软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2023,34(11):5442−5456 [doi: 10.13328/j.cnki.jos.006747]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

双云辅助的超阈值多方隐私集合交集计算协议*

魏立斐[1], 刘纪海[1], 张 蕾[1], 宁建廷[2,3]

[1](上海海洋大学 信息学院, 上海 201306)

[2](福建师范大学 数学与信息学院, 福建 福州 350117)

[3](信息安全国家重点实验室 (中国科学院 信息工程研究所), 北京 100093)

通信作者: 张蕾, E-mail: Lzhang@shou.edu.cn

摘 要: 超阈值多方隐私集合求交协议 (OT-MP-PSI) 是 PSI 协议的变体, 允许 m 个参与方共同计算至少 t ($t\leqslant m$) 个参与方中拥有相同元素的超阈值交集, 且保证仅拥有超阈值元素的参与方才能知晓该元素是否属于超阈值交集, 对于其他信息一无所知. OT-MP-PSI 推广了 PSI 的实际应用场景. 现有方案均基于昂贵的公钥密码来构建, 其较大的计算量导致运行时间缓慢. 首先设计一个基于对称密码的不经意可编程伪随机秘密共享 (OPPR-SS) 密码组件, 并基于 OPPR-SS 组件设计双云辅助的 OT-MP-PSI 协议, 将秘密分发和重构的任务分别交给不可信云服务器来辅助完成, 实现弱计算能力的参与方也能完成 OT-MP-PSI 协议. 在半诚实模型下证明协议安全性. 相比现有的 OT-MP-PSI 协议, 所提协议在秘密分发和重构阶段均具有最优运行时间和通信负载, 参与方、共享方和重构方的通信复杂度不再与阈值 t 有关, 实现参与方常数轮的通信, 通信复杂度仅为 $O(n)$, 秘密分发方和重构方的计算复杂度仅与对称密码次数有关.

关键词: 隐私集合交集; 不经意传输; 秘密共享; 超阈值; 云辅助

中图法分类号: TP309

中文引用格式: 魏立斐, 刘纪海, 张蕾, 宁建廷. 双云辅助的超阈值多方隐私集合交集计算协议. 软件学报, 2023, 34(11): 5442–5456. http://www.jos.org.cn/1000-9825/6747.htm

英文引用格式: Wei LF, Liu JH, Zhang L, Ning JT. Two Cloud-assisted Over-threshold Multi-party Private Set Intersection Calculation Protocol. Ruan Jian Xue Bao/Journal of Software, 2023, 34(11): 5442–5456 (in Chinese). http://www.jos.org.cn/1000-9825/6747.htm

Two Cloud-assisted Over-threshold Multi-party Private Set Intersection Calculation Protocol

WEI Li-Fei[1], LIU Ji-Hai[1], ZHANG Lei[1], NING Jian-Ting[2,3]

[1](College of Information Technology, Shanghai Ocean University, Shanghai 201306)

[2](College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350117, China)

[3](State Key Laboratory of Information Security (Institute of Information Engineering, Chinese Academy of Sciences), Beijing 100093, China)

Abstract: The over-threshold multi-party private set intersection (OT-MP-PSI) protocol is a variant of the conventional PSI protocol. This protocol allows m participants to jointly compute the OT intersection for which at least t ($t\leqslant m$) participants have the common element and ensures that only the participant with the OT element knows whether the element belongs to the OT intersection and nothing else. The OT-MP-PSI protocol extends the practical application scenarios of the PSI protocol. As the existing schemes are all constructed on the basis of expensive public key-based cryptography, their heavy computational burden results in long runtime. This study designs a novel cryptographic component, the oblivious programmable pseudo-random secret-sharing (OPPR-SS) component based on symmetric cryptography. Furthermore, a two cloud-assisted OT-MP-PSI protocol is designed on the basis of the OPPR-SS component, and it assigns

* 基金项目: 国家自然科学基金 (61972241, 61972094); 上海市自然科学基金 (22ZR1427100, 18ZR1417300); 上海海洋大学骆肇尧大学生科技创新基金; 福建省科协第二届青年人才托举工程
收稿时间: 2022-03-16; 修改时间: 2022-06-02; 采用时间: 2022-07-17; jos 在线出版时间: 2023-06-16
CNKI 网络首发时间: 2023-06-19

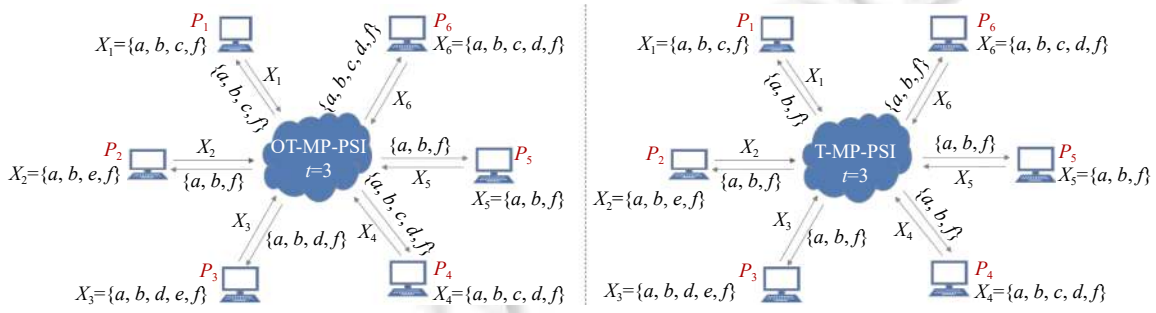
the tasks of secret sharing and reconstructing to untrusted cloud servers, respectively, so that they can assist in the completion of those tasks. As a result, participants with weak computation capability can complete the OT-MP-PSI protocol as well. Furthermore, the study proves that the proposed protocol is secure in the semi-honest model. Compared with the existing OT-MP-PSI protocols, the proposed protocol achieves the optimal runtime and communication overhead at both the secret sharing stage and the secret reconstructing stage. The communication complexities of the participants, the secret sharing cloud, and reconstructing cloud are no longer related to the threshold t. The number of communication rounds for the participants is constant, and the communication complexity is merely O(n). The computational complexities of the secret sharing cloud and the secret reconstructing cloud are only related to the number of symmetric cryptographic operations.

Key words: private set intersection; oblivious transfer; secret sharing; over-threshold; cloud-assisted

隐私集合求交 (private set intersection, PSI) 协议允许参与方 P_1、P_2 分别输入私有集合 X_1、X_2, 输出集合交集 $X_1 \cap X_2$ 且不透露除交集外的任何信息. PSI 根据双方需求可实现一方输出集合交集[1−4]、两方输出集合交集[5]、两方输出秘密共享集合交集[6,7]等功能. 目前基于不经意传输 (oblivious transfer, OT) 的不经意伪随机函数 (oblivious pseudo-random function, OPRF) 构建的 PSI 协议, 在计算和通信均衡中表现优秀[8]. PSI 作为隐私计算的专有协议, 已实际部署诸多应用场景, 如隐私联系人查找[9]、在线广告曝光效率[10]、新冠接触者追踪[11]等.

多方隐私集合求交 (multi-party PSI, MP-PSI) 协议推广了传统 PSI 的参与方数量, 允许 m 个参与方 $P_1, P_2, \ldots, P_m$ 分别输入私有集合 $X_1, X_2, \ldots, X_m$, 共同计算 m 个私有集合的交集, 且不透露除交集外的任何信息. MP-PSI[12−14]具有通信轮数多、计算复杂度高、参与方合谋等问题导致 MP-PSI 协议未能部署于实际应用场景中. 现有的 MP-PSI 协议[12,13]主要通过星型网络结构解决通信轮数多问题. 通过不经意可编程伪随机函数 (oblivious programmable pseudo-random function, OPPRF)[13,14]和零共享解决多方合谋问题.

超阈值多方隐私集合求交 (over-threshold multi-party PSI, OT-MP-PSI) 协议是 MP-PSI 问题的变体, 允许 m 个参与方共同计算至少 t ($t \leqslant m$) 个参与方拥有相同元素的超阈值交集, 并且保证仅拥有超阈值元素的参与方才能获取该元素属于超阈值交集的信息, 对于其他信息一无所知. 与文献 [15] 提及的阈值 PSI 协议有所不同, 阈值 PSI 协议的阈值指所有参与方的交集基数, 而超阈值 PSI 协议的阈值指元素出现在参与方集合中的次数. 如图 1 所示, 左图为超阈值 PSI, 右图为阈值 PSI. 可以发现将阈值和超阈值设置为 3 时, 参与方 P_1、P_3、P_4、P_6除了获取阈值交集外, 还获取了仅参与方拥有的超阈值元素信息. OT-MP-PSI 具有广泛的实际应用场景, 例如网络操作中心协作识别常见威胁[16]: 网络中心需要设置常见威胁且不泄露网络中心的其他信息, 可以采用 MP-PSI 协议将全部网络中心遭受的共同攻击设置为常见威胁, 但该常见威胁的设置过于严格, 因此产生了 OT-MP-PSI 协议的实际应用, 只要 t 个网络中心被同一攻击威胁, 就被设置为常见威胁. 药店协作识别购买同一药物危害个人健康[17]也属于同一应用场景.



(a) 超阈值 PSI (b) 阈值 PSI

图 1 PSI 实例说明图

本文提出了一个高效的双云辅助 OT-MP-PSI 协议, 将秘密分发方和重构方任务交付给不可信云完成, 实现了弱计算能力的参与方也能完成 OT-MP-PSI 协议. 相较于目前最优的 OT-MP-PSI 协议[18], 本方案参与实体的通信复杂度不再与阈值有关, 计算复杂度仅在对称密码下计算而非在公钥密码的指数上计算. 本文的主要贡献如下.

(1) 提出了一个新密码学组件: 不经意可编程伪随机秘密共享 (oblivious programmable pseudo-random secret sharing, OPPR-SS). OPPR-SS 由不经意可编程伪随机函数和 Shamir 秘密共享 (secret sharing, SS) 组合实现. OPPR-SS 具有 OPPRF 的安全性、可编程性和秘密共享的 (t, m) 重构性, 可完成参与方编码指定元素定向输出, 且输出值具有 (t, m) 重构性.

