计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239,CN 11-1777/TP

《计算机研究与发展》网络首发论文

题目：面向隐私保护的集合交集计算综述
作者：魏立斐，刘纪海，张蕾，王勤，贺崇德
收稿日期：2021-06-11
网络首发日期：2021-11-18
引用格式：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[J/OL]．计算机研究与发展.
https://kns.cnki.net/kcms/detail/11.1777.TP.20211117.1534.002.html

计 算 机 研 究 与 发 展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.20210685

面向隐私保护的集合交集计算综述

魏立斐　刘纪海　张蕾　王勤　贺崇德

(上海海洋大学信息学院 上海 201306)

(Lfwei@shou.edu.cn)

A Survey of Privacy Preserving Oriented Set Intersection Computation

Wei Lifei, Liu Jihai, Zhang Lei, Wang Qin, and He Chongde

(College of Information Technology, Shanghai Ocean University, Shanghai 201306)

Abstract With the development of Internet of things and big data technology, increasing distributed applications are booming in the personal computers and mobile phones. However, the existing distributed data processing methods have not met the needs of privacy protection. As a typical privacy-preserving technology for distributed set computation, private set intersection (PSI) protocols allow the participants to input individual sets and jointly calculate the intersection of these sets without disclosing any information except the intersection. As an important application of secure multiparty computation, PSI protocols have been widely used in privacy-preserving computation, which has theoretical and practical significance. Many PSI protocols have been emerged, but due to the lack of related surveys on PSI, we have written this paper. This paper introduces the fundamentals of PSI protocols such as the cryptographic technology, adversary model, security proof and implementation framework. And then the paper systematically summarizes the cryptographic framework of traditional PSI protocol from three aspects: the framework based on public key cryptosystem, garbled circuit, and oblivious transfer. Some of the key technologies in the set element comparison are introduced such as oblivious pseudo-random function, oblivious polynomial evaluation, and bloom filter. Furthermore, a few of emerging PSI application scenarios are described in detail such as cloud-based PSI, unbalanced PSI, threshold PSI, and multiparty PSI. Finally, the paper summarizes the problems to be solved and prospects some possible development directions of PSI.

Key words private set intersection; secure multiparty computation; privacy preserving; oblivious transfer; garbled circuit

摘要 随着物联网和大数据技术的发展，在计算机和手机上出现了大量分布式应用程序．然而现有的分布式数据处理方式已不能很好地满足用户对隐私保护的需求．隐私集合交集计算(private set intersection, PSI)协议作为一项典型的面向隐私保护的分布式集合计算技术，允许各参与方输入其私有集合，共同计算集合的交集，且不泄露除交集以外的任何信息．PSI 协议作为安全多方计算的一种重要应用，已被广泛应用于隐私计算领域，具有重要的理论和实践意义．首先介绍 PSI 协议的基本密码技术、敌手模型、安全证明、编程框架等基础知识；其次系统总结了构造传统 PSI 协议的设计框架：基于公钥加密体制的框架、基于混淆电路的框架、基于不经意传输的框架；随后介绍 PSI 协议核心的隐私集合元素比较技术/工具：不经意伪随机函数、不经意多项式评估、布隆过滤器等；进一步地详细阐述了适应新型应用场景的 PSI 方案：基于云辅助的 PSI、非平衡型 PSI、基于阈值的 PSI 和多方 PSI；最后总结并展望面向隐私保护的集合交集计算中亟待解决问题和发展方向．

收稿日期：2021-06-11；**修回日期**：2021-09-24
基金项目：国家自然科学基金项目（61972241）；上海市自然科学基金项目(18ZR1417300)；上海市高可信计算重点实验室开放课题(OP202102)；上海市青年科技英才扬帆计划(21YF1417000)；上海海洋大学骆肇尧大学生科技创新基金项目(A1-2004-20-201312, A1-2004-21-201311)
This work was supported by the National Natural Science Foundation of China (61972241), the Natural Science Foundation of Shanghai (18ZR1417300), the Open Project of Shanghai Key Laboratory of Trustworthy Computing (OP202102), Shanghai Sailing Program (21YF1417000), and the Luo Zhaorao College Student Science and Technology Innovation Fund of Shanghai Ocean University (A1-2004-20-201312, A1-2004-21-201311).
通信作者：张蕾（Lzhang@shou.edu.cn）

关键词 隐私集合求交; 安全多方计算; 隐私保护; 不经意传输; 混淆电路

随着互联网大数据时代的到来, 人们通过对大量分布的数据进行挖掘得到其潜在价值, 从而更好地服务于人们, 如用户爱好推荐系统、广告精准营销等. 然而, 在挖掘数据潜在价值的过程中, 也会产生个人隐私数据泄露等问题, 如英国咨询公司剑桥分析公司在未经 Facebook 用户同意的情况下获取数百万用户的个人数据. 因此, 实现数据可用不可见, 解决数据协同计算和挖掘过程中的数据安全和隐私保护问题就显得迫在眉睫. 相关国家和组织也出台保护隐私数据的法令法规, 如《中华人民共和国密码法》《中华人民共和国数据安全法》《中华人民共和国个人信息保护法》和欧盟《通用数据保护条例》都强调对数据的治理和隐私保护. 数据隐私保护已成为学术界和工业界关注的热点问题.

