高效可扩展的隐私集合交集基数方案研究

李功丽[1]，刘威辰[1]，郑东[2]
（1.河南师范大学计算机与信息工程学院，河南 新乡 453007；2.西安邮电大学网络空间安全学院，陕西 西安 710061）

摘　要：针对现有两方隐私集合交集基数（PSI-CA）方案计算开销大的问题，提出了一种高效的两方 PSI-CA 协议。该协议利用不经意键值存储（OKVS）和不经意密钥共享伪随机函数（OKS-PRF）隐藏交集元素信息，能显著优化协议的执行时间，同时可扩展到多方 PSI-CA 场景。实验结果表明，当集合大小为 2^{20} 时，所提两方 PSI-CA 协议能够在 36.61 s 内完成，执行速度是目前最快两方协议的 1.8 倍。当参与方数量为 2^3，集合大小为 2^{20} 时，所提多方 PSI-CA 协议可在 115.32 s 内完成，并能抵抗 $N-2$ 个参与方合谋。

关键词：隐私集合交集基数；抗合谋；不经意键值存储；不经意密钥共享伪随机函数
中图分类号：TP309
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2025085

Research on efficient and scalable private set intersection cardinality schemes

LI Gongli[1], LIU Weichen[1], ZHENG Dong[2]
1. School of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China
2. School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710061, China

Abstract: To address the issue of high computational overhead in existing two-party private set intersection cardinality (PSI-CA) protocols, an efficient two-party PSI-CA protocol was proposed. Oblivious key-value store (OKVS) and oblivious key-sharing pseudorandom function (OKS-PRF) were leveraged by this protocol to hide information of intersection elements, with execution time being significantly optimized. Moreover, the protocol could be extended to multi-party PSI-CA scenarios. Experimental results show that when the set size is 2^{20}, the proposed two-party PSI-CA protocol completes in 36.61 seconds, and the execution speed is 1.8 times than the fastest existing two-party protocol. When the number of participants is $2^3$and the set size is 2^{20}, the proposed multi-party PSI-CA protocol completes in 115.32 seconds and can resist collusion among up to $N-2$ participants.

Keywords: private set intersection cardinality, collusion-resistance, oblivious key-value store, obliviou key-sharing pseudorandom function

0　引言

隐私集合交集（PSI, private set intersection）允许参与方联合计算交集，而不泄露交集以外的任何信息。在数据由不同管理者持有的情况下，PSI 技术实现了隐私保护和信息共享的双赢。随着时间的推移，PSI 技术越来越成熟，并且出现了许多重要的密码学原语。隐私集合交集基数（PSI-CA, private set intersection cardinality）是 PSI 的一种变体

收稿日期：2025–01–06；修回日期：2025–04–25
通信作者：郑东，zhengdong@xupt.edu.cn
基金项目：国家自然科学基金资助项目（No.62372157）；河南省科技攻关计划基金资助项目（No.232102211057）
Foundation Items：The National Natural Science Foundation of China (No.62372157), The Scientific and Technological Research Project of Henan Province (No.232102211057)

形式，仅允许参与方计算交集的大小。相比PSI而言，PSI-CA还要保护交集元素，泄露的信息更少。随着对隐私保护需求的持续增长，PSI-CA技术被不断地应用到许多现实场景中，如保护隐私的接触者追踪[1-3]、隐私记录链接[4]和数据相似性评估[5]等。诸如此类的现实应用场景都仅计算交集的大小，而不泄露交集元素。

目前，针对两方PSI-CA的研究大多是利用同态加密或基于密钥交换的不经意伪随机函数（OPRF, oblivious pseudorandom function）实现的。这些方法都需要在明文数据上执行幂运算，开销较大。近年来，许多方案选择将加密后的数据委托给云服务器来完成PSI-CA协议，但这种方案基于一个强假设，即云服务器是可信的。实际上云服务器可能存在恶意行为，甚至可能与某一参与方合谋，导致信息泄露或计算结果不准确。在现实生活中，凡触及敏感信息的场景，如银行与保险公司利用PSI-CA技术进行金融风险分析，找到一个完全可信的第三方并不容易。对此，本文使用开销较小的对称加密技术，同时不依赖任何额外的第三方（云服务器），设计实现了高效且可扩展的PSI-CA协议，主要贡献如下。

1) 利用不经意密钥共享伪随机函数（OKS-PRF, oblivious key-sharing pseudorandom function）和不经意键值存储（OKVS, oblivious key-value store）密码学原语提出了一种不依赖任何第三方的两方隐私集合交集基数（TPSI-CA, two-party private set intersection cardinality）协议。该协议不需要对明文数据执行大量的幂运算，且协议的计算开销和通信开销与集合大小线性相关。

2) 将TPSI-CA协议进一步扩展到多方场景，构建了多方隐私集合交集基数（MPSI-CA, multi-party private set intersection cardinality）协议。该协议能够有效实现多个参与方之间交集基数的计算，同时在保证交集数据隐私的情况下，可抵抗$N-2$个参与方的合谋。

3) 实验结果表明，在集合大小为2^{20}时，TPSI-CA协议能够在36.61 s内完成计算，执行速度是目前最快的两方PSI-CA协议的1.8倍。在参与方数量为8、集合大小为2^{20}时，MPSI-CA协议可在115.32 s内完成，与目前的多方PSI-CA协议相比，不依赖第三方的MPSI-CA协议具有更好的实用性。

1　相关工作

随着人们隐私保护意识的增强，隐私计算技术[6]被广泛关注，PSI作为其中的重要研究对象之一，允许参与方在不泄露任何额外信息的情况下获得交集元素。Freedman等[7]和Kissner等[8]利用不经意多项式评估（OPE, oblivious polynomial evaluation）和同态加密技术提出了两方PSI协议，然而当面临较大的集合时，其大量的高阶多项式运算会导致计算开销急剧增加。随后具有线性复杂度的两方PSI[9-17]协议相继被提出。Hazay等[9-10]给出了代数OPRF，通过使发送方获得密钥，接收方获得输入元素x对应的函数值来构造有效的PSI协议。Kerschbaum[14]、Dong等[15]和Debnath等[16-17]皆利用布隆过滤器提出了一系列的PSI协议，其中文献[15-17]可抵抗恶意敌手。近几年，关于PSI[18-25]的密码学技术相继出现。其中，Kolesnikov等[18]首先基于不经意传输（OT, oblivious transfer）提出了单点OPRF的高效构造方法，大幅提升了PSI协议的执行效率，并在文献[23]中提出了有条件零共享与无条件零共享设计方案，可以有效抵抗多个参与方之间的合谋。随后，Pinkas等[21]在OT方案上再次扩展，开发出了多点OPRF协议，进一步丰富了PSI方案的技术路线。接着，Garimella等[20]提出了新的OKVS编码方案，可将多个键值对编码到一个紧凑的数据结构中，为高效PSI协议的设计开辟了新途径。此后，Raghuraman等[22]利用带状矩阵，优化了OKVS的求解过程，从而大幅缩短了编解码时间，显著提升了PSI方案的整体性能。