(2) 提出了双云辅助 OT-MP-PSI 协议: 考虑到秘密分发方和重构方因计算能力有限或通信能力受限等问题无法完成 OT-MP-PSI 协议, 设计了一个双云辅助 OT-MP-PSI 协议, 双云分别承担繁重的数据共享生成和重构任务. 参与方的通信轮数仅 3 轮, 通信复杂度仅 $O(n)$, 秘密分发方的计算复杂度仅 $O(sym \cdot nm)$, 重构方的计算复杂度仅 $O(sym \cdot n(m \cdot \log n/t)^{2t})$.

(3) 通过实验仿真, 实现了双云辅助 OT-MP-PSI 协议, 分别在集合大小 n={10, 10^2, 10^3, 10^4, 10^5, 10^6}和超阈值 t={2, 3, 4, 5, 6, 7}的情况下测试协议的性能表现, 详细地给出了现存 OT-MP-PSI 协议的性能对比, 本文协议的运行时间和通信负载具有显著优势.

1 相关工作

现有的隐私集合求交协议主要围绕两方面向计算效率快、通信负载低、安全模型高等目标展开研究: 一方面是安全多方计算框架的研究, 另一方面是数据打包技术的研究.

安全多方计算框架主要分为 3 大类: 公钥加密框架、混淆电路框架和不经意传输框架. 公钥加密框架具有计算昂贵、通信负载低等特点, Rosulek 等人[19]基于 Diffe-Hellman 密钥交换实现集合大小小于 500 时, 通信负载和计算效率最优的 PSI 协议. 混淆电路 (garbled circuit, GC) 框架具有通用性的特点, Chandran 等人[20]实现了在交集基础上安全执行求基数、关联属性之和的电路 PSI 协议. 不经意传输 (oblivious transfer, OT) 框架具有计算效率快, 通信负载较低的特点. Kolesnikov 等人[1]基于 OT 扩展安全框架构造出第 1 个大集合下实用的 PSI 协议, 目前仍是高带宽下计算效率最快的 PSI 协议. Chase 等人[2]通过更改 Kolesnikov 等人[1]构造的 OPRF 密码学组件实现普通网络带宽下计算效率最快的 PSI 协议.

数据打包技术通过优化数据结构的存储空间、编码时间和解码时间来提高协议性能. Pinkas 等人[21]基于多项式插值技术打包集合降低通信负载, 多项式打包技术的存储空间为 $O(n)$, 编解码时间为 $O(n\log n^2)$. 周素芳等人[22]基于多项式性质实现无安全多方计算框架的隐私集合求交协议. Dong 等人[3]基于布隆过滤器构造的混淆布隆过滤器技术打包集合, 存储空间为 $O(n/\lambda)$, 编解码时间为 $O(n\lambda)$. Pinkas 等人[23]基于布谷鸟哈希算法构造的 PaXoS (probe and XOR of strings, PaXoS) 技术打包集合, 存储空间和编解码时间为现有 PSI 协议中最优. 为进一步提高数据结构的效率, Pinkas 等人[24]采用布谷鸟哈希算法将数据结构大小下降到 $O(\log n/\log\log n)$, Pinkas 等人[25]采用 2 维布谷鸟哈希算法将数据结构大小下降到次常数项.

OT-MP-PSI 是隐私集合求交的变体, 仅存在少量的研究. Kissner 等人[17]首次提出 OT-MP-PSI 问题, 采用多项式插值技术和公钥加密安全框架实现 OT-MP-PSI 协议, 但通信复杂度随参与方个数呈 3 次方增长, 计算复杂度在公钥操作上呈指数增长. Mahdavi 等人[18]尝试采用现有的 MP-PSI 协议[26]解决 OT-MP-PSI 问题, 但需要执行 C_m^t 次 MP-PSI 协议, 在实际部署时无法接受. 随后提出了基于公钥加密结合秘密共享构建安全且具有 (t, m) 重构性的新密码学组件不经意伪随机秘密共享 (oblivious pseudo-random secret sharing, OPR-SS). 秘密分发方通过 OPR-SS 组件为参与方的每个元素执行公钥指数上的 (t, m) 零共享, 保证秘密分发方不知晓元素值且参与方对共享值无法区分. 重构方对参与方发送的朴素哈希表进行 (t, m) 重构, 若多项式插值结果为 0 则该元素至少在 t 个参与方中拥有. 基于 OPR-SS 的 OT-MP-PSI 协议具有高效的通信轮数 3 轮和通信复杂度 $O(nmt)$, 但其计算复杂度仍在公钥密码操作上呈指数增长. Chandran 等人[27]基于 OPPRF 的安全比较协议和 Bay 等人[28]基于 TPKE 的安全比较协议实现类似 OT-MP-PSI 协议的功能: 只能计算指定方的集合中至少存在于 $t-1$ 个参与方集合中的元素. Chandran 等人[27]的协议和 Bay 等人[28]的协议分别在大集合场景和小集合场景下具有高效运行时间和低通信成本. 当我们试图将指定方集合设定为元素域时, 指定方可计算元素域中至少存在于 $t-1$ 个集合中的元素, 但无法实现仅拥有该元素的参与方才能获得超阈值元素信息的功能.

2 系统架构与安全模型

2.1 系统架构

目前实用 OT-MP-PSI 协议[18]涉及 3 类实体: 秘密分发方、参与方和重构方. 由于秘密分发方负责秘密生成工作, 重构方负责秘密重构工作, 涉及非常大的运算量和通信量, 对产生大量数据的终端设备是不能接受的. 因此, 本文引入两个辅助云分别替代秘密分发方和重构方, 实现弱计算能力的参与方也能完成 OT-MP-PSI 协议.

m (m>2) 个参与方 P_i 拥有集合 X_i. 参与方与云辅助 C_1 执行 OPPR-SS 协议, C_1 为参与方的元素生成秘密共享值. 参与方 P_i 将秘密共享值打包给云辅助 C_2, C_2 本地重构秘密共享值并将结果发送给参与方. 最终, 参与方获得超阈值交集, 且保证仅拥有超阈值元素的参与方才能知晓该元素是否属于超阈值交集, 对于其他信息一无所知. 云场景系统模型如图 2 所示.

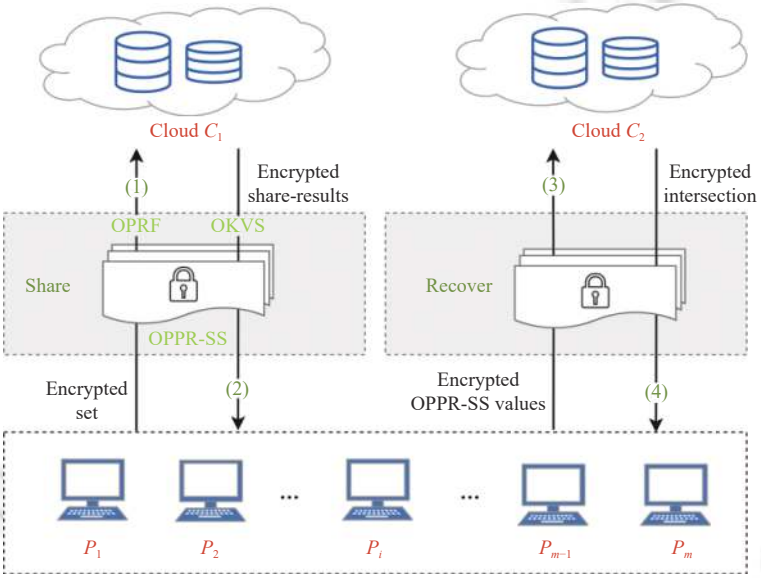


图 2 双云辅助的 OT-MP-PSI 系统架构

2.2 安全模型

本文安全模型假设与文献 [18] 一致, 包含 m (m>2) 个参与方、半诚实敌手 A 控制参与方集合 $C \leqslant m-2$、共享方和重构方. 半诚实敌手 A 允许获取 C 中参与方的所有信息, 并挖掘潜在信息, 但无法干预参与方诚实的执行协议. 重构方不与任何一方合谋. PSI 是基于安全多方计算框架构建的协议, 一般采用理想-现实模型 (ideal-real model)[29]进行安全性证明. 令 F 为 PSI 协议 π 的功能函数, f_i 为构建 F 的安全多方计算底层功能函数. 证明在 f_i-混合模式下模拟者 (Sim) 运行协议 π 的攻击者视角 Sim_F (仅包括输入和输出), 与直接运行协议 π 的攻击者视角 $View_\Pi$ (输入、随机数以及整个协议期间收到的所有消息) 不可区分, 即可证明协议 π 安全实现功能函数 F.

● 混合模式: 如果协议 π_i 安全实现功能函数 f_i, 协议 π 利用协议 π_i 安全实现功能函数 F, 则称协议 π 在 f_i-混合模式下安全实现功能函数 F.

● 理想-现实模型: P_i 分别拥有私有输入集 $X_i=\{x_{i,1},x_{i,2},\ldots,x_{i,n}\}$, 输出集 $Y_i=\{y_{i,1},y_{i,2},\ldots,y_{i,n}\}$, $i\in[m]$. Π 是 PSI 协议, F 是 PSI 功能函数, C 为敌手控制参与方集合 (C 一般取 $m/2$, $m/3$, $m-1$), 安全参数 k.

现实模型 $View_\Pi(k,C;X_1,X_2,\ldots,X_m)$: m 个参与方 P_i 分别输入 X_i 到 Π, 参与方共同计算 $(Y_1,Y_2,\ldots,Y_m)\leftarrow\Pi(X_1,X_2,\ldots,X_m)$, 令 $View_i$ 作为 P_i 最终视图 $\{View_i|i\in C\}$.

理想模型 $Sim_F(k,C;X_1,X_2,\ldots,X_m)$: 定义 Π 所需安全性要求 S, m 个参与方 P_i 分别输入 X_i 到可信第三方 T 计算 $(Y_1,Y_2,\ldots,Y_m)\leftarrow F(X_1,X_2,\ldots,X_m)$. 模拟者 Sim 模仿现实模型中敌手视图: $Sim(C,\{(X_i,Y_i)|i\in C\})$.