隐私数据保护最早源于安全多方计算(secure multiparty computation, MPC), 由姚期智[1]借百万富翁问题提出, 指各计算参与方无法得到除计算结果外的任何其他信息, 解决互不信任的数据持有者如何对隐私数据进行计算的问题. 隐私集合交集(private set intersection, PSI)是安全多方计算中的热点问题, 允许在分布式场景下各自持有隐私集合的参与方联合计算出集合交集而不泄露除交集以外的任何隐私信息. 在隐私保护的场景中, PSI 协议具有重要意义, 如新冠接触者追踪[2]、隐私通讯录查找[3]、在线广告实际效果计算[4]、基因序列匹配检测[5]等.

传统的 PSI 协议针对 2 个参与方设计, Meadows[6]基于公钥加密和利用 Diffie-Hellman 密钥交换的乘法同态性质提出了第 1 个 PSI 协议. 随后, 由 Huberman 等人[7]对 Meadows[6]的方案做出了完整描述. 2004 年由 Freedman 等人[8]借助不经意多项式求值和同态加密构造了第 1 个安全 PSI 协议. 2017 年 Shen 等人[9]对安全多方计算框架下的 PSI 协议进行了简要总结. 之后涌现了大量 PSI 的研究成果, 一大批新技术手段和构造框架被提出. 除了传统的安全多方计算理论中的混淆电路(garbled circuit, GC)、不经意传输(oblivious transfer, OT)、秘密共享(secret sharing, SS)、同态加密(homomorphic encryption, HE)等技术外, 不经意伪随机函数(oblivious pseudo-random function, OPRF)、不经意多项式求值(oblivious polynomial evaluation, OPE)、布隆过滤器(Bloom filter, BF)等集合元素比较技术的应用, 使得 PSI 的效率得到了很大的提高.

现有 PSI 已经非常高效, 但现有很多实际应用中仍然以使用高效但存在安全隐患的解决方案为主, 了解现有基于不同密码原语构建的 PSI 及其特定适用场景, 对促进实际场景中使用安全的方案替换存在隐患的方案有很大帮助. 在敌手模型方面, 研究人员从诚实且好奇的安全模型出发, 开始考虑在恶意模型下安全的 PSI 协议. 随着研究人员对隐私集合交集协议的深入研究, 除了传统 2 方 PSI 协议之外, 已衍生出了云辅助 PSI、阈值 PSI(threshold PSI, TPSI)、不平衡 PSI(unbalanced PSI, UPSI)和多方 PSI 新型应用场景.

本文首先介绍 PSI 协议的理论基础、敌手模型、安全证明、实现框架, 其次系统总结了传统 PSI 协议的设计框架, 随后介绍 PSI 协议中的集合元素比较技术, 进一步地详细阐述了适应新型应用场景的 PSI 方案, 最后总结并展望面向隐私保护的集合交集计算中亟待解决问题和发展方向.

1 定义与基础知识

1.1 基础协议

1.1.1 秘密共享

秘密共享技术是许多安全多方计算协议的核心. 如何构建秘密分配算法和秘密重构算法是秘密共享方案的重点. Shamir[10]提出的(k,n)秘密共享方案基于多项式插值构建秘密分配算法和秘密重构算法, 通过控制多项式的未知系数实现阈值门限. Blakley[11]秘密共享方案基于线性几何投影实现. Jia 等人[12]秘密共享方案基于中国剩余定理实现. 除了上述特定环境下的门限秘密共享方案外, 研究者们还设计了安全多方计算环境下的(n,n)秘密共享方案, 可直接在比特碎片上进行相应的计算.

1.1.2 同态加密

同态加密允许对密文进行加或乘的操作, 满足密文计算后的解密值和明文计算的结果相同. 由于整个计算过程是在密文的基础上进行, 故同态加密往往被用做外包计算的隐私保护工具. 同态加密方案由 4 个算法组成: KeyGen 算法生成公钥和私钥、Encrypt 算法加密明文、Decrypt 算法解密密文、Evaluate 算法计算密文. 同态加密根据同态性质可分为加法同态加密(Paillier 加密[13])、乘法同态加密(RSA 加密[14]、ElGamal 加密[15])、全同态加密(FHE[16]). 尽管目前全同态加密算法的效率不足以应用于实际场景, 但其支持直接密文计算的性质使得同态加密仍受到学术界和工业界的重点关注.

1.1.3 不经意传输

不经意传输协议是安全多方计算的基础协议. 标准 1-out-of-2 OT 协议[17]允许发送方输入长度相同的 2 个比特串 m_0, m_1, 接收方输入 $b\in\{0, 1\}$, 结果是接收方输出 m_b. OT 协议保证接收方不知道$m_{\bar{b}}$, 发送方不知道 b.

Crépeau[18]提出随机不经意传输协议(random oblivious transfer, ROT), ROT 协议分为离线-在线阶段, 离线阶段通过少量 OT 使得发送方接收随机消息对(r_0,r_1), 接收方接收随机选择位(b,r_b). 在线阶段双方只需使用随机消息对对真实输入进行盲化而无需进行昂贵的公钥操作. Beaver[19]提出非黑盒子构造, 实现 Yao[1]协议中通过少量公钥操作生成大量 OT 实例. Ishai 等人[20]依据矩阵变化思想实现用少量公钥转化为大量 OT 实例. Kolesnikov 等人[21]对 OT 扩展矩阵进行重复编码实现 1-out-of-n OT. 2013 年 Gilad 等人[22]提出了几种 OT 扩展优化和 OT 相关变体. Boyle 等人[23]基于 Vector-OLE 关联的伪随机发送器和基于 LPN 假设提出 Silent OT 扩展(Boyle 等人[23]、 Schoppmann 等人[24]、Weng 等人[25]、Yang 等人[26]), 离线阶段双方执行相关 OT 生成相关短种子, 本地利用 PCG 将相关短种子无交互的局部扩展为大量相关随机性长源, 在线阶段即可利用长源执行多个 ROT, 大大降低通信复杂度.