PSI-CA作为PSI的一种变体形式，仅允许参与方获得交集的大小，而不泄露交集元素本身，这一概念最早由Agrawal等[26]提出。起初，Hohenberger等[27]和Camenisch等[28]也是利用OPE的方式计算交集的大小，但同样会导致较高的计算开销。随后，Cristofaro等[29]和Debnath等[30]分别基于密钥交换的OPRF和布隆过滤器提出了具有线性复杂度的两方PSI-CA协议，其中文献[30]可在Decisional Diffie-Hellman假设下实现恶意安全。接着，Garimella等[31]提出了一种基于不经意转换的隐私集合操作（PSO, private set operation），可分别实现隐私集合并集（PSU, private set union）和PSI-CA。最近，Chen等[32]构造了多查询反向私有成员测试

（mqRPMT, multi-query reverse private membership test）协议，可实现两方PSI-CA，也是目前在半诚实模型下最快的两方PSI-CA。与此同时，一些设备没有较高的计算能力，于是基于云辅助的两方PSI-CA协议相继被提出。参与方可把大量的数据委托给云服务器进行计算，这大大减轻了客户端的负担。在传输云端前，客户端分别使用不同的方式对本地数据进行掩码。其中，Tajima等[33]使用同态的方式对本地数据进行加密，Yang等[34]和Duong等[35]使用对称加密技术并借助多个云服务器来完成协议，Gao等[36]利用云服务器提出了一种新的快速洗牌实现两方PSI-CA，是目前在半诚实模型下借助单云的最快两方PSI-CA协议。在引入云服务器的同时，需要考虑云可能存在的恶意行为，于是Chen等[2]提出了一种高效验证云辅助的PSI-CA协议。

以上是针对两方PSI-CA协议的研究，在许多实际应用场景中，多方PSI-CA协议可以更广泛地适用于各种联合分析和数据共享的场景。Kissner等[8]最早提出的多方PSI-CA也是基于OPE的方式实现的。Debnath等[37]和Jolfaei等[38]使用的方法很相似，二者分别利用阈值同态加密和同态加密技术构造掩码的布隆过滤器。Wu等[39]借助累加器技术，可减少客户端在线交互轮次，当有过大的集合参与时，协议的在线时间将急剧增加，因此该方案并不实用。于是，Yang等[40]和Gao等[36]开始将对称加密技术应用到多方PSI-CA协议中。其中，文献[40]构造了一种零共享的混淆布隆过滤器，可以允许$N-1$个参与方合谋，文献[36]提出了一种依赖第三方快速洗牌的方式实现多方PSI-CA，这是目前最快的多方PSI-CA协议。

针对两方PSI-CA协议计算开销大且大多依赖云服务器的问题，本文基于对称加密技术，提出了一种新的TPSI-CA协议，实现了无云辅助下对PSI-CA的高效计算，并将其扩展为多方，然后与目前最先进的协议进行了对比分析。

2　预备知识

本节介绍协议用到的密码学原语和相关知识，包括OKVS、OT、OKS-PRF和安全模型。为了更好地进行描述，将协议用到的参数及其含义在表1中说明。

表1　参数及其含义

参数	含义
OT_n^κ	执行κ次n bit的OT操作
$i\in[n],i\in[2,n]$	i的取值范围是1,2,⋯,n和2,3,⋯,n
$w\|\|u$	将w和u这2个数据连接在一起
X_i	第i个参与方的集合
x_j^i	集合X_i中的第j个元素

2.1　不经意键值存储

不经意键值存储[20,22]由两部分组成，分别是编码Enc阶段和解码Dec阶段。编码是从键值域$K\times V$中取出多组键值对(k_i,v_i)作为输入，对象T为输出（或一个可忽略的误差指示符⊥）。解码是在T的基础上输入一个p，并输出一个v，具体介绍如下所示。

1) Enc。对于一组拥有n个键值对的集合$S\{(k_1,v_1)\cdots(k_i,v_i)\cdots(k_n,v_n)\}$，可编码$T\leftarrow\mathrm{Enc}(S)$（或一个可忽略的误差指示符⊥）。

2) Dec。对于一个元素p，解码得到$v=\mathrm{Dec}(T,p)$，如果$p=k_i$，则$v=v_i$，否则v是一个随机值。

编码阶段可以理解为将n个键值对$\{(k_1,v_1)\cdots(k_i,v_i)\cdots(k_n,v_n)\}$进行多项式插值，得到插值后的系数方程就是对象$T$。解码阶段就是把元素$p$代入系数方程中计算的结果，如果$p=k_i$，则计算结果为$v_i$，否则就是一个随机值。

2.2　不经意传输

不经意传输是一种密码学原语，包括2个参与方，发送方S拥有数据(x_0,x_1)，接收方R拥有选择位$b\in\{0,1\}$。最终，R学习到x_b，且无法获取x_{b-1}，同时S对R选择位b一无所知，F_{OT}的理想功能函数如图1所示。

参数：发送方S拥有数据(x_0,x_1)，接收方R拥有选择位$b\in\{0,1\}$。

协议：

1) S输入(x_0,x_1)，R输入$b\in\{0,1\}$。

2) R得到x_b。

图1　F_{OT}的理想功能函数

2.3　不经意密钥共享伪随机函数

不经意密钥共享伪随机函数[25]是一个两方交互的协议，一个作为接收方R，输入集合$Y\{y_1,y_2,\cdots,y_n\}$，另一个作为发送方S，选择密钥K

和s作为输入。最终，R获得输入集合Y中每个元素对应的函数值，而S无法获知R的输入和输入元素对应的函数值。同时，R对密钥K和s也一无所知，OKS-PRF协议的详细过程介绍如下。

参数。S作为发送方，选择密钥K和$s \in \{0,1\}^{\kappa}$作为输入。R作为接收方，拥有集合$Y\{y_1,y_2,\cdots,y_n\}$，伪随机函数$F:\{0,1\}^{*}\times\{0,1\}^{*}\to\{0,1\}^{\kappa}$，功能函数$C:\{0,1\}^{*}\to\{0,1\}^{\kappa}$，其中$\kappa$为安全参数。

1) S随机选择密钥K和$s \in \{0,1\}^{\kappa}$。

2) R随机选择密钥k'，并根据所有的$y_{i\in[n]}\in Y\{y_1,y_2,\cdots,y_n\}$生成2个$n\times\kappa$的矩阵$\boldsymbol{A}$和矩阵$\boldsymbol{B}$，其中矩阵$\boldsymbol{A}$的第$i$行为$\boldsymbol{A}_i=\boldsymbol{F}(\boldsymbol{k}',\boldsymbol{y}_i)$，矩阵$\boldsymbol{B}$的第$i$行为$\boldsymbol{B}_i=\boldsymbol{F}(\boldsymbol{k}',\boldsymbol{y}_i)\oplus\boldsymbol{C}(\boldsymbol{y}_i)$，$\boldsymbol{A}^j$和$\boldsymbol{B}^j$分别表示矩阵$\boldsymbol{A}$和矩阵$\boldsymbol{B}$的第$j$列。

3) S和R调用OT_n^{κ}，S作为接收方输入s，R作为发送方输入$\{(\boldsymbol{A}^1,\boldsymbol{B}^1),(\boldsymbol{A}^2,\boldsymbol{B}^2),\cdots,(\boldsymbol{A}^{\kappa},\boldsymbol{B}^{\kappa})\}$。最终$S$得到一个新的$n\times\kappa$矩阵$\boldsymbol{Q}$，然后对矩阵$\boldsymbol{Q}$的每一行分别异或$F(K,s)$，记为矩阵$\boldsymbol{D}$。最后$S$将矩阵$\boldsymbol{D}$发送给$R$。