● 安全性: 若 $View_{\Pi}(k,C;X_1,X_2,\ldots,X_m)$ 与 $Sim_F(k,C;X_1,X_2,\ldots,X_m)$ 的输出在安全参数 k 下不可区分, 则认为协议 Π 在安全性要求 S 下安全实现功能函数 F.

3 基础知识

3.1 不经意键值对存储

键值对存储 (key-value stores, KVS) 是一种表示键值对正确映射的紧凑型数据结构. 当随机输入键 K 至数据结构, 输出值 V 无法区分时, 称数据结构为不经意键值对存储 (oblivious key-value stores, OKVS)[30]. OKVS 形式化描述如下.

● 参数: 键集合 $K=(k_1,k_2,\ldots,k_m)$, 值集合 $V=(v_1,v_2,\ldots,v_m)$, 一簇哈希函数 H, 敌手 A.

● OKVS 结构:

$Encode_H$: 输入键值对集 (K, V), 输出数据结构 $\mathbb{S}$ ($Encode_H(K,V)=\perp$ 可忽略)

$Decode_H$: 输入数据结构 $\mathbb{S}$ 和键 k, 输出值 v.

● OKVS 正确性:

如果 $k\in K$, 则 $v=Decode_H(\mathbb{S},k)\in V$.

● OKVS 完备性:

敌手 A 拥有两组键值对 $(K_1,V_1),(K_2,V_2)$, 对其进行 $Encode_H$ 得到数据结构 $\mathbb{S}_1$ 和 $\mathbb{S}_2$. 敌手 A 对 $\mathbb{S}_1$ 和 $\mathbb{S}_2$ 无法区分.

OKVS 具有隐藏键的特性且结构紧凑, 在 PSI 协议中用作存储集合元素的数据结构, 如多项式[21,22]、布隆过滤器[31,32]、PaXoS[23,33] 等. 然而, 仅利用 OKVS 的隐藏特性构建 PSI 协议不具有安全性, OKVS 接收方仅受自身计算能力限制查询次数, 会泄露发送方未在交集中的元素信息. 常见的方案结合公钥加密[34]、不经意传输[35,36]、混淆电路[37] 等安全框架限制查询次数.

3.2 秘密共享

秘密共享 (secret sharing, SS) 指秘密值 s 通过秘密分配算法 $Share(s,m)$ 分发给 m 个参与方, t ($t\leqslant m$) 个参与方通过秘密重构算法 $Recon(t,S_t)$ 重构秘密 s, 小于 t 个参与方无法重构秘密 s. 本文采用基于多项式插值的 Shamir 秘密共享[38], 具体步骤如下.

● 参数: 秘密值 s, 阈值 t, m 个参与方 P_i

(1) 秘密分配 $Share(s,m)$

随机选择 $t-1$ 个随机值 $r_1,r_2,\ldots,r_{t-1}$, 构造多项式 $f(x)=s+r_1x+\ldots+r_{t-1}x^{t-1}$. 随机选择 m 个随机数 $x_1,x_2,\ldots,x_m$, 计算秘密碎片 $s_i=f(x_i)$ 并将 (x_i,s_i) 发送给参与方 P_i.

(2) 秘密重构 $Recon(t,S_t)$

任意 t 个参与方通过拉格朗日插值恢复秘密 s:

$$s=\sum_{i\in S_t}\left[s_i\cdot\prod_{j\in S_t,j\neq i}\frac{-x_j}{x_i-x_j}\right].$$

3.3 Hash-to-bin

哈希技术是降低协议复杂度的重要工具. 本文采用接收方执行布谷鸟哈希算法, 发送方执行朴素哈希算法来降低 OKVS 的编解码时间以及减少元素的比较次数. 布谷鸟哈希算法和朴素哈希算法如算法 1 和算法 2.

● 参数: k 个哈希函数 $Hash_k:\{h_1,h_2,\ldots,h_k\}$, 集合 $X:\{x_1,x_2,\ldots,x_n\}$, 布谷鸟哈希表 TC 行 b 列 1, 朴素哈希表 TS 行 b 列 $b_{\max}$.

算法 1. 布谷鸟哈希算法.

输入: 集合 $X:\{x_1,x_2,\ldots,x_n\}$;

输出: 布谷鸟哈希表 TC.

1. Insert(x_i){ //元素 $x_i \in X$ 填入表 TC 中某一个位置
2. 计算 $Hash_k(x_i)$ 查看 $TC[Hash_k(x_i)]$;
3. if ($TC[Hash_k(x_i)] = \perp$) //如果至少有一个位置为空
4. $TC[Hash_k(x_i)] \leftarrow x_i$ //随机选择一个位置插入元素 x_i;
5. else ($TC[Hash_k(x_i)] \neq \perp$){ //如果都不为空
6. $x' \leftarrow TC[Hash_k(x_i)]$; //随机选择一个位置插入元素 x_i, 替代元素 x';
7. $TC[Hash_k(x_i)] \leftarrow x_i$;
8. Insert(x');} //对元素 x' 进行如上操作;
9. endif;}

算法 2. 朴素哈希算法.

输入: 集合 $X: \{x_1, x_2, \ldots, x_n\}$;
输出: 朴素哈希表 TS.

1. for i=1 to n do
2. 计算 $Hash_k(x_i)$ 将 $TS[Hash_k(x_i)][r] \leftarrow x_i$; //将 x_i 存储在表 TS 的 k 个位置, $r \in [b_{\max}]$.
3. 对于表 TS 中未填充的位置使用不同的虚拟元素 $d \notin X$ 填满.

3.4 不经意可编程伪随机函数

不经意可编程伪随机函数 (OPPRF) 是不经意伪随机函数 (OPRF) 的一种变体 (分别如图 3 和图 4 所示). OPPRF 在 OPRF 的基础上, 添加特定输入产生特定输出的额外属性. 现有 OPPRF 协议采用 OPRF 和 OKVS 实例化. Kolesnikov 等人[13]采用基于 OT 的 OPRF[1] 和基于 Table 的 OKVS 实现 OPPRF. Rindal 等人[33]采用基于 VOLE 的 OPRF 和基于 PaXoS 的 OKVS[23]实现 OPPRF 功能.

图 3 不经意伪随机函数

图 4 不经意可编程伪随机函数

OPRF 理想功能 F_{OPRF} 如下.

● 参数: 元素域 $\mathcal{F}$, 伪随机函数 $F: \mathcal{F} \to \{0,1\}^k$, 计算安全参数 κ, 发送方 S, 接收方 R.

● 功能: R 输入元素 $x \in \mathcal{F}$, 输出 $F_k(x)$. S 输出伪随机函数密钥 k.

OPPRF 理想功能 F_{OPPRF} 如下.

● 参数: 元素域 $\mathcal{F}$, 伪随机函数 $F: \mathcal{F} \to \{0,1\}^k$, 计算安全参数 κ, 发送方 S 拥有一组私有集合对 $Y = \{(y_1, z_1), (y_2, z_2), \ldots, (y_n, z_n)\}$, $y_i \in \mathcal{F}$, $z_i \in \{0,1\}^k$, 接收方 R 拥有一组私有集合 $X = \{x_1, x_2, \ldots, x_n\}$, $x_i \in \mathcal{F}$, $i = \{1, 2, \ldots, n\}$.

● 功能: S 输入 $Y = \{(y_1, z_1), \ldots, (y_n, z_n)\}$, 输出伪随机函数密钥 k. R 输入元素 x_i, 输出 $\{F_k(x_i), F_k(x_i) \oplus r\}$. 若 $x_i = y_j$, 则 $r = z_j$, 否则 r 是一个随机值, $j \in [n]$.

4 超阈值多方隐私集合求交协议

本节首先提出了一个新密码学组件 OPPR-SS, 仅使用对称密码实现了 OPR-SS 功能. 其次考虑参与方计算能力或通信能力受限等问题, 引入不可信云承担秘密值共享和重构工作, 最终构建 OT-MP-PSI 协议.

4.1 不经意可编程伪随机秘密共享

不经意可编程伪随机秘密共享具有以下功能: 发送方对私有集合 Y 编程得到具有 (t, m) 重构特性的输出值.

接收方输入元素 e, 若 $e \in Y$, 则输出集合 Y 中对应元素的 (t, m) 秘密共享值, 否则输出随机值. 功能函数 $F_{\text{OPPR-SS}}^{S,R}$ 形式化描述如下.

● 参数: 发送方 S 持有私有集合 $Y=\{y_1,y_2,\ldots,y_n\}$, 接收方 R 有元素 e, 阈值 t, $Idx(R)$ 为 R 标号, $Idx(S)$ 为 S 标号.

● 输入: S 输入私有集合 Y, R 输入元素 e.

● 计算: $F_{\text{OPPR-SS}}^{S,R}$ 为元素 y_i 选择 $t-1$ 个随机数 $r_1,r_2,\ldots,r_{t-1}, i=\{1,2,\ldots,n\}$, 构建多项式 $P_i(x)=0+r_1x+\ldots+r_{t-1}x^{t-1}$;

● 输出: 对于 R: 如果 $e=y_i$ 输出 $P_i(Idx(R))$; 否则输出随机值. 对于 S: 输出 $P_i(Idx(S))$, $i=\{1,2,\ldots,n\}$.

$F_{\text{OPPR-SS}}^{S,R}$ 由 OPPRF 和秘密共享实例化为新密码学组件 OPPR-SS. OPPR-SS 具有 OPPRF 的安全性、可编程性以及秘密共享的共享生成和重构性. OPPR-SS 组件可实现持有相同元素的参与方才能获得具有 (t, m) 重构性的编程值, 否则获得随机值的功能. $\Pi_{\text{OPPR-SS}}^{S,R}$ 构造如下.

协议 1. 不经意可编程伪随机秘密共享 ($\Pi_{\text{OPPR-SS}}^{S,R}$).

● 参数: 发送方 S 持有私有集合 $Y=\{y_1,y_2,\ldots,y_n\}$, 接收方 R 持有元素 e, 伪随机函数 $F:\mathcal{F}\to\{0,1\}^k$, 计算安全参数 κ, OKVS 编码函数 $Encode_H()$, OKVS 解码函数 $Decode_H()$, k 个哈希函数 $Hash_k:\{h_1,h_2,\ldots,h_k\}$.