1.1.4 混淆电路

混淆电路是 1 种将任意功能函数转化为电路的通用型基础协议. 通过混淆电路表和 OT 加密电线值实现安全隐私保护. 混淆电路的构造思路主要分为 3 种: 第 1 种通过电路混淆者持有两个可能的电线标签, 电路评估者持有标签, 间接实现电线值秘密共享的 Yao[1]协议. 第 2 种电路评估者直接持有电线值的秘密份额的 GMW[27]协议. 第 3 种秘密不再由参与者之间秘密共享而是在电线之间共享, 如基于信息安全论构造协议[28]. 目前混淆电路的研究主要涉及扩展安全模型(如半诚实模型扩展为恶意模型)、减少密文尺寸(如点置换协议[29]、密文混淆协议[30]、免费异或门[31])和降低计算代价.

1.1.5 Hash 技术

Hash 技术是 PSI 协议中优化通信复杂度和计算复杂度的重要工具之一, 本文列举了最常用的 Hash 技术构造方法.

朴素 Hash(plain hash): 使用$hash_k(\cdot)$将元素映射到具有 b 个桶的 Hash 表 T 中的 k 个位置, 每个桶最多有 lb(n)个元素(n 为集合的元素个数). $hash_k(\cdot)$表示 k 个不同的 Hash 函数 $h_1(\cdot)$, $h_2(\cdot)$,...,$h_k(\cdot)$.

布谷鸟 Hash(cuckoo hash)[32]: 使用$hash_k(\cdot)$将元素 e 映射到具有 b 个桶的 Hash 表 T 中的某一个位置, 确保每个桶只能有 1 个元素: 计算 $h_1(e)$, $h_2(e)$, ..., $h_k(e)$, 如果 $T[h_1(e)]$, $T[h_2(e)]$, …, $T[h_k(e)]$至少有 1 个桶为空, 则随机插入; 如果都不为空, 则随机选择 $T[h_i(e)]$, 替换桶中的元素 $T[h_i(e')]$, 再对被替换元素 e'执行上述操作. 当上述替换操作达到一定阈值时, 则将 e'放置在额外的存储空间 stash 中. 因此, 元素 e 必定在以下容器中找到: $T[h_1(e)]$, $T[h_2(e)]$, …, $T[h_k(e)]$或 stash. 由于 stash 可能会存在溢出威胁而导致 Hash 错误, Pinkas 等人[33]通过实验分析出 Hash 函数个数、stash 大小和桶数 b 的最佳关系.

置换 Hash(permutation-based hash)[34]: 将元素转化为更短的字符串并存储在 Hash 表中, 以此减少存储空间和计算复杂度. 元素插入如下: 元素 x 表示为 bit 的形式并拆分为 2 部分 x_1, x_2. 为元素获取 Hash 表的索引: $x_1\oplus H(x_2)$, H 为 Hash 函数. 最后桶中存储大小为$|x_2|=|x|-|x_1|$. $|x|$表示 x 的比特长度. $\oplus$表示按位异或.

1.2 敌手模型

敌手模型由 2 个主要方面构成: 允许敌手的行为方式和腐败策略. 根据敌手是否指示参与方行事可分为半诚实模型、增强半诚实模型、恶意模型. 半诚实模型: 即使被敌手腐败的参与方也会诚实地执行协议, 但在执行过程中会主动收集相关信息, 并试图利用这些信息学习协议中的保密信息. 增强半诚实模型: 在半诚实模型基础上, 允许敌手更改起始输入, 但在其它输入上诚实的执行协议. 恶意模型: 允许敌手控制的参与方根据敌手的指示执行协议, 偏离原本协议.

根据参与方何时处于敌手的控制可分为静态腐败模型、自适应腐败模型、主动安全模型. 静态腐败模型: 敌手控制的参与方固定, 诚实的参与方从协议开始到结束始终是诚实的, 腐败方始终是腐败的. 自适应腐败模型: 在整个协议执行过程中, 敌手可依据需求选择何时腐败和被腐败的参与者. 但腐败后的参与者将一直保持腐败模式到协议结束. 主动安全模型: 参与方只在一段时间内可能被敌手控制, 诚实的一方可能会变得腐败, 腐败的一方也可能变得诚实.

1.3 安全性证明方法

理想/真实模拟范式[35]是安全多方计算协议最常用的 1 种证明方法, 通过模拟具有安全保证的理想模型与现实 PSI 协议比较, 间接证明其安全性. 通过定义理想模型相关的安全目标, 避免协议设计过程中安全目标的不完整性. 理想模型由完全受信任的三方计算功能函数, 并将结果返回给参与方. 真实模

型通过 PSI 协议将功能函数拆分为多个消息函数并在参与方之间相互交流完成计算. 最后理想模型的视图与真实模型的视图达到不可区分来证明 PSI 协议的安全性.

1.4 编程框架

借助 MPC 通用编译器, 改善 PSI 现有技术, 减轻设计自定义协议的负担, 帮助研究人员快速建立协议实验. 本文从支持输入语言、参与方数量、敌手模型和所支持的协议进行简要总结如表 1 所示, 具体详情可参考文献[36].