4) R将矩阵$\boldsymbol{A}$和矩阵$\boldsymbol{D}$各个对应位置做异或操作，得到的新矩阵记为$\boldsymbol{E}$，则矩阵$\boldsymbol{E}$的第i行是元素y_i对应的函数值，即$F(K,s)\oplus(C(y_i)\wedge s)$。

本文在实现第3节协议的时候，借助了文献[35]的思想，需令功能函数C为线性函数且满足异或同态性，即$C(a\oplus b)=C(a)\oplus C(b)$。

2.4　安全模型

经典的安全模型包括半诚实模型和恶意模型，如下所示。

1) 半诚实模型。敌手各自诚实地执行协议而不偏离指令，但可能试图从协议的输入和输出中学习到额外的信息。

2) 恶意模型。敌手存在多种破坏协议的情况，包括违背协议的执行、恶意篡改数据、提前终止协议等。

本文协议是在半诚实模型下设计的，并采用现实−理想模型进行安全证明，若现实世界中执行的协议π与理想世界执行的功能函数F是不可区分的，则协议是安全的。

定义1　令$\mathrm{view}_{i\in[N]}^{\pi}(X_1,X_2)$为参与方在执行协议$\pi$时的真实视图，$F_{i\in[N]}(X_1,X_2)$是参与方的理想功能函数的输出。如果存在概率多项式时间模拟器Sim，对于所有输入满足$\{\mathrm{Sim}_{i\in[N]}(\lambda,X_1,F(X_1,X_2))\}\approx\{\mathrm{view}_{i\in[N]}^{\pi}(X_1,X_2)\}$，那么存在的协议$\pi$就安全地实现了理想功能函数$F$，该协议的理想函数如图2所示。

参数：N个参与方$P_1,P_2,\cdots,P_N$，每个参与方$P_{i\in[N]}$拥有集合元素$X_i\{x_1^i,x_2^i,\cdots,x_n^i\}$。

协议：

1) $P_{i\in[N]}$输入$X_i\{x_1^i,x_2^i,\cdots,x_n^i\}$。

2) 当$N=2$时，P_2输出$|X_1\cap X_2|$；当$N>2$时，P_N输出$|X_1\cap X_2\cap\cdots\cap X_N|$。

图2　$F_{\text{PSI-CA}}$的理想函数

3　协议的构造

本节利用OKS-PRF和OKVS的对称加密技术提出了两方隐私集合交集基数协议TPSI-CA，然后将其扩展到多方，实现了MPSI-CA协议，并分析了2种协议的正确性和安全性。

3.1　两方隐私集合交集基数协议的构造

两方隐私集合交集基数协议包括2个参与方P_1和P_2，P_1拥有集合$X\{x_1,x_2,\cdots,x_n\}$，P_2拥有集合$Y\{y_1,y_2,\cdots,y_n\}$。该协议的目的是在不泄露额外信息的情况下得到集合X和集合Y的交集大小，即$|X\cap Y|$。TPSI-CA协议的具体构造方式如下。

参数。P_1拥有集合$X\{x_1,x_2,\cdots,x_n\}$和随机种子r，P_2拥有集合$Y\{y_1,y_2,\cdots,y_n\}$、随机种子r和密钥(K,s)。协议用到的安全组件有哈希函数$H:\{0,1\}^{*}\to[m]_{m>n}$、伪随机函数$F:\{0,1\}^{*}\times\{0,1\}^{\kappa}\to\{0,1\}^{\kappa}$、功能函数$C:\{0,1\}^{*}\to\{0,1\}^{\kappa}$和伪随机函数生成器（PRG, pseudorandom function generator）:$\{0,1\}^{*}\times\{0,1\}^{\kappa}\to\{0,1\}^{*}$，其中$\kappa$为安全参数。

1) P_1和P_2都利用随机种子r和伪随机函数生成器PRG产生相同集合$V\{v_1,v_2,\cdots,v_m\}\leftarrow\mathrm{PRG}(r)$，集合$V$作为后续哈希函数$H$映射的目标域。

2) P_1生成大小为m的集合$Q\{q_1,q_2,\cdots,q_m\}$，其中元素均为随机选取。随后对所有的$x_{i\in[n]}\in X\{x_1,x_2,\cdots,x_n\}$，利用哈希函数$H$计算$u_{i\in[n]}=v_{H(x_i)\in[m]}\oplus q_{H(x_i)\in[m]}$，得到集合$U\{u_1,u_2,\cdots,u_n\}$。其中$q_{H(x_i)\in[m]}$和$v_{H(x_i)\in[m]}$分别表示集合$Q\{q_1,q_2,\cdots,q_m\}$和集合$V\{v_1,v_2,\cdots,v_m\}$中下标为

$H(x_i)$的元素值。通过计算集合$U\{u_1,u_2,\cdots,u_n\}$，可将P_1拥有的集合$X\{x_1,x_2,\cdots,x_n\}$映射为集合$V\{v_1,v_2,\cdots,v_m\}$中的元素，并用集合$Q\{q_1,q_2,\cdots,q_m\}$中的$q_{H(x_i)}$对$v_{H(x_i)}$进行掩码，从而隐藏集合$X\{x_1,x_2,\cdots,x_n\}$与集合$V\{v_1,v_2,\cdots,v_m\}$之间的映射关系。

3) P_2利用伪随机函数F和密钥(K,s)计算得到$F(K,s)$，将密钥(K,s)拆分为(k_1,s)和(k_2,s)这2个密钥份额，并满足$F(K,s)=F(k_1,s)\oplus F(k_2,s)$。随后，$P_2$利用密钥份额$(k_1,s)$计算自身元素$y_{i\in[n]}\in Y\{y_1,y_2,\cdots,y_n\}$的函数值$F(k_1,s)\oplus(C(v_{H(y_i)})\wedge s)$，其中$C(v_{H_{(y_i)}})$表示集合$V\{v_1,v_2,\cdots,v_m\}$中元素$v_{H(y_i)\in[m]}$经函数$C$计算的输出值。接着，$P_2$将集合$Y\{y_1,y_2,\cdots,y_n\}$中的元素与其计算的函数值依次进行OKVS编码，即$T\leftarrow \mathrm{Enc}(y_{i\in[n]},F(k_1,s)\oplus(C(v_{H(y_i)})\wedge s))$。最后，$P_2$将$T$发送给$P_1$。

4) P_1以集合$X\{x_1,x_2,\cdots,x_n\}$作为输入，对T进行解码，得到集合$G\{g_1,g_2,\cdots,g_n\}$，其中$g_{i\in[n]}=\mathrm{Dec}(T,x_i)$。如果$x_{i\in[n]}=y_{j\in[n]}$，则$P_1$解码$T$后，可获得$P_2$对该元素用密钥份额$(k_1,s)$计算的函数值，即$g_i=F(k_1,s)\oplus(C(v_{H(y_j)})\wedge s)$，若$x_i\notin Y\{y_1,y_2,\cdots,y_n\}$，则解码结果为随机值。