● 输入: S 输入私有集合 $Y=\{y_1,y_2,\ldots,y_n\}$, R 输入元素 e.

● 输出: R 输出 OPPR-SS(e).

(1) OPRF 阶段 (R 与 S 执行 1 次 F_{OPRF}):

R 输出 OPRF 值 $F_k(e)$.

S 输出 OPRF 密钥 k.

(2) SS-Share 阶段 (S 本地编程元素的秘密共享值):

S 为每个元素随机生成 t–1 个随机值 y_i: $\{r_{i,1},r_{i,2},\ldots,r_{i,t-1}\}$, $i=\{1,2,\ldots,n\}$.

S 为每个元素构建一个常数项为 0 的多项式 $P_{y_i}(x)=0+r_{i,1}x+\ldots+r_{i,t-1}x^{t-1}$.

(3) Hash-to-bin 阶段:

S 使用 $Hash_k(y_i)$ 作为索引, 将 $\{F_k(y_i)+P_{y_i}(Idx(R))\}$ 映射到朴素哈希表 TS 中的 k 个位置. TS 长为 b, 宽为 $b_{\max}$.

R 使用相同的 $Hash_k(e)$ 作为索引, 将 $F_k(e)$ 映射到布谷鸟哈希表 TC 中的一个位置. TC 长为 b, 宽为 1.

(4) OKVS 阶段:

S 使用 OKVS 编码函数得到 $\mathbb{S}_i \leftarrow Encode_H(TS_i), i=\{1,2,\ldots,b\}$, 并将 $\mathbb{S}_i$ 发送给 R.

R 计算元素 e 所在 TC 的位置 j, 求解 OPPR-SS 值 $x'=Decode_H(\mathbb{S}_j,e)\oplus F_k(e)$.

4.2 双云辅助的 OT-MP-PSI 协议

本文考虑到秘密分发方和重构方计算能力有限或通信能力受限等问题无法承担 OT-MP-PSI 协议开销, 提出由不可信云承担数据共享和数据重构任务, 设计实现双云辅助的 OT-MP-PSI 协议, 超阈值交集元素仅返回给拥有该元素的参与方. 双云辅助的 OT-MP-PSI 功能 $F_{\text{OT-MP-PSI}}^{m,n,C}$ 形式化描述如下.

● 参数: m 个参与方 $P_1,P_2,\ldots,P_n$, 两个不可信云 C_1,C_2.

C_1: P_i 输入集合大小为 n 的集合 X_i 给 C_1, C_1 计算共享值 S_{X_i} 并发送给 P_i.

C_2: P_i 将共享值 S_{X_i} 发送给 C_2, C_2 计算 $\bigcap_{i=1}^{t}S_{X_i}$, 若重构结果正确, 则将 S_{X_i} 返回 P_i.

双云辅助 OT-MP-PSI 协议采用 OPPR-SS 组件作为构建块, 分为两个主要阶段进行构建.

● 共享阶段: 参与方 P_i 与云 C_1 执行 OPPR-SS 协议使得 P_i 得到 X_i 的 (t, m) 共享值集合 S^i. 在共享阶段中, 云 C_1 为元素域 $\mathcal{F}$ 中的每个元素随机生成 $t-1$ 个随机数, 用于构建 t 阶常数项为 0 的多项式 $P(x)$, 以实现所有参与方的元素秘密共享值生成. 参与方 P_i 分别与云 C_1 执行 OPPR-SS 协议: 参与方 P_i 与云 C_1 执行 n 次 OPRF 实例, 参与方输入集合元素 $\{x_{i,1},x_{i,2},\ldots,x_{i,n}\}$, 输出 n 个 OPRF 值 $F_{k_i}(x_{i,j})$, 云 C_1 输出密钥 k_i. 云 C_1 本地计算元素域 $\mathcal{F}$ 中元素 x_k 的 OPRF 值 $F_{k_i}(x_k)$, 并进行 OKVS编码 $\{x_k,F_{k_i}(x_k)\oplus P_k(i)\}$ 得到数据结构 $\mathbb{S}_i$, 将 $\mathbb{S}_i$ 发送给 P_i. P_i 本地计算 OPPR-SS 值 $\{F_{k_i}(x_{i,j})\oplus Decode(\mathbb{S}_i,x_{i,j})\}$.

● 重构阶段: 参与方 P_i 将朴素哈希表发送给云 C_2, 对共享值进行 (t, m) 重构, 满足正确重构值则为超阈值交集元素并将 t 个共享值原路返回给参与方. 在重构阶段中, P_i 使用相同的 $Hash_k$ 将 $x_{i,j}$ 共享值 $\{F_{k_i}(x_{i,j})\oplus Decode(\mathbb{S}_i, x_{i,j})\}$ 存储在以 $Hash_k(x_{i,j})$ 为索引的朴素哈希表中, 并发送给云 C_2. 云 C_2 从 m 个哈希表中选择 t 个, 从 t 个表中选择其中一行, 从行中选择一个值进行多项式插值. 若插值结果为 0, 则将插值返回给对应参与方 P_i.

双云辅助的 OT-MP-PSI 协议 $\Pi^{m,n,C}_{\text{OT-MP-PSI}}$ 构造如下.

协议 2. 双云辅助的超阈值多方隐私集合求交协议 ($\Pi^{m,n,C}_{\text{OT-MP-PSI}}$).

● 参数: 不可信云 C_1, C_2, m 个参与方 $P_1, P_2, \ldots, P_m$, P_i 拥有集合 $\{x_{i,1}, x_{i,2}, \ldots, x_{i,n}\}$, n 为集合大小, $\mathfrak{F}$ 为元素域, t 为超阈值, k 个哈希函数 $Hash_k: \{h_1, h_2, \ldots, h_k\}$. 哈希表 TS 长为 b 宽为 $b_{\max}$.

(1) 共享阶段

C_1 与 P_i 分别执行 OPPR-SS 协议, $i \in \{1,2,\ldots,m\}$:

输入: C_1 无输入, 接收方 P_i 输入集合元素: $\{x_{i,1}, x_{i,2}, \ldots, x_{i,n}\}$.

输出: P_i 输出集合 $\{\text{OPPR-SS}(x_{i,j})\}$, 其中 $\text{OPPR-SS}(x_{i,j})$ 等于 $\{F_{k_i}(x_{i,j})\oplus Decode(\mathbb{S}_i, x_{i,j})\}$, $j \in \{1,2,\ldots,n\}$.

(2) 重构阶段

① 对于每一个 P_i, 依据朴素哈希算法通过 $Hash_k(x_{i,j})$ 将 $\{F_{k_i}(x_{i,j})\oplus Decode_H(\mathbb{S}_j, x_{i,j})\}$ 存储在朴素哈希表 TS_i 中, 并将其发送给 C_2;

② C_2 计算超阈值交集方法如下.

```
for m 个 TS_i 中选取 t 子集:
  for b_k, k = 1,2,...,b //从朴素哈希表中选择一行
    for b_{max,h}, h = 1,2,...,b_max //从一行中选择一个元素
      if t 个 {i, TS_i[b_k][b_{max,h}]} 拉格朗日插值结果为 0
        return F_{k_i}(x_{i,j}) ⊕ Decode_H(S_j, x_{i,j}) //返回给 P_i;
      endif
    endfor
  endfor
endfor
```

5 安全性证明

5.1 OPPR-SS 协议安全性证明

定理 1. 在 F_{OPRF} 混合模式下, $\Pi^{S,R}_{\text{OPP-SS}}$ 实现了针对半诚实敌手的 $F^{S,R}_{\text{OPPR-SS}}$ 功能.

证明: 本证明采用理想-现实模型进行形式化证明, 首先在理想模型下运行现实模型构造半诚实发送方 S 和半诚实接收方 R 的模拟器 $Sim^S_{\text{OPPR-SS}}$ 和 $Sim^R_{\text{OPPR-SS}}$. 通过执行 $\Pi^{S,R}_{\text{OPPR-SS}}$ 得到半诚实发送方 S 和半诚实接收方 R 的视图 $View^S_{\text{OPPR-SS}}$ 和 $View^R_{\text{OPPR-SS}}$. 最后证明 $Sim^S_{\text{OPPR-SS}}$ 与 $View^S_{\text{OPPR-SS}}$、$Sim^R_{\text{OPPR-SS}}$ 与 $View^R_{\text{OPPR-SS}}$ 不可区分.

(1) 模拟器生成接收方 R 的 $Sim^R_{\text{OPPR-SS}}$: OPRF 阶段模拟器发送元素 x 给 F_{OPRF}, 然后随机选取 OPRF 的输出值 $F_k(x)$. R 在 OPRF 阶段的视图为 $Sim^R_{\text{OPPR-SS}}(x, F_k(x))$. 秘密共享阶段和 Hash-to-bin 阶段, R 无信息交互, 视图仍为 $Sim^R_{\text{OPPR-SS}}(x, F_k(x))$. OKVS 阶段模拟器随机编码 b 个数据结构 $\mathbb{S} = \{\mathbb{S}_1, \mathbb{S}_2, \ldots, \mathbb{S}_b\}$ 模拟 S 的 $Encode_H(T1_i)$, $i \in \{1,2,\ldots,b\}$, 并计算 $\mathbb{S}(x)\oplus F_k(x)$ 得到秘密共享值 x', 将这些获得的信息添加到模拟视图得到最终视图 $Sim^R_{\text{OPPR-SS}}(x, F_k(x), x', \mathbb{S})$. 为证明其和 $View^R_{\text{OPPR-SS}}$ 不可区分, 我们对 $F_k(x)$、$\mathbb{S}$、x' 这 3 个输出值进行不可区分证明.

$F_k(x)$: 由 OPRF 协议的安全性可知 $F_k(x)$ 均匀分布不可区分.

$\mathbb{S}$: 由 OKVS 协议的安全性可知均匀分布不可区分.

x': $\mathbb{S}$ 中的值由 OPRF 值和秘密共享值组成均匀分布不可区分. 若 $x \in Y$, 则 x' 为秘密共享值, 否则为不可区分

的随机值.