Table 1 MPC Framework Comparison

表 1 MPC 框架对比

框架	输入语言	参与方数量	敌手	协议
ABY[37]	Custom low-level	2	半诚实	GC, MC
EMP-toolkit[38]	C++ Library	2	半诚实、恶意	GC
Obliv-C[39]	C + extensions	2	半诚实	GC
Sharemind[40]	SecreC	3	半诚实	Hy
PICCO[41]	C + extensions	3+	半诚实	Hy

MC: 多方电路协议；GC: 混淆电路；Hy:混合模型

2 隐私集合交集的设计框架

2.1 基于公钥加密体制的设计框架

隐私集合求交早期思想直接对元素进行加密, 然后在密文上进行相应的比较操作. 其最常用的技术是同态加密, 发送方加密集合发送给接收方. 接收方利用同态加密的性质对密文进行计算, 并将计算结果发给发送方, 发送方利用私钥对其解密并得到集合交集. 基于公钥加密的安全性假设主要分为 3 类: 1) 基于 DH(Diffie-Hellman) 假设. Meadows [6]基于离散对数困难问题实现 DH 密钥交换协议并以此实现 PSI 功能, Huberman 等人[7]发现基于椭圆曲线密码的 PSI 相较于基于离散对数密码的 PSI 具有更高的安全性和高效性. 2) 基于 RSA 假设. Cristofaro 等人[42]基于整数分解困难问题的 RSA 盲签名技术实现半诚实 PSI 协议, 文献[43]分析基于离散对数密码的 PSI 协议比基于整数分解密码的 PSI 协议更加高效. 3) 基于同态加密. Freedman 等人[8]将元素表示为多项式的根, 利用 Paillier 同态加密技术加密多项式系数和零知识证明实现两方恶意攻击安全的 PSI 协议, 2016 年 Freedman 等人[44]使用 ElGamal 加密加快计算效率和布谷鸟 Hash 技术降低计算复杂度. Kissner 等人[45]采用不同的多项式表示方法将计算开销下降到与参与人数呈线性关系. Hazay 等人[46]构建一个具有门限解密的加法同态加密方案实现多方半诚实 PSI 协议 . Abadi 等人[47]提出基于点-值对的 d 次多项式表示集合方法, 通过 Paillier 加密方案完成, 将乘法复杂度从 $O(d^2)$下降到 $O(d)$. Jarecki 等人[48]使用加法同态加密和零知识证明来实现伪随机函数(pseudo-random function, PRF). Dou 等人[49]基于有理数编码和三角形面积计算公式并结合 Paillier 加密实现有理数上的 PSI 协议. 基于公钥加密的 PSI 协议一般具有较小的通信轮数, 适用于具有较强计算能力的模型, 但通信带宽和时间复杂度是实际应用中一个很大的瓶颈障碍.

2.2 基于混淆电路的设计框架

混淆电路可将任意函数转化为布尔电路, 再进行通用的安全计算. 早期基于通用电路的设计方案 DPSZ[50]描述了基于算术电路实现集合求交问题: 电路生成者通过对称密钥对电路门进行加密, 再生成混淆电路并将混淆电路发送给电路评估者; 电路评估者对混淆电路对应线路进行解密得到交集, 且得不到电路中其他线路的任何信息. 其构造的复杂度随电路深度的增加而增加. 本文主要讨论专用的电路 PSI 协议, 即在预处理阶段减少电路的比较次数和电路的深度, 电路阶段只进行元素的相等性测试以实现通用的 PSI 协议, 并可在电路 PSI 协议的基础上执行对称函数(交集基数、交集和、阈值-交集). 混淆电路有两种抵抗半诚实敌手的电路 Yao[1]协议和 GMW[27]协议. Huang 等人[51]对元素进行特定排序, 通过 Yao 电路合并后进行相邻元素的相等性测试, 构造出排序比较乱序电路实现半诚实安全的 PSI 协议. Pinkas 等人[52-54]和 Chandran 等人[55]基于 GMW 电路和 Hash 存储结构进行隐私集合的成员测试构造

出 OPRF 电路实现半诚实安全 PSI 协议. 上述方案通过 Hash 技术降低比较次数, 通过隐私成员测试协议降低电路等值比较的深度, 使得电路 PSI 越来越高效. 然而, 此类协议需要额外的密钥计算过程和通信, 如参与方需要密钥协商等.

2.3 基于不经意传输的设计框架

基于不经意传输技术构造 PSI 协议通过随机值盲化集合元素产生隐私保护效果. 构造思想: 首先将集合元素通过特定数据结构存储, 然后双方为每一个桶运行 OT 协议: 发送方使用随机值盲化集合元素并将盲化结果发送给接收方, 接收方在本地执行相等性测试得到隐私集合交集. 由于需要使用大量的 OT, 传统 OT 协议限制了 PSI 协议的安全性和效率性, 通过 OT 扩展技术(oblivious transfer extension, OTE)可有效解决该瓶颈. OTE 依据设计思想可分为 2 类: 一类是基于矩阵变换的思想利用少量公钥 OT 实现大量 OT 实例的 IKNP-OT. 依据抵御敌手行为能力可分为半诚实安全模型 OT 协议[20]和恶意安全模型 OT 协议[56] . 文献[58]基于布隆过滤器和 IKNP03-OT[20]构造出半诚实 PSI 协议. Pinkas 等人[43]将文献[58]中 OT 协议换为 ALSZ13-OT[22]使接收方到发送方的通信量减少一半. 文献[59-60]将文献[58]中 OT 协议换为 KSO15-OT[56]并结合 Cut-and-Choose 技术, 以确保 Dong 协议中的发送方输入不能明显多于其 BF 中 1 的数量, 实现恶意攻击安全的 PSI 协议. 文献[61]对文献[59]的 k-out-of-n OT 参数进行改进提供了更好的安全保证. Kolesnikov 等人[62-63]、Pinkas 等人[64]、Chase 等人[65]分别基于 1-out-of-n KK13-OT[21]和伪随机函数构造出单点 OPRF、不经意可编程伪随机函数(oblivious programmable pseudorandom function, OPPRF)、多点 OPRF(multi-point OPRF, mOPRF)、带权多点 OPRF 实现具有半诚实安全的 PSI 协议. Pinkas 等人[66]基于 OOS17-OT 协议[57]构造具有恶意安全的 PSI 协议. 另一类是基于子向量不经意线性评估实现具有更低的通信效率但计算复杂度增加的 Silent-OT. Rindal 等人[67]基于半诚实安全的 Schoppmann 等人[24]协议和恶意安全的 Weng 等人[25]协议分别构造出半诚实安全和恶意安全的 PSI 协议. 基于 OT 的 PSI 协议一般具有较低通信和计算量, 本文依据设计思想、功能和安全模型对现有 PSI 协议进行分类如表 2 所示:

Table 2 PSI Protocols Based on OT Framework

表 2 基于 OT 设计框架的 PSI 协议

设计思想	功能	安全模型	OT 协议	PSI 协议
IKNP -OT	1-out-of-2 OT	半诚实模型	IKNP03-OT[20]	文献[58, 84-85]
			ALSZ13-OT[22]	文献[43]
		恶意模型	KOS15-OT[56]	文献[59-61]
	1-out-of-n OT	半诚实模型	KK13-OT[21]	文献[62-65, 74]
		恶意模型	OOS17-OT[57]	文献[66]
Slient-OT	1-out-of-2 OT	半诚实模型	SGRR19-OT[24]	文献[67]
		恶意模型	WYKW20-OT[25]	文献[67]

3 隐私集合元素比较技术/工具

3.1 不经意伪随机函数

利用 OPRF 设计 PSI 协议是一种常见的思想. OPRF 允许接收方输入x, 发送方输入密钥 k, 接收方输出$F_k(x)$, 发送方无任何输出. 然后发送方本地计算$F_k(y)$并将其发送给接收方. 接收方通过比较$F_k(x)$和$F_k(y)$可以构造出 PSI. 基于 OPRF 构造 PSI 的论文进展如图 1 所示:

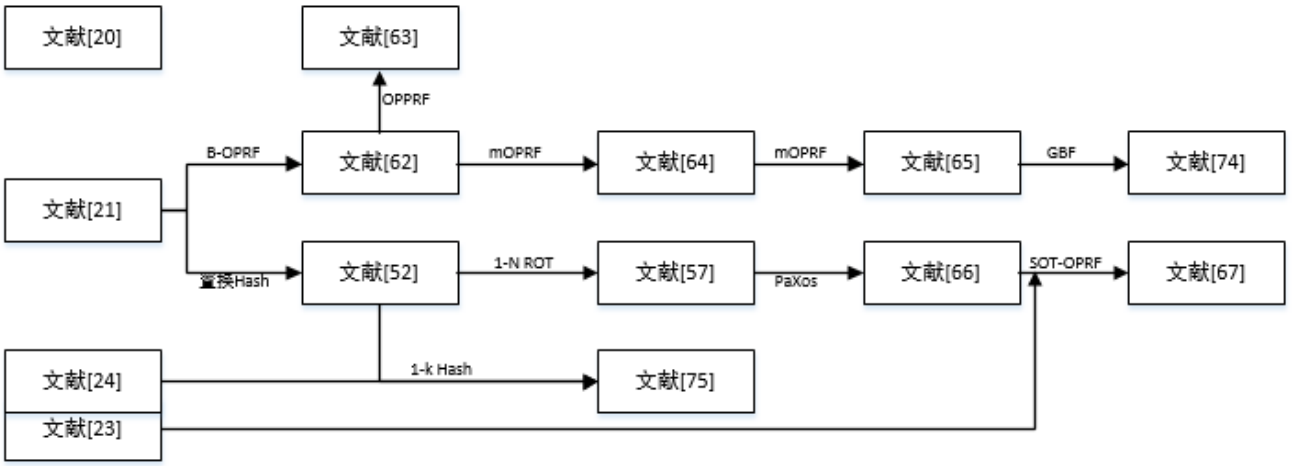

Fig. 1 The progress of OPRF based PSI protocols

图 1 基于 OPRF 的 PSI 论文进展

Naor 等人[68]基于 Diffie-Hellman 假设和标准 1-out-of-2 OT 构造出第 1 个 OPRF, Hazay 等人[69]利用上述 OPRF 实现安全的 PSI 协议, 满足单边恶意敌手模型, 但其使用的是非单向映射的 PRF, 可能会因恶意选择 PRF 密钥而产生冲突. Jarecki 等人[48]基于 Dodis-Yampolskiy[70]提出的具有$O(1)$指数幂的单射伪随机函数: $f_k(x)=g^{1/(k+x)}$和 Paillier 同态加密完成 OPRF 构建. 协议需要多个模N^2的指数运算, 导致计算量非常大. Debnath 等人[71]通过零知识证明技术和引入半可信第三方, 构造出双方获取交集结果的 OPRF. Jarecki 等人[72]基于不可预测函数: $f_k(x)=(H(x)^k)$ 和 Hash 函数完成 OPRF: $f_k(x)=H'(H(x),H(x)^k)$构建. 构造 PSI 协议如下: P_2 使用随机值a盲化$H(x)$得到$y=H(x)^a$并将其发送给 P_1. P_1 使用密钥k加密y得到$z=y^k$并将其发送给 P_2. P_2 即可得到$f_k(x)=z/a=H(x)^k$. 通过外层 Hash 函数和对密钥 k 进行零知识证明以抵抗恶意敌手的攻击.