5) P_1和P_2调用OKS-PRF协议，P_2作为发送方输入(k_2,s)，P_1作为接收方输入$q_{H(x_1)},q_{H(x_2)},\cdots,q_{H(x_n)}$，并得到集合$B\{b_1,b_2,\cdots,b_n\}$，其中$b_{i\in[n]}=F(k_2,s)\oplus(C(q_{H(x_i)})\wedge s)$。

6) P_1利用集合$G\{g_1,g_2,\cdots,g_n\}$和集合$B\{b_1,b_2,\cdots,b_n\}$计算得到集合$W\{w_1,w_2,\cdots,w_n\}$，其中$w_{i\in[n]}=g_i\oplus b_i$。如果$x_i$是集合$Y\{y_1,y_2,\cdots,y_n\}$中的元素，则$w_{i\in[n]}=F(k_1,s)\oplus(C(v_{H(x_i)})\wedge s)\oplus F(k_2,s)\oplus(C(q_{H(x_i)})\wedge s)=F(K,s)\oplus(C(v_{H(x_i)}\oplus q_{H(x_i)})\wedge s)$，其中功能函数$C$满足异或同态性，即$C(a\oplus b)=C(a)\oplus C(b)$。若$x_i\notin Y\{y_1,y_2,\cdots,y_n\}$，则$w_{i\in[n]}$为随机值。随后对集合$U\{u_1,u_2,\cdots,u_n\}$和集合$W\{w_1,w_2,\cdots,w_n\}$应用相同的随机置换$\pi:[n]\rightarrow[n]$得到$u_{\pi(1)},u_{\pi(2)},\cdots,u_{\pi(n)}$和$w_{\pi(1)},w_{\pi(2)},\cdots,w_{\pi(n)}$，并将$u_{\pi(1)}\|w_{\pi(1)},u_{\pi(2)}\|w_{\pi(2)},\cdots,u_{\pi(n)}\|w_{\pi(n)}$发送给$P_2$。随机置换$\pi$是通过打乱数据顺序而不改变数据内容（如$\{x_1,x_2,x_3\}=\{1,2,3\}$经过随机置换$\pi$得到$\{x_{\pi(1)},x_{\pi(2)},x_{\pi(3)}\}=\{3,2,1\}$）来隐藏交集元素的位置信息，且由于$P_1$没有密钥$(K,s)$，故无法通过$w_i$验证$x_i$是否为交集元素。

7) P_2拥有密钥(K,s)，对收到的所有$u_{\pi(i)\in[n]}$，计算$F(K,s)\oplus(C(u_{\pi(i)})\wedge s)$（其中$C(u_i)=C(v_{H(x_i)}\oplus q_{H(x_i)})$），并判断结果是否等于$w_{\pi(i)}$，若相等则交集基数加1。从而$P_2$可利用密钥$(K,s)$验证隐藏在$u_{\pi(i)}$中的元素是否为交集元素，并且因为$P_2$不掌握置换规则，无法确定交集元素的位置信息，所以仅能计算交集集合的大小而无法获取交集元素。

3.1.1 正确性分析

P_1生成集合$V\{v_1,v_2,\cdots,v_m\}$和集合$Q\{q_1,q_2,\cdots,q_m\}$后，对于$x_i\in X\{x_1,x_2,\cdots,x_n\}$，计算得到

$$u_i=q_{H(x_i)}\oplus v_{H(x_i)} \tag{1}$$

P_2生成集合$V\{v_1,v_2,\cdots,v_m\}$和密钥份额(k_1,s)后，对于$y_{i\in[n]}\in Y\{y_1,y_2,\cdots,y_n\}$计算$F(k_1,s)\oplus(C((v_{H(y_i)})\wedge s)$，根据计算结果构造OKVS表$T$，如式(2)所示。

$$T\leftarrow \mathrm{Enc}(y_i,F(k_1,s)\oplus(C(v_{H(y_i)})\wedge s)) \tag{2}$$

并将T发送给P_1，对于$x_i\in X\{x_1,x_2,\cdots,x_n\}$，$P_1$解码$T$表得到

$$g_i=\mathrm{Dec}(T,x_i) \tag{3}$$

当P_1和P_2执行OKS-PRF协议时，P_2输入密钥份额(k_2,s)，P_1输入$q_{H(x_1)},q_{H(x_2)},\cdots,q_{H(x_n)}$并得到

$$b_i=F(k_2,s)\oplus(C(q_{H(x_i)})\wedge s) \tag{4}$$

然后，P_1通过计算得到

$$w_i=g_i\oplus b_i \tag{5}$$

当元素$x_i\in(X\cap Y)$时，式(3)的计算结果为

$$g_i=F(k_1,s)\oplus(C(v_{H(x_i)})\wedge s) \tag{6}$$

将式(4)和式(6)代入式(5)，计算结果为

$$\begin{aligned}w_i&=F(k_1,s)\oplus(C(v_{H(x_i)})\wedge s)\oplus F(k_2,s)\oplus(C(q_{H(x_i)})\wedge s)\\&=(F(k_1,s)\oplus F(k_2,s))\oplus((C(v_{H(x_i)})\oplus C(q_{H(x_i)}))\wedge s)\\&=F(K,s)\oplus(C(v_{H(x_i)}\oplus q_{H(x_i)})\wedge s)\\&=F(K,s)\oplus(C(u_i)\wedge s)\end{aligned} \tag{7}$$

P_1将随机置换后的$w_{\pi(i)}\|u_{\pi(i)}$发给P_2后，P_2利用密钥(K,s)和$u_{\pi(i)}$可计算得到

$$w_i' = F(K,s)\oplus(C(u_{\pi(i)})\wedge s) \tag{8}$$

当$w_i' = w_{\pi(i)}$时，说明是交集元素，交集基数加1。若$x_i \notin (X\cap Y)$，则P_1在式(3)和式(5)中计算的g_i和w_i均是随机值，且$w_i' \neq w_{\pi(i)}$，不是交集元素。

最终，P_2通过查询满足$F(K,s)\oplus(C(u_{\pi(i)})\wedge s) = w_{\pi(i)}$的数量，确定交集基数的大小。

3.1.2　安全性分析

通过对P_1和P_2的视图进行模拟来展示协议的安全性。如果这些模拟视图与真实视图无法区分，那么协议是安全的。模拟P_1和P_2的本地数据和接收到的数据的过程如下。

