M_5=(g)$ 发送给用户 B。

6) 用户 B 收到消息 M_5 之后，验证 g 的值与 $F_{\mathrm{mk}_{SB}}(B\|S)$ 是否相等，若不相等，则终止；否则计算 $\mathrm{sk}_{BA}=F_{\mathrm{mk}_{SB}}(A\|S)\oplus F_{\mathrm{mk}_{SB}}(B\|S)\oplus m_B$。此时用户 A 和 B 拥有共同的会话密钥。

上述协议中，步骤 3)中服务器对用户的身份间接地进行了验证，步骤 4)～步骤 5)中用户都对服务器身份进行了验证，步骤 6)可以验证用户之间得到了相同的会话密钥。因此协议显式地实现了用户与服务器之间的双向认证，并确认通信用户之间得到了相同的会话密钥。

正确性。协议诚实执行，2 个用户之间得到相同的会话密钥的概率是不可忽略的。根据 ASPH 函数的平滑性，参与者 $A(B)$得到的 tk_{SA}（tk_{SB}）和 S 得到的 tk_{SA}（tk_{SB}）之间的汉明距离至多为 2ϵ。此外，纠错码 ECC 能够纠错 2ϵ 部分，$A(B)$和 S 将得到相同的 mk_{SA}（mk_{SB}），因此 A 和 B 可以得到相同的会话密钥 sk，协议的正确性得到满足。

4 安全性证明

本节在上述安全模型中证明本文 3VPAKE 协议的安全性。

定理 1 如果 $\mathcal{PKE}=(\mathrm{KeyGen},\mathrm{Enc},\mathrm{Dec})$ 为基于格的 CCA 安全的公钥加密体制；$(K,\ell,\{\mathrm{Hash}_{\mathrm{hk}}:X\to\{0,1\}^{\ell}\}_{\mathrm{hk}\in K},\mathrm{HP},\mathrm{Proj}:K\to\mathrm{HP})$ 为相应的 ASPH 函数，其中，HP 为投射密钥空间；$\mathcal{H}=(\mathrm{Setup},\ \mathrm{PreSalt},\mathrm{PreHash},\ \mathrm{Salt},\mathrm{Hash})$ 为基于格的随机口令哈希方案；$\mathrm{ECC}:\{0,1\}^{\ell}\to\{0,1\}^{\kappa}$ 为可以纠错 2ϵ 小部分的纠错码；$F=\{F_{\mathrm{mk}}\}$ 为安全的伪随机函数，则 3VPAKE 协议是语义安全的。

证明 假设模拟器控制了概率多项式敌手 $\mathcal{A}$ 访问的所有预言机。若在 Test 询问中敌手 $\mathcal{A}$ 猜对模拟器选择的 b 值，则称敌手成功。为了形式化地证明定理 1，通过一系列游戏来估计敌手的优势，G_0 是语义安全时的真实攻击游戏，从 G_0 到 G_9 敌手的优势之差至多为 $\frac{Q(\kappa)}{|\mathcal{D}|}+\mathrm{negl}(\kappa)$。定义 $\mathrm{Adv}_{\mathcal{A},i}(\kappa)$ 表示敌手在游戏 G_i 中的优势。

游戏 G_1。修改对口令 Hash 进行 Execute 询问的响应。修改存储的预盐值 s_P、盐值 s_H 和哈希值 H，根据盐值的不可区分性，可知敌手在游戏 G_1 和游戏 G_0 中的优势之差可忽略，则有 $\left|\mathrm{Adv}_{\mathcal{A},1}(\kappa)-\mathrm{Adv}_{\mathcal{A},0}(\kappa)\right|=0$。

游戏 G_2。修改 Execute 询问的响应，替换密文 c_A 为加密虚拟口令得到的密文。

由于 $\mathcal{PKE}=(\mathrm{KeyGen},\mathrm{Enc},\mathrm{Dec})$ 为基于格的 CCA 安全加密方案，因此存在 $|\mathrm{Adv}_{\mathcal{A},2}(\kappa)-\mathrm{Adv}_{\mathcal{A},1}(\kappa)|\leqslant\mathrm{negl}(\kappa)$。

游戏 G_3。继续修改 Execute 询问的响应，替换密文 $c_{SA}=\mathrm{Enc}(\mathrm{pk},\mathrm{label}_{SA},H_{SA};r_{SA})$ 为虚拟口令 π_0 相应验证元的密文 $c_{SA}=\mathrm{Enc}(\mathrm{pk},\mathrm{label}_{SA},H_0;r_{SA})$。