Kolesnikov 等人[62]基于随机 OT[20]协议改进了 1-out-of-2 OT 扩展并结合对称密钥和位运算构造出单点 OPRF: $f_k(x)=H(q\oplus[C(x)\wedge s])$(其中密钥$k$由$q,s\in\{0,1\}^n$组成, $\wedge$表示按比特位与操作): 双方首先通过布谷鸟$hash_k$将集合映射到布谷鸟 Hash 表中, 以降低比较次数. 再对每一个桶执行单点 OPRF 操作, P_2 随机均匀采样字符串$r_0\in\{0,1\}^n$计算$r_1=r_0\oplus C(y)$, P_1 随机采样字符串$s\in\{0,1\}^n$. 双方分别输入 r_1 和 s 执行 n 轮 OT, 最后 P_1 接收 n 位$\{r_{s[j]}[i]\}_{i\in n,s[j]\in\{0,1\}}$. P_1 计算$q=r_{s[1]}[1]||r_{s[2]}[2]||\ldots||r_{s[n]}[n]$得到 OPRF 密钥$k=(q,s)$. P_2输入y到 OPRF 输出值$H(r_0)$. 如果$x=y$, 则$q\oplus[C(x)\wedge s]=r_0$, 即得到交集.

Kolesnikov 等人[63]基于零共享和 OPPRF 实现多方隐私集合求交协议. OPPRF 是 OPRF 协议的扩展, 通过在 OPRF 的安全属性上附加特定输入产生特定输出属性, 将两方协议扩展为适用的多方协议. OPPRF 允许密钥持有者编程 F, 它可以控制特定值具有特定的输出, 其他值都是伪随机输出. 评估 OPPRF 的另一方不知道获得的是特定输出还是伪随机值. 该文设计了 3 种构造 OPPRF 的方法并通过实验对比: 基于多项式插值的 OPPRF 其具有最小的通信, 但多项式插值计算复杂度为$O(n^2)$. 基于布隆过滤器的 OPPRF, 插入算法只需要$O(n)$,通信仍然是$O(n)$, 但其常数系数很高. 基于表的 OPPRF 具有最优的通信开销和计算成本. 为实现多方 PSI 协议首先各方安全的生成零共享: $\sum_{i=1}^{n}s_j^i$=0. 参与方两两执行 OPPRF 协议, 如果接收方与输入方具有相同元素, 则接收方输出特定共享值. 各参与方对收到的共享值进行本地重组得到 S'. 指定方在重构阶段为保证参与方信息不被泄露, 指定方 P_1 需与每个 P_i 再次执行 OPPRF 协议输出 S', 若x_j在交集中, 则重构结果为 0.

由于文献[62]中每个元素被多个 Hash 函数映射, 从而导致每个元素都需执行多个单点 OPRF 操作. Pinkas 等人[64]通过多项式插值技术结合 IKNP-OT 设计了 1 个稀疏 OT 扩展, 其允许接收方从 n 个随机秘密中不经意的选取 k 个以此实现多点 OPRF(multi-point OPRF, mOPRF), 即实现每一个元素只需要 1 个 OPRF 操作. 该协议通过选择与稀疏 OT 吻合的 Hash 结构: 2-选择 Hash[73], 不需要伪随机填充值, 且每个桶可放入多个元素, 使得比较次数下降. 但是与文献[62]相比, mOPRF 需要在 1 个大的域上计算和插值 1 个高阶多项式从而产生较高的计算开销, 比文献[62]仅使用对称密码和位运算要花费更多的成本. Chase

等人[65]仅利用 OT、Hash 函数、对称密码和位运算构建 mOPRF, 相比于文献[64]基于多项式插值的 OPRF 更有效率, 通过分析文献[62]的协议构造可知无论发送方选择不同 s 所得到的密钥 k, 都有$f_k(y)=H(r_0)$, 基于这一发现构建了 1 个新的 mOPRF: 新的 OPRF 密钥包含一个大小为$m\times w$的矩阵 M. 发送方选择随机字符串 $s\in\{0,1\}^w$, 接收方准备 2 组列向量$A_1,A_2,\ldots,A_w\in\{0,1\}^m$, $B_1,B_2,\ldots,B_w\in\{0,1\}^m$. 2 方执行 w 个 OT,发送方作为接收者得到 w 个列向量, 由此得到 OPRF 密钥 M. 接收方输入元素 y 得到$f_M(y)$. 发送方计算自身集合的$f_M(\cdot)$发送给接收方即可实现 PSI 协议.

Kavousi 等人[74]采用星型-路径(star-path)网络结构并结合文献[65]的 mOPRF 构造出半诚实多方 PSI 协议. star 模块用于指定方 P_t作为发送方与其他所有参与者执行 OT, 使得 $P_j(j\in[1,t-1])$秘密共享P_t的矩阵并得到 1 个随机字符串的列向量矩阵. path 模块用于重构秘密, 即每一个参与方仅向最后参与方的方向的相邻参与方发送混淆布隆过滤器(garbled bloom filter, GBF), 一直持续到 $P_{t\text{-}1}$, $P_{t\text{-}1}$ 计算 GBF 和 OPRF 值, 并将 OPRF 值发送给指定方 P_t, P_t 通过比较 OPRF 值得到交集. 这种设计使得每一方(指定方除外)的通信和计算复杂度仅取决于其自己的输入集大小, 而不取决于协议中涉及的参与方数.