(2) 模拟器生成发送方 S 的 $Sim^{S}_{\text{OPPR-SS}}$: OPRF 阶段模拟器为发送方 S 随机选择 OPRF 密钥 k, S 在 OPRF 阶段的视图为 $Sim^{S}_{\text{OPPR-SS}}(k)$. 剩余阶段接收方 R 无消息发送, S 在剩余阶段无输出, 由 OPRF 协议安全性可知 k 仅发送方 S 知道, 故 $Sim^{R}_{\text{OPPR-SS}}$ 与 $View^{R}_{\text{OPPR-SS}}$ 不可区分.

5.2 云辅助的 OT-MP-PSI 协议安全性证明

定理 2. 在 $F_{\text{OPPR-SS}}$ 混合模式下, $\Pi^{m,n,C}_{\text{OT-MP-PSI}}$ 实现了针对半诚实敌手的 $F^{m,n,C}_{\text{OT-MP-PSI}}$ 功能.

证明: 正确性. 假设 Y_1 为真实超阈值交集, Y_2 为协议 $\Pi^{m,n,C}_{\text{OT-MP-PSI}}$ 输出超阈值交集, 需证明 $Y_1 = Y_2$. 从两方面论证.

(1) 证明 $Y_1 \subseteq Y_2$. 假设元素 $e \in Y_1$, 云 C_1 为元素 e 构造 (t, m) 秘密共享多项式 $P(\cdot)$. 由 OPRF 协议的正确性可知参与方 P_i 获得 $F_{k_i}(e)$. 通过哈希函数的映射属性可知, 布谷鸟哈希算法和朴素哈希算法映射同一元素, 元素必定在布谷鸟哈希表和朴素哈希表的同一行. 因此云对每一行进行 OKVS 编码并发送给参与方 P_i, P_i 输入 e, 由 OKVS 的性质异或 $F_{k_i}(e)$ 后得到元素 e 的秘密共享值 $P(i)$. 由朴素哈希算法性质可知, 若 t 个参与方拥有相同元素, 则 $TS_i(j), j = \{Hash_k(e)\}$ 必在同一行, 因此云 C_2 对 t 个进行重构插值后得到 0. 因此 $e \in Y_2$, 由此证明 $Y_1 \subseteq Y_2$.

(2) 证明 $Y_2 \subseteq Y_1$. 假设元素 $e \in Y_2$ 但 $e \notin Y_1$. $e \in Y_2$ 则重构阶段拉格朗日插值结果为 0, 由 OPPR-SS 的正确性可知只有参与方和云 C_1 拥有相同元素时, 参与方才能获得对应的多项式值 $P(\cdot)$, 因此错误只可能发生在构建朴素哈希表时添加的虚拟元素 d 与参与方持有元素相同. 构造时明确规定虚拟元素与集合元素不同, 出现的概率为 $2^{-\delta}$ (δ 为统计安全参数), 是可忽略不计的. 因此该假设不成立, 则 $Y_2 \subseteq Y_1$.

综合 (1)(2) 可知, $Y_1 \subseteq Y_2$ 和 $Y_2 \subseteq Y_1$, 得到 $Y_1 = Y_2$.

安全性. 设 C 为合谋集合 ($|C|<m-2$). 本文协议可分为 4 类实体, 即合谋集合 C、参与方 $P_i, i \in \{1,2,\ldots,m\} \cap i \notin C$、云 C_1 (可属于 C) 和云 C_2. 假设所有参与方和云诚实遵循协议步骤但会主动收集和推导信息. 云 C_2 不与任何参与方合谋. 以下证明 4 类实体分别在理想模型中按照 $\Pi^{m,n,C}_{\text{OT-MP-PSI}}$ 协议过程获得的模拟视图与真实协议视图是不可区分的.

(1) 模拟器生成参与方 $P_{i,i\in\{1,2,\ldots,m\}\cap i\notin C}$ 的 $Sim^{P}_{\text{OT-MP-PSI}}$:

共享阶段: 模拟器模拟执行 OPPR-SS 协议输入 $X_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,n}\}$ 和输出 $F_{k_i}(X_i)$、$\mathbb{S}_i$. 更新 $Sim^{P}_{\text{OT-MP-PSI}}(X_i, F_{k_i}(X_i), \mathbb{S}_i)$.

重构阶段: 模拟器模拟执行重构协议输入 TS_i 和输出 $Y_i = \{F_{k_i}(x_{i,j}) \oplus \mathbb{S}_i(x_{i,j})\}$ ($x_{i,j}$ 为超阈值交集元素). 更新 $Sim^{P}_{\text{OT-MP-PSI}}(X_i, F_{k_i}(X_i), \mathbb{S}_i, TS_i, Y_i)$.

为证明 $Sim^{P}_{\text{OT-MP-PSI}}$ 和 $View^{P}_{\text{OT-MP-PSI}}$ 不可区分, 只需证明除协议开始和结束的输入和输出外的值是不可区分的. 由 OPRF 协议安全性保证 $F_{k_i}(X_i)$ 均匀分布不可区分, $\mathbb{S}_i$ 由 OKVS 算法对 $\{x_k, F_{k_i}(x_k) \oplus P_k(i)\}$ 进行编码得到, OKVS 协议保证 x_k 安全性, 且 $F_{k_i}(x_k) \oplus P_k(i)$ 在 P_i 视角下均匀分别不可区分. TS_i 为本地朴素哈希算法计算 $\{F_{k_i}(x_{i,j}) \oplus Decode_H(\mathbb{S}_i, x_{i,j})\}$ 生成, 对于没有密钥 k_i 的云 C_2 来说是均匀分布不可区分. 因此 $Sim^{P}_{\text{OT-MP-PSI}}$ 和 $View^{P}_{\text{OT-MP-PSI}}$ 不可区分.

(2) 模拟器生成云 C_1 的 $Sim_{\text{OT-MP-PSI}}C_1$:

共享阶段: 模拟器模拟执行 OPPR-SS 协议输入 $\mathbb{S} = \{\mathbb{S}_1, \mathbb{S}_2, \ldots, \mathbb{S}_m\}$ 和输出 $k = \{k_1, k_2, \ldots, k_m\}$. 更新 $Sim^{C_1}_{\text{OT-MP-PSI}}(k, \mathbb{S})$.

重构阶段: 云 C_1 不参与, 因此 $Sim^{C_1}_{\text{OT-MP-PSI}}$ 不更新.

为证明 $Sim^{C_1}_{\text{OT-MP-PSI}}$ 和 $View^{C_1}_{\text{OT-MP-PSI}}$ 不可区分. 只需证明输出值 k 的隐私性即可, 由 OPRF 协议保证密钥 k 的安全性, 因此 $Sim^{C_1}_{\text{OT-MP-PSI}}$ 和 $View^{C_1}_{\text{OT-MP-PSI}}$ 不可区分.

(3) 模拟器生成云 C_2 的 $Sim^{C_2}_{\text{OT-MP-PSI}}$:

共享阶段: 云 C_2 不参与, $Sim^{C_2}_{\text{OT-MP-PSI}}$ 不更新.

重构阶段: 模拟器模拟执行重构协议输出 $TS = \{TS_1, TS_2, \ldots, TS_n\}$ 和输入 $Y_i = \{F_{k_i}(x_{i,j}) \oplus \mathbb{S}_i(x_{i,j})\}$ ($x_{i,j}$ 为超阈值交集元素) 给参与方 P_i. 更新 $Sim^{C_2}_{\text{OT-MP-PSI}}(TS, Y_i)$.

为证明 $Sim^{C_2}_{\text{OT-MP-PSI}}$ 和 $View^{C_2}_{\text{OT-MP-PSI}}$ 不可区分, 只需证明除协议开始和结束的输入和输出外的值是不可区分的.

云C_2得到 TS, 由 OPRF 和 OKVS 的安全性可知, TS 对云C_2来说是不可区分的. Y_i 为 TS 中的值, 云C_2无信息泄露. 因此$Sim^{C_2}_{\text{OT-MP-PSI}}$和$View^{C_2}_{\text{OT-MP-PSI}}$是不可区分的.

(4) 模拟器生成合谋集合 C 的$Sim^{C}_{\text{OT-MP-PSI}}$:

情况 1: $C_1 \in C$

共享阶段: 模拟器模拟执行$\Pi^{m,n,C}_{\text{OT-MP-PSI}}$输入 $X_i=\{x_{i,1},x_{i,2},\ldots,x_{i,n}\}$ ($P_i \in C$)、 $\mathbb{S}=\{\mathbb{S}_1,\mathbb{S}_2,\ldots,\mathbb{S}_m\}$和输出 $F_{k_i}(X_i)$、 $\mathbb{S}_i$、$k=\{k_1, k_2, \ldots, k_m\}$. 更新$Sim^{C}_{\text{OT-MP-PSI}}(X_i, F_{k_i}(X_i), \mathbb{S}, k)$.

重构阶段: 模拟器模拟执行重构协议输入 TS_i, $i \in C$ 和输出 $Y_i=\{F_{k_i}(x_{i,j}) \oplus \mathbb{S}_i(x_{i,j})\}$ ($x_{i,j}$ 为超阈值交集元素). 更新$Sim^{C}_{\text{OT-MP-PSI}}(X_i, F_{k_i}(X_i), \mathbb{S}, k, TS_i, Y_i)$.

由 (1) 和 (2) 的模拟结果证明未合谋不可区分. 考虑 m–2 个最大合谋方, 假如 $x_{i,j}$ 为超阈值交集元素, C 中只有 t–1 个参与方拥有该元素, 也无法判断另两个参与方谁拥有该元素. 因此$Sim^{C}_{\text{OT-MP-PSI}}$和$View^{C}_{\text{OT-MP-PSI}}$是不可区分的.

情况 2: $C_1 \notin C$

共享阶段: 模拟器模拟执行$\Pi^{m,n,C}_{\text{OT-MP-PSI}}$输入 $X=\{X_i\}$ 和输出 $F=\{F_{k_i}(X_i)\}$、 $\mathbb{S}=\{\mathbb{S}_i\}$ ($P_i \in C$). 更新$Sim^{C}_{\text{OT-MP-PSI}}(X, \mathbb{S}, F)$.