根据定义 3 中判定 LWE 问题困难性假设成立，公钥加密算法是 CCA 安全的，因此能够区分 H_{SA} 和 H_0 相应的密文的概率是可忽略的，得到敌手在游戏 G_3 和游戏 G_2 优势之差可忽略，存在 $|\mathrm{Adv}_{\mathcal{A},3}(\kappa)-\mathrm{Adv}_{\mathcal{A},2}(\kappa)|\leqslant\mathrm{negl}(\kappa)$。

游戏 G_4。继续修改 Execute 询问的响应，修改值 tk_{SA}，该值由相应的哈希密钥 hp_A、hk_S 和相应的密文进行计算，即 $\mathrm{tk}_{SA}=\mathrm{Hash}(\mathrm{hk}_S,L_{(s_{H_1},H_{SA})},c_A)\oplus\mathrm{ProjHash}(\mathrm{hp}_A,L_{H_{SA}},c_S,r_S)$。

根据判定性 LWE 问题难解性和 ASPH 函数的平滑性得知，对于 $\mathrm{hk}=(\boldsymbol{g}_1,\cdots,\boldsymbol{g}_{\ell})$，$\mathrm{hp}=\boldsymbol{u}_i=\boldsymbol{B}'^{\mathrm{T}}\boldsymbol{g}_i=\mathrm{Proj}(\mathrm{hk})$，哈希值 $z_i=\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{s}$ 和 $z_i'\leftarrow\mathbb{Z}_q^{\ell}$，分布 $(\boldsymbol{B}'^{\mathrm{T}}\boldsymbol{g}_i,\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{s})$ 和 $\mathbb{Z}_q^n\times\mathbb{Z}_q$ 上的均匀分布是不可区分的。因此得到敌手在游戏 G_4 和 G_3 中的优势之差可忽略，在判定 LWE 困难问题假设下，存在 $|\mathrm{Adv}_{\mathcal{A},4}(\kappa)-\mathrm{Adv}_{\mathcal{A},3}(\kappa)|\leqslant\mathrm{negl}(\kappa)$。

游戏 G_5。继续修改 Execute 询问的响应，修改 m_A，将其替换为均匀选取的随机值。根据伪随机函数簇的性质，得到敌手在游戏 G_5 和 G_4 优势之差可忽略，存在 $|\mathrm{Adv}_{\mathcal{A},5}(\kappa)-\mathrm{Adv}_{\mathcal{A},4}(\kappa)|\leqslant \mathrm{negl}(\kappa)$。

至此已经完成了 Execute 查询的修改，主要以用户 A 为例，敌手不能从这些攻击中获得任何信息。下面考虑 Send 询问，将 Send 询问分为以下几类。

$\mathrm{Send}_0(A,i_1,B,i_2,S,j)$。敌手激活协议，得到用户发送的第一轮消息。

$\mathrm{Send}_1(S,j,\mathrm{msg}_1)$。敌手给服务器发送一轮消息，得到服务器返回的下一轮消息。

$\mathrm{Send}_2(B,i_2,\mathrm{msg}_2)$。敌手向用户实例 $\prod_B^{i_2}$ 发送下一轮消息。

$\mathrm{Send}_3(A,i_1,\mathrm{msg}_3)$。敌手向用户实例 $\prod_A^{i_1}$ 发送下一轮消息。

游戏 G_6。修改 Send_1 询问，通过解密预言机或已知的解密密钥进行修改。敌手给服务器发送消息 $\mathrm{msg}_1=M_2=(M_1,\mathrm{hp_B},c_B,\mathrm{lable}_B^{'})$，如果 S 端的 $H_{SB}(H_{SA})$ 被泄露，则诚实地回复查询，否则，存在以下 2 种情况。

情况 1　消息 msg_1 被敌手生成或修改，解密密文得到前哈希值 $\pi_B(\pi_A)$。

① 利用服务器的 $(s_{P_2},s_{H_2},\mathrm{H}_{SB})$，如果 $\mathrm{Check}(\mathrm{pp},s_{P_2},s_{H_2},\pi_B,H_{SB})=1$，且 $\mathcal{A}$ 访问 Test 询问到会话密钥，则敌手成功并终止模拟 tk_{SB}；②均匀选取。