1) 模拟P_1。集合$Q\{q_1,q_2,\cdots,q_m\}$是由P_1本地产生的，模拟器可用m个随机值替代它们。集合$V\{v_1,v_2,\cdots,v_m\}$是通过随机种子r产生的，模拟器可生成一个随机种子替代r。P_1利用输入集合$X\{x_1,x_2,\cdots,x_n\}$、集合Q和集合V，可计算得到集合$U\{u_1,u_2,\cdots,u_n\}$，将集合U发给模拟器，其中$u_{i\in[n]} = q_{H(x_i)}\oplus v_{H(x_i)}$。至此，$P_1$本地产生的数据模拟完成，随后对$P_1$接收到的数据进行模拟。由于$P_1$不知道密钥份额$(k_1,s)$，因此对来自$P_2$的$T\leftarrow \mathrm{Enc}(y_i,F(k_1,s)\oplus(C(v_{H(y_i)})\wedge s))$是无法区分的，模拟器可用$n$个随机键值对模拟收到的表$T$。之后，$P_1$对所有的$x_i\in X$解码$T$，得到集合$G\{g_1,g_2,\cdots,g_n\}$，其中$g_i = \mathrm{Dec}(T,x_i)$。当$P_1$和$P_2$执行OKS-PRF协议时，$P_1$作为接收方输入$\{q_{H(x_1)},q_{H(x_2)},\cdots,q_{H(x_n)}\}$，并得到集合$B\{b_1,b_2,\cdots,b_n\}$，其中$b_i = F(k_2,s)\oplus(C(q_{H(x_i)})\wedge s)$。由于不知道密钥份额$(k_2,s)$，因此得到的集合$B\{b_1,b_2,\cdots,b_n\}$对$P_1$来说是全是随机的，模拟器可用$n$个随机值来模拟得到的集合$B$。随后，$P_1$利用集合$G$和集合$B$，可计算得到集合$W\{w_1,w_2,\cdots,w_n\}$，其中$w_i = g_i\oplus b_i$。最后，$P_1$利用随机置换$\pi:[n]\rightarrow[n]$对集合$W$和集合$U$打乱顺序，并将得到的$u_{\pi(1)}\|w_{\pi(1)},u_{\pi(2)}\|w_{\pi(2)},\cdots,u_{\pi(n)}\|w_{\pi(n)}$发送给模拟器。

2) 模拟P_2。P_2拥有随机种子r和密钥(K,s)，可计算得到函数值$F(K,s)$、$F(k_1,s)$、$F(k_2,s)$和集合$V\{v_1,v_2,\cdots,v_m\}\leftarrow \mathrm{PRG}(r)$，模拟器可用随机值来模拟它们。对于所有的元素$y_{i\in[n]}\in Y$，$P_2$可计算得到$F(k_1,s)\oplus(C(v_{H(y_i)})\wedge s)$，并将其编码为OKVS表$T\leftarrow \mathrm{Enc}(y_i,F(k_1,s)\oplus(C(v_{H(y_i)})\wedge s))$发送给模拟器。至此，$P_2$本地产生的数据模拟完成，随后对$P_2$接收到的数据进行模拟。当得到$P_1$发来的数据$u_{\pi(1)}\|w_{\pi(1)},u_{\pi(2)}\|w_{\pi(2)},\cdots,u_{\pi(n)}\|w_{\pi(n)}$后，由于不知道$P_1$用来掩码的集合$Q\{q_1,q_2,\cdots,q_m\}$和随机置换$\pi$，则$u_{\pi(1)}\|w_{\pi(1)},u_{\pi(2)}\|w_{\pi(2)},\cdots,u_{\pi(n)}\|w_{\pi(n)}$对$P_2$来说完全是随机的，模拟器可选择$n$组随机值来模拟。最后，对于所有$u_{\pi(i)\in[n]}$，$P_2$利用密钥$(K,s)$计算其对应的函数值$F(K,s)\oplus(C(u_{\pi(i)\in[n]})\wedge s)$是否等于$w_{\pi(i)}$，并将该计算结果发给模拟器。

整体上，P_1和P_2相互接收的数据对彼此来说均是随机的，模拟器可以完美地模拟这2个参与方的视图，并无法与真实视图区分。

3.2　多方隐私集合交集基数协议的构造

多方隐私集合交集协议可以更广泛地适用于各种联合分析和数据共享的场景。为了提升TPSI-CA协议的实用性，将其进一步扩展成为多方PSI-CA协议。MPSI-CA协议包含N个参与方$P_1,P_2,\cdots,P_N$，每个参与方$P_{i\in[N]}$拥有集合$X_i\{x_1^i,x_2^i,\cdots,x_n^i\}$。该协议的目的是求得所有参与方的交集大小，即$\left|\bigcap_{i=1}^{i=N} X_i\right|$。MPSI-CA协议的具体构造方式如下。

参数。N个参与方$P_1,P_2,\cdots,P_N$，每个参与方$P_{i\in[N]}$拥有集合$X_i\{x_1^i,x_2^i,\cdots,x_n^i\}$。伪随机函数$F$:$\{0,1\}^*\times\{0,1\}^\kappa\rightarrow\{0,1\}^\kappa$，其中$\kappa$为安全参数。

1) P_1生成$N-2$个随机种子$s_{i\in[2,N-1]}$，并将s_i发送给$P_{i\in[2,N-1]}$。

2) P_N生成$N-1$个随机种子$r_{i\in[2,N]}$，并将r_i发送给$P_{i\in[2,N]}$。

3) 每个参与方$P_{i\in[2,N-1]}$收到随机种子s_i和r_i后，对于所有的$x_{j\in[n]}^i\in X_i\{x_1^i,x_2^i,\cdots,x_n^i\}$，通过伪随机函数$F$计算$F(s_i,x_{j\in[n]}^i)\oplus F(r_i,x_{j\in[n]}^i)$，并将$x_{j\in[n]}^i$与其对应的函数值执行OKVS编码，可得到表$T_i\leftarrow \mathrm{Enc}(x_j^i,F(s_i,x_{j\in[n]}^i)\oplus F(r_i,x_{j\in[n]}^i))$。随后，

$N-2$个参与方$P_{i\in[2,N-1]}$均把$T_{i\in[2,N-1]}$发送给P_1。

4) P_N拥有随机种子$r_{i\in[2,N]}$，对于所有$x^N_{j\in[n]}\in X_N\{x^N_1,x^N_2,\cdots,x^N_n\}$，计算$F(r_2,x^N_{j\in[n]})\oplus F(r_3,x^N_{j\in[n]})\oplus\cdots\oplus F(r_N,x^N_{j\in[n]})$，并编码为OKVS表$T_N\leftarrow\mathrm{Enc}(x^N_{j\in[n]},\ F(r_2,x^N_{j\in[n]})\oplus F(r_3,x^N_{j\in[n]})\oplus\cdots\oplus F(r_N,x^N_{j\in[n]}))$后发送给$P_1$。

5) P_1收到$T_{i\in[2,N]}$表后，对于所有的$x^1_{j\in[n]}\in X_1\{x^1_1,x^1_2,\cdots,x^1_n\}$，分别解码$N-1$个$T_{i\in[2,N]}$表，可得到集合$\theta\{v_1,v_2,\cdots,v_n\}$，其中$v_{j\in[n]}=\mathrm{Dec}(T_2,x^1_j)\oplus\mathrm{Dec}(T_3,x^1_j)\oplus\cdots\oplus\mathrm{Dec}(T_N,x^1_j)$。再根据随机种子$s_{i\in[2,N-1]}$和集合$\theta$，计算得到集合$\theta'\{\rho_1,\rho_2,\cdots,\rho_n\}$，其中$\rho_{j\in[n]}=v_j\oplus F(s_2,x^1_j)\oplus\cdots\oplus F(s_{N-1},x^1_j)$。当元素$x^1_{j\in[n]}$为交集元素时，即$x^1_{j\in[n]}\in\bigcap_{i=1}^{i=N}X_i$，则$v_{j\in[n]}=\bigoplus_{i=2}^{i=N-1}F(s_i,x^1_j)\oplus F(r_N,x^1_j)$，$\rho_{j\in[n]}=F(r_N,x^1_j)$；若$x^1_{j\in[n]}\notin\bigcap_{i=1}^{i=N}X_i$，则$v_{j\in[n]}$和$\rho_{j\in[n]}$皆为随机值。