Falk 等人[75]提出基于 1-out-of-k Hash 和 Silent-OT[23]构建 OPRF 实例. Silent-OT 相较于 IKNP-OT 具有通信量显著下降的特点. 使用 Silent-OT 协议和 1-out-of-k Hash 实现离线预处理阶段计算其元素的最优分配. 1-out-of-k Hash 通过数组 A 表示集合 S, 对于$hash_k$查找 x 只需检查是否$A[h_i(x)]=x$, 其具有高效的空间利用率和恒定的查找时间. 当双方不需要动态插入元素时, 多项选择 Hash 比布谷鸟 Hash 具有更好的性能, 即可以在本地计算最优分配.

Pinkas 等人[66]通过改进布谷鸟 Hash 算法(probe-and-XOR, PaXoS)并结合 OOS-OT[57]实现了第 1 个基于布谷鸟 Hash 的恶意安全 PSI 协议. PaXoS 是 1 个将 n 个二进制字符串映射到 m 个二进制字符串的随机函数, 由 Hash 映射到对应插槽值异或得到集合元素, 以消除布谷鸟 Hash 构建恶意 PSI 时泄露发送方未在集合交集中的集合信息问题. 同时 PaXoS 具有与 GBF 相同的渐进编码和解码, 但计算速率却比 GBF 快. 该文主要通过 OOS 协议的同态性质构建 PSI 协议.

Rindal 等人[67]基于 Vector-OLE[24]和 PaXoS[66]数据结构提出批量化的 OPPRF (batch-OPPRF, B-OPPRF)新构造. 基于 Vector-OLE[24]的 Silent-OT 构造新的 OPRF, 具有$O(n)$的通信量和计算量非常高效, 并且以很小的开销可实现恶意安全性. 基于 PaXoS 数据结构实现了新型可编程 PRF, PaXoS 具有编解码高效, 其只需 $O(1)$时间复杂度计算.

3.2 不经意多项式评估

Freedman 等人[8]将元素比较问题转化为多项式求根问题, 通过乘法同态加密性质实现 PSI. 参与方 P_1 和 P_2 分别拥有集合 X_1 和 X_2. 首先 P_2 利用具有乘法同态属性的方案生成密钥对(PK,SK), 将集合 X_2 的元素作为多项式的根构建$|X_2|$阶多项式$Q(\cdot)$, 加密多项式系数发送给 P_1. P_1 对集合 X_1 中的元素打乱并选择 1 个随机值 r_j, 利用同态加密方案计算$r_j\cdot Q(x_j)+x_j$并将其发送给 P_2. P_2 解密, 如果 z 属于$X_1\cap X_2$, 则对任意的 r 都有$r\cdot Q(z)+z=r\cdot 0+z=z$. 否则$r.Q(z)+z$是 1 个随机值. 如果多项式的阶数较大, 将导致同态加密的指数计算成本较高.

Kissner 等人[45]提出了 1 种具有 2 次计算的集合多项式表示, 即不局限其只能在交集上操作. 设 f,g 分别为集合 S 和 T 的多项式表示. r 和 s 分别是 f,g 多项式环上的随机多项式. 依据多项式的数学性质: $f\cdot r+g\cdot s$是$S\cap T$的多项式表示. Cheng 等人[76]、Zhou 等人[77]充分利用该性质并结合数学困难问题实现 PSI 协议，避免了文献[45]中繁琐的公钥操作. Hazay 等人[46]利用星型拓扑网络结构和文献[8]构造了 1 个半诚实多方隐私交集协议. 通过加法同态加密方案为参与方生成持有同一公钥 PK 而私钥 $\{SK_1, SK_2,\ldots,SK_n\}$各不相同的密钥对, 其具有门限解密特点. 执行星型拓扑交互, P_1 单独与每一方执行 Π[8]: P_1 计算 $r_j\cdot Q(x_1^j)$并保留, 参与方 P_i 通过 SK_i 加密集合得到密集$(C_1^i,C_2^i,\ldots,Cm_i^i)$, m_i 表示 P_i 的集合大小, 并将其发送给 P_1. P_1 对所有密集组合得到 $C_1=\sum_{i=2}^{n}C^i_{m_{\max}}(m_{\max}=\{m_1,m_2,\ldots,m_n\})$, 并将其表示为多项式$Q_1=Q_2(\cdot)+Q_3(\cdot)+\ldots+Q_n(\cdot)$. 最后通过门限解密协议参与方共同解密$C_1^j=r_1\cdot Q_1(x_1^j)$, 仅当$C_1^j$为 0 时, x_1^j是交集元素. 以上将集合表示为 d-多项式系数, 当 2 个多项式相乘时复杂度为$O(d^2)$, 且多项式乘法需在乘法同态加密下完成, 使得计算更加复杂.