情况 2　如果是对之前的消息进行重放，那么可以利用密文 c_S 和密钥 hk_B 计算临时会话密钥 tk_{SB}。

情况 1 的①中，敌手的优势增加了可忽略部分；根据口令策略检查的正确性，词语不在 ASPH 函数的语言 L 中，根据判定 LWE 问题的困难性和 ASPH 函数的平滑性，情况 1 的②中变化是不可区分的，所增加的优势是可忽略的。情况 2 并不影响敌手的优势。因此存在 $\mathrm{Adv}_{\mathcal{A},5}(\kappa)\leqslant \mathrm{Adv}_{\mathcal{A},6}(\kappa)+\mathrm{negl}(\kappa)$。

游戏 G_7。修改 Send_2 询问，服务器发送消息 $\mathrm{msg}_2=(s_{H_2},\mathrm{hp}_S,c_{SB},\Delta_{SB},g_2,m_B)$，其中的密文 c_{SB} 是由敌手伪造的，那么存在以下 2 种情况。

情况 1　如果实例 $\prod_B^{i_2}$ 持有的口令和实例 $\prod_S^{j}$ 持有的哈希值相对应，则令用户 B 与服务器拥有相同的 tk_{SB}，即令 $\mathrm{sk}_{AB}=\mathrm{sk}_{BA}$。

情况 2　均匀选取 sk。

情况 1 增加了敌手的优势；根据 ASPH 函数的平滑性可知，分布 $(\boldsymbol{B}'^{\mathrm{T}}\boldsymbol{g}_i,\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{s})$ 和 $\mathbb{Z}_q^n\times\mathbb{Z}_q$ 上的均匀分布是不可区分的；情况 2 的优势至多为解决判定 LWE 问题的优势，因此并不增加敌手的优势，那么得到 $|\mathrm{Adv}_{\mathcal{A},7}(\kappa)-\mathrm{Adv}_{\mathcal{A},6}(\kappa)|\leqslant \mathrm{negl}(\kappa)$。

游戏 G_8。对 Send_2 询问进行修改，服务器 S 向用户发送消息 $\mathrm{msg}_2=M_3=(s_{H_1},s_{H_2},\mathrm{hp}_S,c_{SA},c_{SB},\Delta_{SA},\Delta_{SB},g_1,g_2,m_A,m_B)$，如果泄露了 B 的口令 π_B，那么利用该口令诚实地回答并计算 sk；否则，存在以下情况。

S 收到 B 发送的消息之后，返回消息 $\mathrm{msg}_2=M_3$，然后敌手解密 c_{SB} 计算出哈希值 H_{SB}。

①如果口令与 (s,H) 一致，即 $H_{S_2}=\mathrm{PHash}(\mathrm{pp},s_{P_2},s_{H_2},P_B)$，并且随后攻击者通过 Test 得知会话密钥，那么认为敌手成功并结束模拟。

②均匀随机选取 sk。

③如果消息 msg_2 是被服务器生成的，实例 $\prod_S^{j}$ 和 $\prod_B^{i_2}$ 互为匹配会话，则令它们拥有相同的 tk_{SB}，即令 $\mathrm{sk}_{AB}=\mathrm{sk}_{BA}$。

④均匀随机选取 sk。

敌手的优势在①的变化只增加可忽略部分；根据 ASPH 函数的平滑性，在判定 LWE 困难问题假设下，②和④中的变化是不可区分的，敌手的优势只增加了可忽略的部分；③的变化不影响敌手的优势。因此得到 $\mathrm{Adv}_{\mathcal{A},7}(\kappa)\leqslant \mathrm{Adv}_{\mathcal{A},8}(\kappa)+\mathrm{negl}(\kappa)$。

游戏 G_9。修改 Send_3 询问，服务器发送消息 $\mathrm{msg}_3=(s_{H_1},\mathrm{hp}_S,c_{SA},\Delta_{SA},g_1,m_A)$，其中的密文 c_{SA} 是由敌手伪造的，那么存在以下 2 种情况。

情况 1　如果实例 $\prod_A^{i_1}$ 持有的口令和实例 $\prod_S^{j}$ 持有的哈希值相对应，则令用户 A 与服务器拥有相同的 tk_{SA}，即令 $\mathrm{sk}_{AB}=\mathrm{sk}_{BA}$。