6) P_1和P_N调用TPSI-CA协议，P_1作为发送方，输入集合元素$\theta'\{\rho_1,\rho_2,\cdots,\rho_n\}$，$P_N$作为接收方，输入集合元素$\{F(r_N,x^N_1),F(r_N,x^N_2),\cdots,F(r_N,x^N_n)\}$，最终$P_N$输出交集的大小。

3.2.1 正确性分析

当$P_{i\in[2,N-1]}$收到P_1和P_N发来的随机种子s_i和r_i后，可编码OKVS表$T_i\leftarrow\mathrm{Enc}(x^i_j,F(s_i,x^i_{j\in[n]})\oplus F(r_i,x^i_{j\in[n]}))$发给$P_1$。同时，$P_N$编码OKVS表$T_N\leftarrow\mathrm{Enc}(x^N_{j\in[n]},\ F(r_2,x^N_{j\in[n]})\oplus F(r_3,x^N_{j\in[n]})\oplus\cdots\oplus F(r_N,x^N_{j\in[n]}))$也发送给$P_1$。

为便于理解，取$N=3$。对所有$x^1_{j\in[n]}\in X_1$，P_1解码收到的2个T_i表，计算得到集合$\theta\{v_1,v_2,\cdots,v_n\}$，其中$v_j=\mathrm{Dec}(T_2,x^1_j)\oplus\mathrm{Dec}(T_3,x^1_j)$。再计算得到集合$\theta'\{\rho_1,\rho_2,\cdots,\rho_n\}$，其中

$$\rho_j=v_j\oplus F(s_2,x^1_j)\tag{9}$$

当$x_j\in(X_1\cap X_2\cap X_3)$时，对应的

$$v_j=\mathrm{Dec}(T_2,x_j)\oplus\mathrm{Dec}(T_3,x_j)=F(s_2,x_j)\oplus F(r_3,x_j)\tag{10}$$

将式(10)的结果代入式(9)后，可得

$$\rho_j=F(s_2,x_j)\oplus F(r_3,x_j)\oplus F(s_2,x_j)=F(r_3,x_j)\tag{11}$$

则$\rho_j=F(r_3,x_j)\in\{F(r_3,x^3_1),\cdots,F(r_3,x^3_n)\}$属于交集元素。当$x_j\notin(X_1\cap X_2\cap X_3)$时，$v_j$就会在式(9)中得到随机值。

最终，P_1拥有集合$\{\rho_1,\rho_2,\cdots,\rho_n\}$，$P_3$拥有集合$\{F(r_3,x^3_1),F(r_3,x^3_2),\cdots,F(r_3,x^3_n)\}$，通过调用TPSI-CA协议后可得到交集基数。

3.2.2 安全性分析

由于参与方$P_2,\cdots,P_{N-1}$仅对自己的集合进行了OKVS编码，而没有获取到其他参与方的元素信息，因此仅对P_1和P_2的视图进行模拟，来说明协议的安全性。如果这些模拟视图与真实视图无法区分，则可证明协议是安全的。模拟P_1和P_2的本地数据和接收到的数据的过程如下。

1) 模拟P_1。P_1产生的$N-2$个随机种子$s_{i\in[2,N-1]}$，模拟器可用$N-2$个随机值替代它们。对所有$x^1_{j\in[n]}\in X_1$的元素，P_1可计算得到$F(s_2,x^1_j)\oplus F(s_3,x^1_j)\oplus\cdots\oplus F(s_{N-1},x^1_j)$，并将这些函数值的结果发送给模拟器。至此，$P_1$本地产生的数据模拟完成，随后对$P_1$接收到的数据进行模拟。当收到$N-1$个OKVS表$T_i$时（在$P_1,P_2,\cdots,P_{N-1}$进行合谋时，$P_1$可得知$T_{i\in[2,N-1]}$表中键值对的数据，但$P_1$不与$P_N$合谋，则$P_N$的$T_N$表中的键值数据对$P_1$来说均是随机的，交集元素仍然被隐藏），模拟器可用$n(N-1)$个随机键值对模拟这$N-1$个$T_i$表。$P_1$利用所有$x^1_{j\in[n]}\in X_1$的元素对收到的这$N-1$个$T_i$表进行解码，可计算得到集合$\theta'\{\rho_1,\rho_2,\cdots,\rho_n\}$，其中$\rho_j=\mathrm{Dec}(T_2,x^1_j)\oplus\cdots\oplus\mathrm{Dec}(T_N,x^1_j)\oplus F(s_2,x^1_j)\oplus\cdots\oplus F(s_{N-1},x^1_j)$，并将集合$\theta'$发给模拟器。随后$P_1$和$P_N$执行TPSI-CA协议的安全性分析在3.1.2节可见。

2) 模拟P_N。P_N生成$N-1$个随机种子$r_{i\in[2,N]}$，模拟器可用$N-1$个随机值替代它们。对所有$x^N_{j\in[n]}\in X_N$的元素，P_N计算得到函数值$F(r_2,x^N_j)\oplus F(r_3,x^N_j)\oplus\cdots\oplus F(r_N,x^N_j)$，并将其编码为OKVS表$T_N\leftarrow\mathrm{Enc}(x^N_j,F(r_2,x^N_j)\oplus F(r_3,x^N_j)\oplus(r_N,x^N_j))$后发给模拟器。至此，$P_N$本地产生的数据模拟完成。随后$P_N$和$P_1$执行TPSI-CA协议的安全性分析在3.1.2节可见。

整体上，P_1和P_N相互接收的数据对彼此来说是随机的，模拟器可以完美地模拟这2个参与方的视图，并无法与真实视图区分。

4　性能测试与分析

4.1　实验环境与性能分析

实验使用C++语言来编写代码，并在Ubuntu 20.04 LTS和AMD Ryzen 7 7735HS 3.20 GHz处理器、16 GB RAM的Linux操作系统上执行。本文协议的OKVS方案根据文献[22]进行实例化，OKS-PRF方案是导入OpenSSL和LibOTe库来实现的，安全计算参数$\kappa = 128$，安全统计参数$\lambda = 40$。表2~表4分别是将本文所提TPSI-CA和MPSI-CA协议与目前最先进的PSI-CA协议在计算和通信复杂度方面的对比。其中，n表示集合元素的个数，κ表示安全计算参数，λ表示安全统计参数，γ表示哈希函数个数，m表示借助云服务器的个数，$|T_n|$表示将n组键值对编码为OKVS表的通信开销，$|\text{Odk-PRF}_n|$表示2个参与方对n个数据执行Odk-PRF协议的通信开销，$|\text{pack}_n|$表示将n组键值对编码成点集包的通信开销。

4.2　测试比较

本节将本文方案与目前众多PSI-CA协议在不同集合大小下分别进行测试比较，具体比较结果如表5~表9所示，时间单位为s，其中“—”表示运行时间过长（超过1 000 s）。

表5是TPSI-CA与无云辅助的两方PSI-CA协议时间对比。目前最快的无云辅助两方PSI-CA[32]协议依然是基于密钥交换OPRF[29]的思想来完成的。本文提出的TPSI-CA协议不需要对明文数据执行大量的幂运算，只需要执行开销低的异或和与操作。因此，在集合规模为百万级时，TPSI-CA协议的执行速度是文献[32]的1.8倍，并且显著快于文献[31]。