重构阶段: 模拟器模拟执行重构协议输入 $TS=\{TS_i\}$, ($P_i \in C$) 和输出 $Y=\{Y_i=\{F_{k_i}(x_{i,j}) \oplus \mathbb{S}_i(x_{i,j})\}$ ($x_{i,j}$ 为超阈值交集元素). 更新$Sim^{C}_{\text{OT-MP-PSI}}(X, \mathbb{S}, F, TS, Y)$.

由 (1) 的模拟结果证明未合谋不可区分. 考虑 m–2 个最大合谋方, 假如 $x_{i,j}$ 为超阈值交集元素, C 中只有 t–1 个参与方拥有该元素, 也无法判断另两个参与方谁拥有该元素. 因此$Sim^{C}_{\text{OT-MP-PSI}}$和$View^{C}_{\text{OT-MP-PSI}}$是不可区分的.

6 效率分析

6.1 理论分析

协议的效率主要由通信轮数、计算复杂度和通信复杂度进行定性评估. 分析云辅助 OT-MP-PSI 协议的通信轮数、计算复杂度和通信复杂度, 并与现有解决 OT-MP-PSI 问题的方案进行对比, 以证明本文协议定性优势.

(1) 通信轮数: 本方案由两阶段组成. 共享阶段, OPRF 协议采用 Kolesnikov 等人[1]基于 OT 扩展实现的 OPRF 需要一轮通信. 云C_1将 OKVS 编码发送给各参与方 P_i 需要一轮通信. 重构阶段, 参与方 P_i 将朴素哈希表发送给云C_2, 云C_2重构并将结果返回给参与方需要一轮通信. 参与方通信轮数共计 3 轮. 云C_1仅参与共享阶段, 通信轮数为 2 轮. 云C_2仅参与重构阶段, 通信轮数为 1 轮.

(2) 通信复杂度: 共享阶段, 执行 OPRF 协议参与方发送 n 个元素, 接收 n 个 OPRF 值, 通信量为 $2n$. 执行 OKVS 协议参与方无发送, 接收 2.4n. 重构阶段, 执行 OKVS 协议参与方发送 2.4n, 接收小于阈值 t. 通信复杂度约为 $O(6.8n+t)$. 云C_1仅参与共享阶段, 通信复杂度为 O(2.4nm+2n). 云C_2仅参与重构阶段, 通信复杂度为 O(2.4nm+tm).

(3) 计算复杂度: 共享阶段, 参与方执行 OKVS 解码协议计算复杂度为 O(n). 重构阶段, 参与方构造朴素哈希表计算复杂度为 O(n). 因此参与方的计算复杂度约为 O(n). 云C_1仅参与共享阶段, 计算复杂度为 O(nm). 云C_2仅参与重构阶段, 由重构算法可知重构方需重构所有的 t 元素组合, 每个元素来自不同参与方的朴素 hash 表中. 因此从 m 个参与方中选择 t 个参与方的组合大小为$\binom{m}{t}$, 每个参与方的朴素 hash 表长$b=\beta \cdot n/\log n$, 宽$b_{\max}=\beta_m \cdot \log n$ (β 是β_m 系统固定参数, 具体参数值可参考文献 [18]). 每 t 元素组合需执行 t 次拉格朗日插值, 计算次数为$b \cdot (b_{\max})^t \cdot \binom{m}{t} \cdot t$. 由于$\binom{m}{t} < (m \cdot \mathrm{e}/t)^t$ (e 为欧拉常数) 并且假定$t < \log n, \beta_m \cdot \mathrm{e} < m \cdot (\log n)/t$, 计算次数为$O(b \cdot (b_{\max})^t \cdot \binom{m}{t} \cdot t) < O(\beta \cdot n/\log n \cdot (\beta_m \cdot \log n)^t \cdot (m \cdot \mathrm{e}/t)^t \cdot t) < O(\beta \cdot n \cdot (m \cdot (\log n)/t)^{2t})$. 由于 β 为常数, 最终重构方的计算复杂度为$O(n \cdot (m \cdot (\log n)/t)^{2t})$.

参与方、共享方、重构方理论效率对比如表 1–表 3 所示, Sym 和 PK 分别表示对称密码原语操作和非对称密码原语操作. 本方案相较于目前表现最优的 Mahdavi 等人[18]的方案, 参与方和共享方通信复杂度仅与参与方自身集合大小有关, 与超阈值 t 无关. 参与方、共享方、重构方计算复杂度与对称密码操作有关而非公钥密码操作.

表 1 参与方理论效率对比

协议	通信轮数	通信复杂度	计算复杂度	原语
Kissner[17]	m	$O(nm^3)$	—	公钥密码
MPC[26]	$(\log nm)^2$	$O(nm^3(\log nm)^2+t)$	—	对称密码
Mahdavi[18] (t-PSI0)	2	$O(n)$	$O(n\cdot Pk)$	公钥密码
Mahdavi[18] (t-PSI)	3	$O(tn)$	$O(n\cdot Pk)$	公钥密码
Ours	3	$O(n)$	$O(n\cdot Sym)$	对称密码

表 2 共享方理论效率对比

协议	通信轮数	通信复杂度	计算复杂度	原语
Mahdavi[18] (t-PSI0)	1	$O(nm)$	$O(nm\cdot Pk)$	公钥密码
Mahdavi[18] (t-PSI)	2	$O(nmt)$	$O(nm\cdot Pk)$	公钥密码
Ours	2	$O(nm)$	$O(nm\cdot Sym)$	对称密码

表 3 重构方理论效率对比

协议	通信轮数	通信复杂度	计算复杂度	原语
Mahdavi[18] (t-PSI0)	1	$O(nm)$	$O(Pk\cdot n(m\cdot \log n/t)^{2t})$	公钥密码
Mahdavi[18] (t-PSI)	1	$O(nm)$	$O(Pk\cdot n(m\cdot \log n/t)^{2t})$	公钥密码
Ours	1	$O(nm)$	$O(Sym\cdot n(m\cdot \log n/t)^{2t})$	对称密码

6.2 实验分析

为了直观反映协议效率, 本文实验对比了 Mahdavi 等人[18]的协议, 采用 Mahdavi 等人[18]的环境设置, 编程语言采用 C++, 服务器采用 Ubuntu18.04, 16-core 2.10 GHz Intel(R) Xeon(R) Gold 6130 CPUs, 256 GB RAM. 并采用相同的数学工具 NTL 和 GMP 实现大型算术运算, 相同的并行化处理工具 OpenMP. 实验环境在局域网设置, 假设云和参与方均部署在同一数据中心. 本文采用随机数据集对 Mahdavi 等人[18]的两个方案和本文方案进行对比测试. 参数设置: 小集合 n=10、中集合 n=10^3、大集合 n=10^6, 超阈值 t={2, 3, 4, 5, 6, 7}, 参与方 m=10, 元素大小 2 048 b. 共享生成阶段和重构阶段的运行时间和负载通信量, 如表 4 所示.

表 4 性能评估

超阈值t/参与方m	阶段	协议		运行时间 (s)			通信量 (MB)		
				10^1	10^3	10^6	10^1	10^3	10^6
2/10	共享阶段	MHK20[18]	t-PSI0	**0.221**	16.400	15 200	**0.012**	1.239	1239
			t-PSI	3.582	333.400	335 000	0.101	10.127	10 128
		Ours		0.523	**0.549**	**79.202**	0.021	**0.100**	**78.755**
	重构阶段	MHK20[18]	t-PSI0	1.673	7.021	19.229	0.002	0.102	90.567
			t-PSI	**0.042**	**0.064**	**0.087**	0.002	0.102	90.567
		Ours		**0.041**	**0.064**	**0.087**	0.002	0.102	90.567
	总和	MHK20[18]	t-PSI0	1.894	23.421	15 219	**0.014**	1.241	1 329.567
			t-PSI	3.624	333.464	335 000	0.103	10.229	10 218.567
		Ours		**0.564**	**0.613**	**79.289**	0.023	**0.202**	**169.322**
3/10	共享阶段	MHK20[18]	t-PSI0	**0.221**	15.800	14 800	**0.012**	1.239	1239
			t-PSI	5.074	481.600	508 400	0.132 6	13.263	13 260
		Ours		0.583	**0.635**	**69.038**	0.020	**0.100**	**78.755**
	重构阶段	MHK20[18]	t-PSI0	62.587	541.090	3 013.47	0.002	0.102	90.567
			t-PSI	**0.107**	**0.388**	**1.948**	0.002	0.102	90.567
		Ours		**0.076**	**0.397**	**2.041**	0.002	0.102	90.567
	总和	MHK20[18]	t-PSI0	62.808	541.749	17 813	**0.014**	1.241	1 329.567
			t-PSI	5.181	481.988	508 401	0.132 8	13.497 6	13 350.567
		Ours		**0.659**	**1.032**	**71.079**	0.022	**0.202**	**169.322**

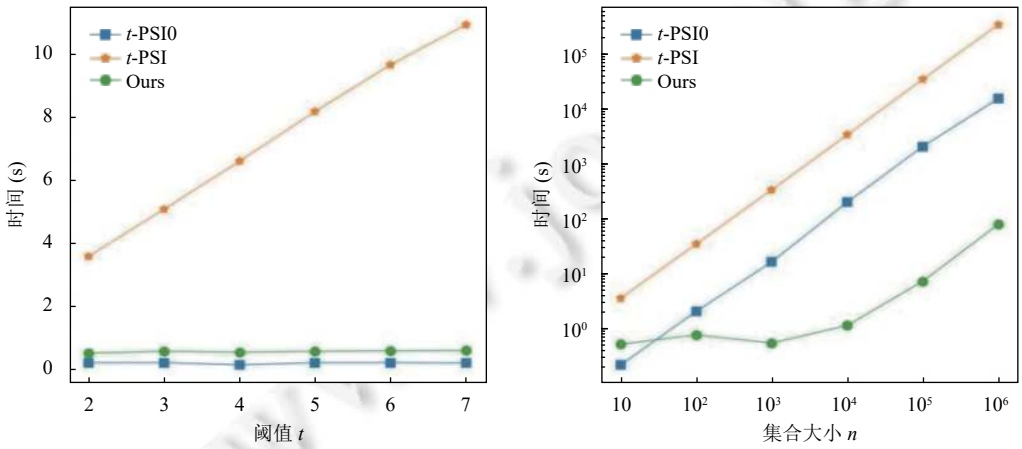
表 4 性能评估 (续)