情况 2　均匀选取 sk。

情况 1 增加了敌手的优势，根据 ASPH 函数的平滑性，分布 $(\boldsymbol{B}'^{\mathrm{T}}\boldsymbol{g}_i,\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{s})$ 和 $\mathbb{Z}_q^n\times\mathbb{Z}_q$ 上的均匀分布是不可区分的；情况 2 的优势至多为解决判定 LWE 问题的优势，因此并不增加敌手的优势，那么得到 $|\mathrm{Adv}_{\mathcal{A},9}(\kappa)-\mathrm{Adv}_{\mathcal{A},8}(\kappa)|\leqslant \mathrm{negl}(\kappa)$。

游戏 G_{10}。修改 Send_0 询问，将密文 $c_A=\mathrm{Enc}(\mathrm{pk},\mathrm{label}_A,\pi_A;r_A)$ 替换成对虚拟口令 π_0 进行加密生成密文 c'_A，c'_A 是对 Send_0 询问的回应。

由于 $\mathcal{PKE}$ 是基于 LWE 的 CCA 安全的公钥加密

体制，区分口令 π_A 和 π_0 相应密文的优势至多为解决判定 LWE 困难问题的优势，因此敌手在游戏 G_{10} 和 G_9 中的优势之差可忽略。在判定性 LWE 困难问题假设下，可知 $|\mathrm{Adv}_{\mathcal{A},10}(\kappa)-\mathrm{Adv}_{\mathcal{A},9}(\kappa)|\leqslant \mathrm{negl}(\kappa)$。

综上所述，游戏 $G_0 \sim G_{10}$ 中，敌手至多可进行 $Q(\kappa)$ 次在线攻击，由此可得 $\mathrm{Adv}_{\mathcal{A},0}(\kappa)\leqslant \frac{Q(\kappa)}{|\mathcal{D}|}+\mathrm{negl}(\kappa)$，定理 1 结论成立，可知 3VPAKE 协议是语义安全的。证毕。

5 性能比较

本文协议和已有的口令认证密钥交换协议[15-16,20,22-23]从安全性和通信效率 2 个方面进行比较，其中，$n_1 < n_2$，$m = O(n\log q) \in \mathbb{Z}$，具体如表 2 所示。

在安全性方面，本文协议基于 LWE 问题构造，在量子计算下还不存在多项式求解算法，此外本文协议可降低服务器信息泄露带来的危害，抵抗不可检测在线字典攻击，并且提供相互认证的功能。文献[20]是基于验证元的 2PAKE 协议（即 2VPAKE 协议），依赖于传统的数学困难问题假设，在量子计算机下是不安全的，且不能抵抗不可检测在线字典攻击。该协议是针对两方设计的，消息传输量为 2 条，所以 3PAKE 协议消息传输量至少需要 4 条，且其通信效率也较低，不能达到双向认证的功能。文献[15]是基于格上困难问题设计的 3PAKE 协议，消息传输量为 6 条，通信量较多且不能抵抗服务器泄露攻击。文献[16]是格上用户匿名的 3PAKE 协议，消息传输量较少，但通信效率较低并且没有提供抵抗服务器泄露的性质。文献[22]是一种改进的 3VPAKE 协议，该协议基于 DDH 假设，消息传输量为 8 条，通信效率较低。文献[23]是新的理想格上的 3VPAKE 协议，可抵抗服务器泄露攻击，消息传输量为 8 条，通信负担较大。

与现有方案[15-16,20,22-23]相比，本文方案是基于 LWE 问题的三方 VPAKE 协议，具有抵抗已知量子攻击的特点，适用于大规模用户端到用户端的通信系统。本文方案只传输 5 条消息量，协议中的投射密钥只依赖于哈希密钥，通信效率较高且具有抵御不可检测在线字典攻击和服务器泄露攻击的特点。

在计算开销方面，给出本文协议与同类型三方协议[14,21-22]的比较，如表 3 所示。这些协议都采用了平滑投射哈希函数，Exp 表示模幂运算，Enc/Dec 表示公钥加密/解密运算，ASPH(SPH)表示近似平滑投射哈希运算，Hash 表示哈希运算。由于指数运算成本要比哈希运算成本高，因此文献[14]协议和本