表6是TPSI-CA与单云辅助的两方PSI-CA协议时间对比。由于文献[36]引入了一个第三方，2个客户端可利用第三方快速完成洗牌的操作，因此性能较高。但引入的这个第三方要求是半诚实的且不与任何一个客户端进行合谋，具有较强的假设，而TPSI-CA协议不需要引入可信的第三方，具有较高的实用性。

表2　TPSI-CA协议与目前最先进的PSI-CA协议的计算和通信复杂度对比

协议	计算复杂度		通信复杂度	
	发送方	接收方	发送方	接收方
TPSI-CA	$O((\lambda+\kappa)n)$	$O((\lambda+\kappa)n)$	$(2\kappa+2\lambda)n+\|T_n\|$	$(2\kappa+2\lambda)n+\|T_n\|$
文献[32]	$O(n)$	$O(n)$	$3n\kappa$	$3n\kappa$
文献[35]	$O((m\kappa+\gamma+\lambda)n)$	$O((\gamma+m)n)$	κ+1.5m|Odk-PRF$_n$|+1.5|pack$_n$|	$(3n+m-1)\kappa$
文献[36]	$O(n)$	$O(n)$	$(n+2)\kappa$	$3n\kappa$

表3　MPSI-CA协议与目前最先进的PSI-CA协议的计算复杂度对比

协议	P_1	P_2	$P_{i\in[3,N-1]}$	P_N
MPSI-CA	$O((\lambda N+N+\kappa)n)$	$O(\lambda n)$	$O(\lambda n)$	$O((\lambda+N+\kappa)n)$
文献[40]	$O(n)$	$O(n)$	$O(n)$	$O(n)$
文献[36]	$O((N-2)n)$	$O((\lambda+2)n)$	$O((\lambda+1)n)$	$O(n+\kappa)$

表4　MPSI-CA协议与目前最先进的PSI-CA协议的通信复杂度对比

协议	P_1	P_2	$P_{i\in[3,N-1]}$	P_N
MPSI-CA	$(n+N)\kappa+N\|T_n\|+2\lambda n$	$2\kappa+\|T_n\|$	$2\kappa+\|T_n\|$	$(2n+N-1)\kappa+2\|T_n\|+2\lambda n$
文献[40]	$12.3n\kappa+n$	$10.3n\kappa$	$10.3n\kappa$	$10.3n\kappa$
文献[36]	$(N+2Nn-2n)\kappa$	$(N+2n+2)\kappa+\|T_n\|$	$3\kappa+\|T_n\|$	$(2\kappa+2\kappa n+\|T_n\|)(N-2)+(2+n)\kappa$

表5　TPSI-CA与无云辅助的两方PSI-CA协议时间对比

协议	运行时间/s		
	集合大小为2^{16}	集合大小为2^{18}	集合大小为2^{20}
TPSI-CA	2.19	9.06	36.61
文献[31]	22.14	82.27	315.12
文献[32]	4.37	17.34	66.52

表6　TPSI-CA与单云辅助的两方PSI-CA协议时间对比

协议	运行时间/s		
	集合大小为2^{16}	集合大小为2^{18}	集合大小为2^{20}
TPSI-CA	2.19	9.06	36.61
文献[36]	0.12	0.48	2.07

表7是TPSI-CA与多云辅助（2个及2个以上的云服务器）的两方PSI-CA协议时间对比。TPSI-CA协议在调用OKS-PRF时，可令发送方提前选择密钥来计算元素的函数值。因此，TPSI-CA的执行时间比文献[35]短，并与文献[34]的执行时间差别不大。

表7　TPSI-CA与多云辅助（2个及2个以上的云服务器）的两方PSI-CA协议时间对比

协议	运行时间/s		
	集合大小为2^{16}	集合大小为2^{18}	集合大小为2^{20}
TPSI-CA	2.19	9.06	36.61
文献[34]	5.87	8.12	12.47
文献[35]	7.34	33.54	142.89

表8和表9分别是MPSI-CA在参与方个数为4和8时与多方PSI-CA协议的时间对比。MPSI-CA协议虽然比文献[36]执行时间长，但却可以抵抗$N-2$个参与方合谋，并且整体上的时间也可以被接受。与文献[40]相比，MPSI-CA协议在执行时间上具有较大的优势。相较于当前最先进的方案，MPSI-CA协议在效率和安全性方面均具有较好的可应用性。

表8　MPSI-CA与多方PSI-CA协议时间对比（$N=4$）

协议	运行时间/s			抗合谋/个
	集合大小为2^{16}	集合大小为2^{18}	集合大小为2^{20}	
MPSI-CA	4.62	18.57	79.35	$N-2$
文献[36]	2.38	9.61	35.25	$N-3$
文献[40]	161.04	646.36	—	$N-1$

表9　MPSI-CA与多方PSI-CA协议时间对比（$N=8$）

协议	运行时间/s			抗合谋/个
	集合大小为2^{16}	集合大小为2^{18}	集合大小为2^{20}	
MPSI-CA	6.42	26.77	115.32	$N-2$
文献[36]	3.53	14.63	51.86	$N-3$
文献[40]	212.57	885.12	—	$N-1$

5　结束语

在传统半诚实模型下，两方PSI-CA协议通常采用同态加密技术或基于密钥交换的OPRF技术，虽然这些方法在协议的实现上较为直观，但存在计算开销大的问题。针对这一局限性，本文提出了一种基于不经意键值存储和不经意密钥共享伪随机函数的高效TPSI-CA协议。该协议不仅在执行效率上是当前最快两方PSI-CA协议的1.8倍，还能扩展到MPSI-CA场景并抵抗$N-2$个参与方的合谋攻击，同时不需要依赖可信的第三方，具有显著的实用性和可扩展性优势。然而，本文方案目前基于半诚实安全模型设计，尚无法抵抗恶意敌手攻击。尽管采用零知识证明等密码学技术可以构建具备恶意安全的PSI-CA协议，但由此带来的计算开销会显著增加，这会严重制约方案的实用价值。因此，在保证计算效率的同时，设计出具备高效且恶意安全的PSI-CA协议是下一步工作的研究重点。

参考文献：

[1] WU M L, YUEN T H. Efficient unbalanced private set intersection cardinality and user-friendly privacy-preserving contact tracing[C]//Proceedings of the 32nd USENIX Conference on Security Symposium. Berkeley: USENIX Association, 2023: 283-300.

[2] CHEN Y F, WU A X, YANG Y E, et al. Efficient verifiable cloud-assisted PSI cardinality for privacy-preserving contact tracing[J]. IEEE Transactions on Cloud Computing, 2024, 12(1): 251-263.

[3] YANG Q H, YANG Y E, XU S Y, et al. PPCT: privacy-preserving contact tracing using concise private set intersection cardinality[J]. Journal of Network and Systems Management, 2024, 32(4): 97.

[4] BECK M, KERSCHBAUM F. Approximate two-party privacy-preserving string matching with linear complexity[C]//Proceedings of the 2013 IEEE International Congress on Big Data. Piscataway: IEEE Press, 2013: 31-37.

[5] KAMP T V D, STRITZL D, JONKER W, et al. Two-client and multi-client functional encryption for set intersection[C]//Information Security

and Privacy. Berlin: Springer, 2019: 97-115.