超阈值t/参与方m	阶段	协议		运行时间 (s)			通信量 (MB)		
				10^1	10^3	10^6	10^1	10^3	10^6
4/10	共享阶段	MHK20[18]	t-PSI0	**0.152**	15.200	14 800	**0.012**	1.239	1239
			t-PSI	6.590	614.400	629 200	0.164	16.395	16 396
		Ours		0.556	**0.656**	**72.305**	0.021	**0.100**	**78.755**
	重构阶段	MHK20[18]	t-PSI0	1 661.38	29 317	—	0.002	0.102	90.567
			t-PSI	**1.204**	**19.584**	**197.497**	0.002	0.102	90.567
		Ours		**1, 123**	**19.209**	**197.288**	0.002	0.102	90.567
	总和	MHK20[18]	t-PSI0	1 661.532	29 333	—	**0.014**	1.241	1 329.567
			t-PSI	7.794	633.984	629 397	0.168	16.661	16 486.567
		Ours		**1.679**	**19.865**	**269.593**	0.023	**0.202**	**169.322**

注: 当表4中数据较大时, 仅保留整数部分. —表示时间非常大, 在实际应用中已意义不大

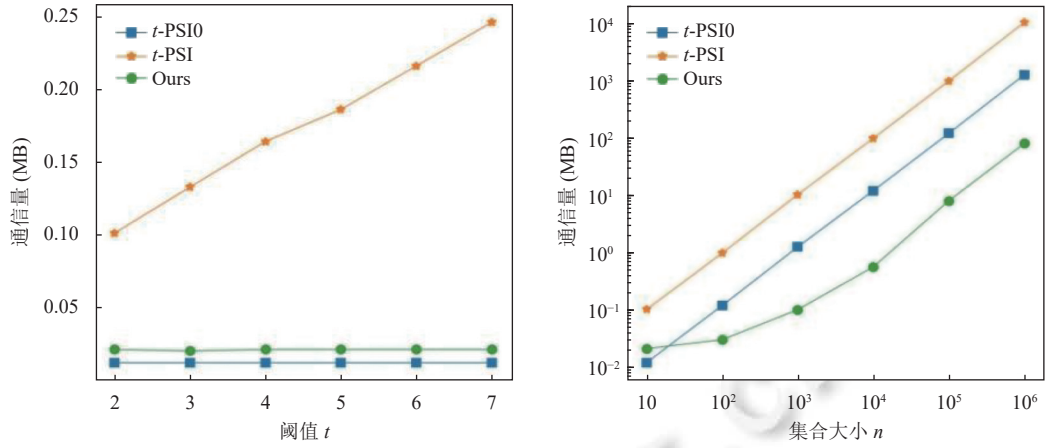
从表 4 中数据可知, 本文方案的运算时间和通信负载具有绝对优势. 目前最优方案[18]在大集合下运算时间是本文方案的 192 倍, 通信负载是本文方案的 7.8 倍. 秘密共享阶段, 当 n>1000 执行 t-PSI0 协议和 t-PSI 协议运算时间过长, 本文测试 t-PSI0 协议和 t-PSI 协议的单个元素秘密共享时间 (通信) 乘以 10 与执行 n=10 的 t-PSI0 协议和 t-PSI 协议秘密共享时间 (通信) 一致. 因此, 本文秘密共享阶段运行时间采用单个元素秘密共享时间乘以集合大小 n 记录, 通信负载采用单个元素秘密共享通信负载乘以集合大小 n 记录. 秘密共享阶段与超阈值 t 关系如图 5(a) 和图 5(c) 所示, t-PSI 协议运行时间和通信负载最高, 运行时间和通信与 t 呈线性增长, 其与 t-PSI 协议需执行同态计算且每一轮传输额外的 t–1 个随机值有关. t-PSI0 协议和本文协议均不与超阈值 t 相关, 本文协议仅使用对称密码相较于公钥具有更快运行时间和低通信负载. 秘密共享阶段与集合大小 n 关系如图 5(b) 和图 5(d) 所示, t-PSI 协议与公钥操作及同态计算呈线性关系, 具有最高的运算时间和通信负载. t-PSI0 协议与公钥操作呈线性关系. 本文协议与对称密码操作呈线性关系, 具有最低运算时间和通信负载.

重构阶段: 本文的重构方式沿袭 t-PSI 和 t-PSI0 的重构协议, 3 种方案均采用同样的重构算法, 仅多项式插值位置有所不同. 分析协议 3 的重构算法, 重构方可并行重构每一行 bin, 因此本文以重构一个 bin 的时间作为整个重构时间. 重构阶段运行时间如图 6(a) 和图 6(b) 所示, t-PSI0 在指数上进行多项式插值, 运行时间随 t 和 n 呈指数增长. t-PSI 与本文协议在常数项上进行多项式插值, 运行时间基本相似. 重构阶段均传输同样大小的朴素哈希表给重构方, 通信负载一样且仅与集合大小相关.



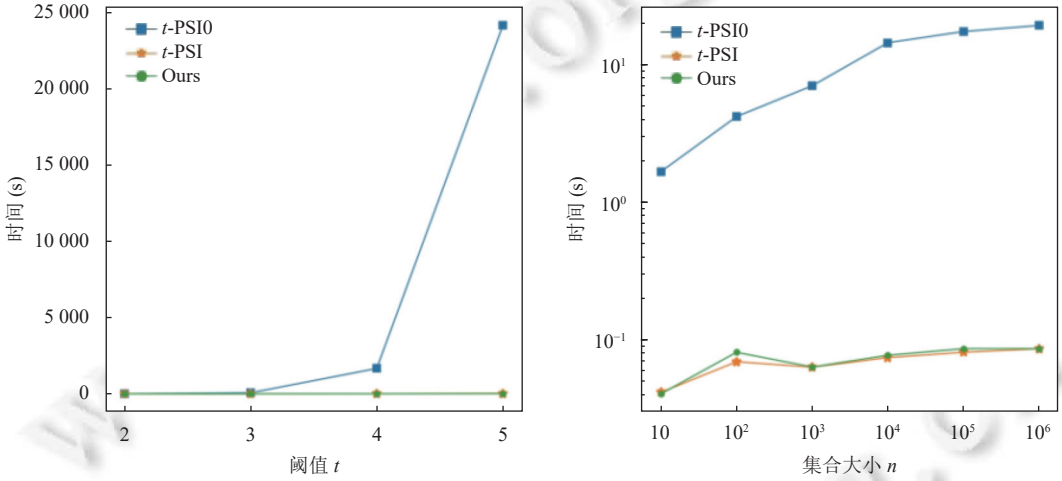
(a) t-秘密共享阶段 (n=10) (b) n-秘密共享阶段 (t=2)

图 5 秘密共享阶段



(c) t-秘密共享阶段 (n=10) (d) n-秘密共享阶段 (t=2)

图 5 秘密共享阶段 (续)



(a) t-秘密重构阶段 (n=10) (b) n-秘密重构阶段 (t=2)

图 6 重构阶段

7 总 结

隐私集合求交是安全多方计算的一种专有协议. 本文针对多方集合下至少存在于 t 个集合中的超阈值多方隐私集合求交问题, 设计了一种具有安全性、可编程性、(t, m) 重构性的密码组件 OPPR-SS 来构建半诚实模型下的云辅助 OT-MP-PSI 协议. 采用 OPRF 和 Shamir 秘密共享构建具有 OPR-SS 功能的 OPPR-SS, 避免了已有 OPR-SS 昂贵的公钥操作及指数上的共享和重构操作, 并借助不可信云服务器完成秘密元素的共享和重构, 解决了弱计算和通信能力的参与方运行协议的问题. 与现有的协议相比, 同等假设条件下, 本文协议 3 类实体的通信复杂度不再与阈值 t 相关, 且具有更快运行时间和更低通信负载. 本文下一步工作将研究更新参与方集合时, 如何保证协议的计算和通信复杂度仅与更新集合大小有关.

References:

[1] Kolesnikov V, Kumaresan R, Rosulek M, Trieu N. Efficient batched oblivious PRF with applications to private set intersection. In: Proc. of the 23rd ACM SIGSAC Conf. on Computer and Communications Security. Vienna: ACM, 2016. 818−829. [doi: 10.1145/2976749.2978381]

[2] Chase M, Miao PH. Private set intersection in the internet setting from lightweight oblivious PRF. In: Proc. of the 40th Annual Int'l Cryptology Conf. on Advances in Cryptology. Santa Barbara: Springer, 2020. 34–63. [doi: 10.1007/978-3-030-56877-1_2]

[3] Dong CY, Chen LQ, Wen ZK. When private set intersection meets big data: An efficient and scalable protocol. In: Proc. of the 20th ACM SIGSAC Conf. on Computer & Communications Security. Berlin: ACM, 2013. 789–800. [doi: 10.1145/2508859.2516701]

[4] Dou JW, Liu XH, Wang WL. Privacy preserving two-party rational set computation. Chinese Journal of Computers, 2020, 43(8): 1397–1413 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2020.01397]

[5] Debnath SK, Dutta R. Towards fair mutual private set intersection with linear complexity. Security and Communication Networks, 2016, 9(11): 1589–1612. [doi: 10.1002/sec.1450]

[6] Mohassel P, Rindal P, Rosulek M. Fast database joins and PSI for secret shared data. In: Proc. of the 27th ACM SIGSAC Conf. on Computer and Communications Security. Virtual Event: ACM, 2020. 1271–1287. [doi: 10.1145/3372297.3423358]

[7] Song XF, Gai M, Zhao SN, Jiang H. Privacy-preserving statistics protocol for set-based computation. Journal of Computer Research and Development, 2020, 57(10): 2221–2231 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2020.20200444]