表 2 安全性和通信开销比较

协议	类型	消息传输量/条	通信效率	双向认证	抵抗服务器泄露	难题假设
文献[15]	3PAKE	6	$9n_2+7n_2\log q+5$	是	否	RLWE
文献[16]	3PAKE	5	$4n_2\log q+7n_2$	是	否	RLWE
文献[20]	2VPAKE	2	$10n_2+2n_1$	否	是	plain DDH
文献[22]	3VPAKE	8	$8n_2+8n_1$	是	是	DDH
文献[23]	3VPAKE	8	$6n_2\log q+7n_2$	是	是	RLWE
本文协议	3VPAKE	5	$2n_2+4(m+n_2)\log q$	是	是	LWE

表 3 计算开销比较

协议	工具	Exp	Enc/Dec	ASPH(SPH)	Hash	总开销
文献[21]	SPH	25	2	2	6	25Exp+1Enc+1Dec+2SPH+6Hash
文献[22]	SPH	18	4	4	8	18Exp+2Enc+2Dec+4SPH+8Hash
文献[14]	ASPH	0	4	4	24	2Enc+2Dec+4ASPH+24Hash
本文协议	ASPH	0	4	4	26	2Enc+2Dec+4ASPH+26Hash

文协议计算开销较小且相当。但是文献[14]协议不能抵抗服务器信息泄露攻击。

相比较而言，本文提出的 3VPAKE 协议具有更高的通信效率和较低的计算成本，在实际应用中更具有可行性。

6　结束语

本文提出了一种新的格上基于验证元的三方 PAKE 协议。利用基于格的公钥加密和平滑投射哈希函数，并采用口令哈希方案和零知识口令策略检查有效实现了口令的机密性和口令的检查，达到了抵抗服务器泄露攻击的目的，最后给出了形式化安全性分析。本文提出的 3VPAKE 协议减少了用户存储口令的负担，适用于大规模用户相互通信场景，且可抵抗不可检测在线字典攻击和已知量子攻击等安全属性。会话密钥私密性在三方 PAKE 是较为重要的安全属性，未来将考虑设计后量子格基 3PAKE 方案，如何能够在不实质增加通信量和计算开销的同时满足该性质，为基于格的 3VPAKE 协议做出进一步的研究。

参考文献：

[1] BELLOVIN S M, MERRITT M. Encrypted key exchange: password-based protocols secure against dictionary attacks[C]// Proceedings 1992 IEEE Computer Society Symposium on Research in Security and Privacy. Piscataway: IEEE Press, 1992: 72-84.

[2] VASCO M I G, POZO A L P D, SORIENTE C. A key for John Doe: modeling and designing anonymous password-authenticated key exchange protocols[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 18(3): 1336-1353.

[3] BRADLEY T, JARECKI S, XU J. Strong asymmetric PAKE based on trapdoor CKEM[C]//2019 Advances in Cryptology. Berlin: Springer, 2019: 798-825.

[4] ABDALLA M, BARBOSA M, BRADLEY T, et al. Universally composable relaxed password authenticated key exchange[C]//2020 Advances in Cryptology. Berlin: Springer, 2020: 278-307.

[5] KATZ J, VAIKUNTANATHAN V. Round-optimal password-based authenticated key exchange[J]. Journal of Cryptology, 2013, 26(4): 714-743.

[6] KATZ J, VAIKUNTANATHAN V. Smooth projective hashing and password-based authenticated key exchange from lattices[C]//2009 Advances in Cryptology. Berlin: Springer, 2009: 636-652.

[7] GENNARO R, LINDELL Y. A framework for password-based authenticated key exchange1[J]. ACM Transactions on Information and System Security (TISSEC), 2006, 9(2): 181-234.

[8] DING Y, FAN L. Efficient password-based authenticated key exchange from lattices[C]//2012 Seventh International Conference on Computational Intelligence and Security. Piscataway: IEEE Press, 2012: 934-938.

[9] GROCE A, KATZ J. A new framework for efficient password-based authenticated key exchange[C]//Proceedings of the 17th ACM conference on Computer and communications security. New York: ACM Press, 2010: 516-525.

[10] DING J, ALSAYIGH S, LANCRENON J, et al. Provably secure password authenticated key exchange based on RLWE for the post-quantum world[C]//2017 Cryptographers' Track at the RSA Conference. Berlin: Springer, 2017: 183-204.

[11] ZHANG J, YU Y. Two-round PAKE from approximate SPH and instantiations from lattices[C]//2017 International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2017: 37-67.