[6] 李凤华, 李晖, 贾焰, 等. 隐私计算研究范畴及发展趋势[J]. 通信学报, 2016, 37(4): 1-11.
LI F H, LI H, JIA Y, et al. Privacy computing: concept, connotation and its research trend[J]. Journal on Communications, 2016, 37(4): 1-11.

[7] FREEDMAN M J, NISSIM K, PINKAS B. Efficient private matching and set intersection[C]//Advances in Cryptology-EUROCRYPT 2004. Berlin: Springer, 2004: 1-19.

[8] KISSNER L, SONG D. Privacy-preserving set operations[C]//Advances in Cryptology-CRYPTO 2005. Berlin: Springer, 2005: 241-257.

[9] HAZAY C, LINDELL Y. Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries[C]// Theory of Cryptography. Berlin: Springer, 2008: 155-175.

[10] HAZAY C. Oblivious polynomial evaluation and secure set-intersection from algebraic PRFs[J]. Journal of Cryptology, 2018, 31(2): 537-586.

[11] DE CRISTOFARO E, KIM J, TSUDIK G. Linear-complexity private set intersection protocols secure in malicious model[C]//Advances in Cryptology-ASIACRYPT 2010. Berlin: Springer, 2010: 213-231.

[12] DE CRISTOFARO E, TSUDIK G. Practical private set intersection protocols with linear complexity[C]//Financial Cryptography and Data Security. Berlin: Springer, 2010: 143-159.

[13] DE CRISTOFARO E, TSUDIK G. Experimenting with fast private set intersection[C]//Trust and Trustworthy Computing. Berlin: Springer, 2012: 55-73.

[14] KERSCHBAUM F. Outsourced private set intersection using homomorphic encryption[C]//Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security. New York: ACM Press, 2012: 85-86.

[15] DONG C Y, CHEN L Q, WEN Z K. When private set intersection meets big data: an efficient and scalable protocol[C]//Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security. New York: ACM Press, 2013: 789-800.

[16] DEBNATH S K, DUTTA R. Efficient private set intersection cardinality in the presence of malicious adversaries[C]//Provable Security. Berlin: Springer, 2015: 326-339.

[17] DEBNATH S K, DUTTA R. Secure and efficient private set intersection cardinality using bloom filter[C]//Information Security. Berlin: Springer, 2015: 209-226.

[18] KOLESNIKOV V, KUMARESAN R, ROSULEK M, et al. Efficient batched oblivious PRF with applications to private set intersection[C]// Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2016: 818-829.

[19] NEVO O, TRIEU N, YANAI A. Simple, fast malicious multiparty private set intersection[C]//Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2021: 1151-1165.

[20] GARIMELLA G, PINKAS B, ROSULEK M, et al. Oblivious key-value stores and amplification for private set intersection[C]//Advances in Cryptology-CRYPTO 2021. Berlin: Springer, 2021: 395-425.

[21] PINKAS B, ROSULEK M, TRIEU N, et al. SpOT-light: lightweight private set intersection from sparse OT extension[C]// Advances in Cryptology-CRYPTO 2019. Berlin: Springer, 2019: 401-431.

[22] RAGHURAMAN S, RINDAL P. Blazing fast PSI from improved OKVS and subfield VOLE[C]//Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2022: 2505-2517.

[23] KOLESNIKOV V, MATANIA N, PINKAS B, et al. Practical multi-party private set intersection from symmetric-key techniques[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2017: 1257-1272.

[24] WEI L F, LIU J H, ZHANG L, et al. Efficient multi-party private set intersection protocols for large participants and small sets[J]. Computer Standards & Interfaces, 2024, 87: 103764.

[25] YANG Y H, YANG Y B, CHEN X, et al. DMPSI: efficient scalable delegated multiparty PSI and PSI-CA with oblivious PRF[J]. IEEE Transactions on Services Computing, 2024, 17(2): 497-508.

[26] AGRAWAL R, EVFIMIEVSKI A, SRIKANT R. Information sharing across private databases[C]//Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2003: 86-97.

[27] HOHENBERGER S, WEIS S A. Honest-verifier private disjointness testing without random oracles[C]//Privacy Enhancing Technologies. Berlin: Springer, 2006: 277-294.

[28] CAMENISCH J, ZAVERUCHA G M. Private intersection of certified sets[C]//Financial Cryptography and Data Security. Berlin: Springer, 2009: 108-127.

[29] CRISTOFARO E D, GASTI P, TSUDIK G. Fast and private computation of cardinality of set intersection and union[C]//Cryptology and Network Security. Berlin: Springer, 2012: 218-231.

[30] DEBNATH S K, DUTTA R. Provably secure fair mutual private set intersection cardinality utilizing bloom filter[C]//Information Security and Cryptology. Berlin: Springer, 2017: 505-525.

[31] GARIMELLA G, MOHASSEL P, ROSULEK M, et al. Private set operations from oblivious switching[C]//Public-Key Cryptography-PKC 2021. Berlin: Springer, 2021: 591-617.

[32] CHEN Y, ZHANG M, ZHANG C, et al. Private set operations from multi-query reverse private membership test[C]//Public-Key Cryptography-PKC 2024. Berlin: Springer, 2024: 387-416.

[33] TAJIMA A, SATO H, YAMANA H. Outsourced private set intersection cardinality with fully homomorphic encryption[C]//Proceedings of the 2018 6th International Conference on Multimedia Computing and

Systems (ICMCS). Piscataway: IEEE Press, 2018: 1-8.

[34] YANG X Y, ZHAO Y Q, ZHOU S F, et al. A lightweight delegated private set intersection cardinality protocol[J]. Computer Standards & Interfaces, 2024, 87: 103760.

[35] DUONG T, PHAN D H, TRIEU N. Catalic: delegated PSI cardinality with applications to contact tracing[C]//Advances in Cryptology-ASIACRYPT 2020. Berlin: Springer, 2020: 870-899.

[36] GAO J H, TRIEU N, YANAI A. Multiparty private set intersection cardinality and its applications[J]. Proceedings on Privacy Enhancing Technologies, 2024(2): 73-90.

[37] DEBNATH S K, STANICA P, KUNDU N, et al. Secure and efficient multiparty private set intersection cardinality[J]. Advances in Mathematics of Communications, 2021, 15(2): 365-386.

[38] JOLFAEI A A, MALA H, ZAREZADEH M. EO-PSI-CA: efficient outsourced private set intersection cardinality[J]. Journal of Information Security and Applications, 2022, 65: 102996.

[39] WU A X, XIN X J, ZHU J H, et al. Cloud-assisted laconic private set intersection cardinality[J]. IEEE Transactions on Cloud Computing, 2024, 12(1): 295-305.

[40] YANG Y B, DONG X L, CAO Z F, et al. EMPSI: efficient multiparty private set intersection (with cardinality)[J]. Frontiers of Computer Science, 2024, 18(1): 199-213.

[作者简介]

李功丽（1981– ），女，河南信阳人，博士，河南师范大学副教授、硕士生导师，主要研究方向为信息安全、密码协议设计、隐私保护。

刘威辰（2000– ），男，河南商丘人，河南师范大学硕士生，主要研究方向为密码协议。

郑东（1964– ），男，山西临汾人，博士，西安邮电大学教授、博士生导师，主要研究方向为密码学理论与云安全。