[8] Wei LF, Liu JH, Zhang L, Wang Q, He CD. Survey of privacy preserving oriented set intersection computation. Journal of Computer Research and Development, 2020, 59(8): 1782–1799 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.20210685]

[9] Demmler D, Rindal P, Rosulek M, Trieu N. PIR-PSI: Scaling private contact discovery. Proc. on Privacy Enhancing Technologies, 2018, 2018(4): 159–178. [doi: 10.1515/popets-2018-0037]

[10] Lv SY, Ye JH, Yin SJ, Cheng XC, Feng C, Liu XY, Li R, Li ZH, Liu ZL, Zhou L. Unbalanced private set intersection cardinality protocol with low communication cost. Future Generation Computer Systems, 2020, 102: 1054–1061. [doi: 10.1016/j.future.2019.09.022]

[11] Duong T, Phan DH, Trieu N. Catalic: Delegated PSI cardinality with applications to contact tracing. In: Proc. of the 26th Int'l Conf. on the Theory and Application of Cryptology and Information Security. Daejeon: Springer, 2020. 870–899. [doi: 10.1007/978-3-030-64840-4_29]

[12] Hazay C, Venkitasubramaniam M. Scalable multi-party private set-intersection. In: Proc. of the 20th IACR Int'l Conf. on Practice and Theory in Public-key Cryptography. Amsterdam: Springer, 2017. 175–203. [doi: 10.1007/978-3-662-54365-8_8]

[13] Kolesnikov V, Matania N, Pinkas B, Rosulek M, Trieu N. Practical multi-party private set intersection from symmetric-key techniques. In: Proc. of the 2017 ACM SIGSAC Conf. on Computer and Communications Security. Dallas: ACM, 2017. 1257–1272. [doi: 10.1145/3133956.3134065]

[14] Kavousi A, Mohajeri J, Salmasizadeh M. Efficient scalable multi-party private set intersection using oblivious PRF. IACR Cryptology ePrint Archive, 2021. https://eprint.iacr.org/2021/484

[15] Badrinarayanan S, Miao PH, Srinivasan R, Rindal P. Multi-party threshold private set intersection with sublinear communication. In: Proc. of the 24th IACR Int'l Conf. on Practice and Theory of Public Key Cryptography. Virtual Event: Springer, 2021. 349–379. [doi: 10.1007/978-3-030-75248-4_13]

[16] Wagner C, Dulaunoy A, Wagener G, Iklody A. MISP: The design and implementation of a collaborative threat intelligence sharing platform. In: Proc. of the 2016 ACM on Workshop on Information Sharing and Collaborative Security. Vienna: ACM, 2016. 49–56. [doi: 10.1145/2994539.2994542]

[17] Kissner L, Song D. Private and Threshold Set-intersection. Pittsburgh: Carnegie Mellon University, 2004.

[18] Mahdavi RA, Humphries T, Kacsmar B, Krastnikov S, Lukas N, Premkumar JA, Shafieinejad M, Oya S, Kerschbaum F, Blass EO. Practical over-threshold multi-party private set intersection. In: Proc. of the 2020 Annual Computer Security Applications Conf. Austin: ACM, 2020. 772–783. [doi: 10.1145/3427228.3427267]

[19] Rosulek M, Trieu N. Compact and malicious private set intersection for small sets. In: Proc. of the 2021 ACM SIGSAC Conf. on Computer and Communications Security. Virtual Event: ACM, 2021. 1166–1181. [doi: 10.1145/3460120.3484778]

[20] Chandran N, Gupta D, Shah A. Circuit-PSI with linear complexity via relaxed batch OPPRF. IACR Cryptology ePrint Archive. 2021. https://eprint.iacr.org/2021/034

[21] Pinkas B, Rosulek M, Trieu N, Yanai A. SpOT-light: Lightweight private set intersection from sparse OT extension. In: Proc. of the 39th Annual Int'l Cryptology Conf. on Advances in Cryptology. Santa Barbara: Springer, 2019. 401–431. [doi: 10.1007/978-3-030-26954-8_13]

[22] Zhou SF, Li SD, Guo YM, Dou JW, Chen ZH. Efficient secure set intersection problem computation. Chinese Journal of Computers, 2018, 41(2): 464–480 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2018.00464]

[23] Pinkas B, Rosulek M, Trieu N, Yanai A. PSI from PaXoS: Fast, malicious private set intersection. In: Proc. of the 39th Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Zagreb: Springer, 2020. 739–767. [doi: 10.1007/978-3-030-45724-2_25]

[24] Pinkas B, Schneider T, Segev G, Zoher M. Phasing: Private set intersection using permutation-based hashing. In: Proc. of the 24th USENIX Conf. Security Symp. Washington: USENIX Association, 2015. 515–530.

[25] Pinkas B, Schneider T, Weinert C, Wieder U. Efficient circuit-based PSI via cuckoo hashing. In: Proc. of the 37th Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Tel Aviv: Springer, 2018. 125–157. [doi: 10.1007/978-3-319-78372-7_5]

[26] Abdelrahaman A, Marcel K, Dragos R, Peter S, Nigel PS, Tim W. SCALE-MAMBA software. 2021. https://homes.esat.kuleuven.be/

~nsmart/SCALE/
[27] Chandran N, Dasgupta N, Gupta D, Obbattu SLB, Sekar S, Shah A. Efficient linear multiparty PSI and extensions to Circuit/Quorum PSI. In: Proc. of the 2021 ACM SIGSAC Conf. on Computer and Communications Security. Virtual Event: ACM, 2021. 1182–1204. [doi: 10.1145/3460120.3484591]
[28] Bay A, Erkin Z, Hoepman JH, Samardjiska S, Vos J. Practical multi-party private set intersection protocols. IEEE Trans. on Information Forensics and Security, 2022, 17: 1–15. [doi: 10.1109/TIFS.2021.3118879]
[29] Evans D, Kolesnikov V, Rosulek M. A pragmatic introduction to secure multi-party computation. Foundations and Trends® in Privacy and Security, 2018, 2(2–3): 70–246. [doi: 10.1561/3300000019]
[30] Garimella G, Pinkas B, Rosulek M, Trieu N, Yanai A. Oblivious key-value stores and amplification for private set intersection. In: Proc. of the 41st Annual Int'l Cryptology Conf. on Advances in Cryptology. Springer, 2021. 395–425. [doi: 10.1007/978-3-030-84245-1_14]
[31] Inbar R, Omri E, Pinkas B. Efficient scalable multiparty private set-intersection via garbled Bloom filters. In: Proc. of the 11th Int'l Conf. on Security and Cryptography for Networks. Amalfi: Springer, 2018. 235–252. [doi: 10.1007/978-3-319-98113-0_13]
[32] Wei LF, Wang Q, Zhang L, Chen CC, Chen YJ, Ning JT. Efficient private set intersection protocols with semi-trusted cloud server aided. Ruan Jian Xue Bao/Journal of Software, 2023, 34(2): 932–944 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6397.htm [doi: 10.13328/j.cnki.jos.006397]
[33] Rindal P, Schoppmann P. VOLE-PSI: Fast OPRF and circuit-PSI from vector-OLE. IACR Cryptology ePrint Archive, 2021. https://eprint.iacr.org/2021/266
[34] De Cristofaro E, Tsudik G. Experimenting with fast private set intersection. In: Proc. of the 5th Int'l Conf. on Trust and Trustworthy Computing. Vienna: Springer, 2012. 55–73. [doi: 10.1007/978-3-642-30921-2_4]
[35] Kolesnikov V, Kumaresan R. Improved OT extension for transferring short secrets. In: Proc. of the 33rd Annual Cryptology Conf. on Advances in Cryptology. Santa Barbara: Springer, 2013. 54–70. [doi: 10.1007/978-3-642-40084-1_4]
[36] Schoppmann P, Gascón A, Reichert L, Raykova M. Distributed vector-OLE: Improved constructions and implementation. In: Proc. of the 2019 ACM SIGSAC Conf. on Computer and Communications Security. London: ACM, 2019. 1055–1072. [doi: 10.1145/3319535.3363228]
[37] Yao AC. Protocols for secure computations. In: Proc. of the 23rd Annual Symp. on Foundations of Computer Science (SFCS 1982). Chicago: IEEE, 1982. 160–164. [doi: 10.1109/SFCS.1982.38]
[38] Shamir A. How to share a secret. Communications of the ACM, 1979, 22(11): 612–613. [doi: 10.1145/359168.359176]

附中文参考文献:

[4] 窦家维, 刘旭红, 王文丽. 有理数域上两方集合的高效保密计算. 计算机学报, 2020, 43(8): 1397–1413. [doi: 10.11897/SP.J.1016.2020.01397]
[7] 宋祥福, 盖敏, 赵圣楠, 蒋瀚. 面向集合计算的隐私保护统计协议. 计算机研究与发展, 2020, 57(10): 2221–2231. [doi: 10.7544/issn1000-1239.2020.20200444]
[8] 魏立斐, 刘纪海, 张蕾, 王勤, 贺崇德. 面向隐私保护的集合交集计算综述. 计算机研究与发展, 2020, 59(8): 1782–1799. [doi: 10.7544/issn1000-1239.20210685]
[22] 周素芳, 李顺东, 郭奕旻, 窦家维, 陈振华. 保密集合相交问题的高效计算. 计算机学报, 2018, 41(2): 464–480. [doi: 10.11897/SP.J.1016.2018.00464]
[32] 魏立斐, 王勤, 张蕾, 陈聪聪, 陈玉娇, 宁建廷. 半可信云服务器辅助的高效隐私交集计算协议. 软件学报, 2023, 34(2): 932–944. http://www.jos.org.cn/1000-9825/6397.htm [doi: 10.13328/j.cnki.jos.006397]

魏立斐(1982－), 男, 博士, 副教授, CCF 杰出会员, 主要研究领域为信息安全, 隐私保护, 密码学.

张蕾(1983－), 女, 博士, 讲师, CCF 专业会员, 主要研究领域为信息安全, 隐私保护, 访问控制.

刘纪海(1998－), 男, 硕士生, CCF 学生会员, 主要研究领域为信息安全, 安全多方计算.

宁建廷(1988－), 男, 博士, CCF 专业会员, 主要研究领域为密码学, 数据安全.