[12] 李子臣, 谢婷, 张卷美. 基于 RLWE 问题的后量子口令认证密钥交换协议[J]. 电子学报, 2021, 49(2):260-267.
LI Z C, XIE T, ZHANG J M. Post quantum password-based authentication key exchange protocol based on ring learning with errors problem[J]. Acta Electronica Sinica, 2021, 49(2): 260-267.

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

[14] 叶茂, 胡学先, 刘文芬. 基于格的三方口令认证密钥交换协议[J]. 电子与信息学报, 2013, 35(6): 1376-1381.
YE M, HU X X, LIU W F. Password authenticated key exchange protocol in the three party setting based on lattices[J]. Journal of Electronics & Information Technology, 2013, 35(6):1376-1381.

[15] XU D Q, HE D B, CHOO K K R. Provably secure three-party password authenticated key exchange protocol based on ring learning with error[R]. 2017.

[16] 王彩芬, 陈丽. 基于格的用户匿名三方口令认证密钥协商协议[J]. 通信学报, 2018, 39(2): 21-30.
WANG C F, CHEN L. Three-party password authenticated key agreement protocol with user anonymity based on lattice[J]. Journal on Communications, 2018, 39(2): 21-30.

[17] 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): 87-97.
YU J X, LIAN H H, TANG Y L, et al. Password-based three-party authenticated key exchange protocol from lattices[J]. Journal on Communications, 2018, 39(11): 87-97.

[18] JIANG S, GONG G, HE J, et al. PAKEs: new framework, new techniques and more efficient lattice-based constructions in the standard model[C]//2020 IACR International Conference on Public-Key Cryptography. Berlin: Springer, 2020: 396-427.

[19] BELLOVIN S M, MERRITT M. Augmented encrypted key exchange: a password-based protocol secure against dictionary attacks and password file compromise[C]//Proceedings of the 1st ACM conference on Computer and communications security. New York: ACM Press, 1993: 244-250.

[20] BENHAMOUDA F, POINTCHEVAL D. Verifier-based password-authenticated key exchange: new models and constructions[R]. 2014.

[21] 杨晓燕, 侯孟波, 魏晓超. 基于验证元的三方口令认证密钥交换协议[J]. 计算机研究与发展, 2016, 53(10): 2230-2238.

YANG X Y, HOU M B, WEI X C. Verifier-based three-party password authenticated key exchange protocol[J]. Journal of Computer Research and Development, 2016, 53(10): 2230-2238.

[22] 张启慧, 胡学先, 刘文芬, 等.改进的三方口令验证元认证密钥交换协议[J].软件学报, 2020, 31 (10): 3238-3250.
ZHANG Q H, HU X X, LIU W F, et al. Improved verifier-based three-party password-authenticated key exchange protocol[J]. Journal of Software, 2020, 31 (10): 3238-3250.

[23] 舒琴, 王圣宝, 胡斌, 等. 理想格上基于验证元的三方口令认证密钥交换协议[J]. 密码学报, 2021, 8(2): 294-306.
SHU Q, WANG S B, HU B, et al. Verifier-based three-party password-authenticated key exchange protocol from ideal lattices[J]. Journal of Cryptologic Research, 2021, 8(2): 294-306.

[24] ABDALLA M, FOUQUE P A, POINTCHEVAL D. Password-based authenticated key exchange in the three-party setting[C]//2005 International Workshop on Public Key Cryptography. Berlin: Springer, 2005:65-84.

[25] REGEV O. On lattices, learning with errors, random linear codes, and cryptography[J]. Journal of the ACM, 2009, 56(6): 1-40.

[26] CRAMER R, SHOUP V. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption[C]// International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology. Berlin: Springer, 2002: 45-64.

[27] KIEFER F, MANULIS M. Zero-knowledge password policy checks and verifier-based PAKE[C]//2014 European Symposium on Research in Computer Security. Berlin: Springer, 2014: 295-312.

[28] NGUYEN K, TAN B H M, WANG H X. Zero-knowledge password policy check from lattices[C]//2017 International Conference on Information Security. Berlin: Springer, 2017: 92-113.

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

[作者简介]

廉欢欢（1993− ），女，河南沁阳人，复旦大学博士生，主要研究方向为密码学、信息安全等。

侯慧莹（1992− ），女，山东济宁人，复旦大学博士生，主要研究方向为应用密码学和信息安全等。

赵运磊（1974− ），男，山东阳谷人，博士，复旦大学特聘教授、博士生导师，主要研究方向为后量子密码、密码协议和计算理论等。