Abadi 等人[47]提出基于点-值多项式表示集合的方法. d 次多项式$Q(\cdot)$可表示为 1 组 $n(n>d)$个点值对$\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$的集合, 其中 x_i 不相等且$y_i=p(x_i)$. 点值多项式表示可通过位加法或乘法完成, 将乘法复杂度降低到$O(d)$. 当求根时, 点值多项式可通过多项式插值转化为多项式系数. Ruan 等人[78]通过盲化多项式的点-值对和伪随机函数计算多项式的根, 实现无加密技术的高效安全 PSI 协议, 通过置换 Hash[34]将元素转化为较短的字符串, 以减少存储空间和计算复杂度, 使每一方将其原始集合分解为几个小子集合以此减少多项式的阶数, 提高协

议的效率. 双方各自将集合作为根构造 d 次多项式 $p(x)=\prod_{i=1}^{d}(x-s_i)$, 利用伪随机函数生成盲化随机值 $r=\{r_1, r_2,...,r_d\}$构建多项式$r(x)=\prod_{i=1}^{d}(z-r_i)$. P_1 将 $p(x)$和 $r(x)$表示为 $n(n>d)$个点值对并计算其和发送给 P_2, P_2 通过插值获得多项式, 通过计算多项式的根来获得交集. 该协议要求双方必须具有相同集合大小.

3.3 混淆电路

混淆电路可将任意函数转化为布尔电路, 故集合求交也可由电路构造. 基于电路的 PSI 协议通常效率低下, 但仍是研究的热点, 因其具有通用性, 不必更改主电路只需添加子电路, 即可实现基于求交的函数, 比如交集基数等. 基于电路的隐私集合求交最朴素的想法是利用与门对大小为 n 的 2 个集合进行 n^2次比较, 但其完全由电路构造产生了很高的通信复杂度. 电路计算的通信量由比较次数、元素大小以及电路安全参数决定. 然而元素大小和电路安全参数都是给定的, 所以设计计算交集的电路困难之处在于决定哪些元素需要进行比较. 基于混淆电路构造 PSI 的论文进展如图 2 所示:

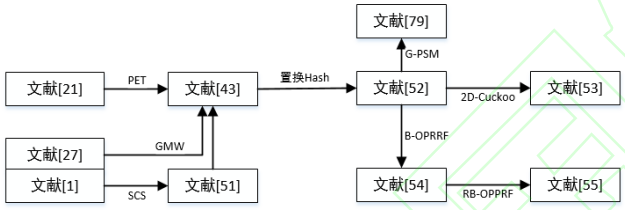

Fig. 2 The progress of GC based PSI protocols

图 2 基于 GC 的 PSI 论文进展

Huang 等人[51]提出了 3 种基于混淆电路的相等性测试(private equality test, PET)以实现隐私求交功能. 第 1 种位与协议: 将集合通过二进制向量表示, 然后通过与门进行逐位与操作即可完成相等性测试. 第 2 种是元素直接比较. 第 3 种排序比较乱序协议(sort compare shuffle, SCS), 双方以特定方式本地排序集合输入, 通过不经意合并排序网络计算 2 个集合的并集排序列表. 然后通过混淆电路进行相邻元素比较得到集合匹配列表, 其比较次数为$O(nlogn)$. 最后通过乱序网络对其进行重新排序以保证匹配元素位置信息不被泄露.

Pinkas 等人[43]使用 GMW 协议优化了 SCS 电路协议, 通过减少 OT 次数对协议的通信进行优化. 随后他们[52]提出电路阶段化的思想: 通过置换 Hash 技术减少元素的存储大小和降低比较次数, 再应用电路进行隐私成员测试(private set membership, PSM)评估. 具体协议如下: 参与方 P_0 使用布谷 Hash 算法构建布谷 Hash 表 T_0, P_1 构造朴素 Hash 表 T_1. 由 Hash 性质可知, P_0 和 P_1 集合中的相同元素一定在 2 个 Hash 表的同一行. 再通过电路逐行进行隐私成员测试, 由于各方须隐藏映射到桶中的元素个数, 因此剩余位置需填充虚拟值, 由此产生不必要的比较. 该协议将比较次数从$O(n^2)$下降到$O(nlogn/loglogn)$(n为集合大小, $logn$由布谷 Hash 表性质决定)且使电路的深度不再和集合大小相关.

Pinkas 等人[53]设计了 1 种 2 维布谷鸟 Hash, 将比较次数下降到$O(n)$使得布尔电路计算交集的开销只需$\omega(n)$. 2 维布谷鸟 Hash 结构由 2 个表 *TL*, *TR* 和 4 个公共 Hash 函数 HL_0, HL_1, HR_0, HR_1 组成. P_2 使用布谷 Hash 算法插入到表 *TR* 中. P_1 通过 HL_0, HL_1 将元素插入 *TL*, 或通过 HR_0, HR_1 将元素插入 *TR*, 通过改进的布谷鸟插入算法实现. 即 P_2 将其每个元素映射到每个表中的 1 个子表. P_1 将其每个元素映射到其中 1 个表的 2 个子表. 确保 P_1 和 P_2 的交集元素被映射到同一个子表中. 最后共同构建 1 个电路, 对每个桶中双方存储的元素进行比较得到交集.

Ciampi 等人[79]基于文献[52]的阶段化思想, 进一步设计了一个结果秘密共享的 PSM 协议, 最后电路只需要做相等性测试, 以此减少电路尺寸. 该协议基于不经意图跟踪构建隐私成员测试协议如下: 发送方构造 1 个二叉树, 每个节点包含 1 个对称密钥. 发送方输入二叉树, 接收方输入测试元素, 双方执行 1-out-of-2 OT. 其允许接收方不经意的遍历树, 将成员测试结果秘密共享. 最后通过电路等值比较得到交集.

Pinkas 等人[54]设计了 1 个 B-OPPRF 协议以完成隐私成员测试, 将结果输入电路执行相等性测试. 通过布谷鸟 Hash 和朴素 Hash 将隐私集合求交问题简化为 h 个隐私成员测试问题. 当调用 b 个 OPPRF 实例时, 桶中隐私元素个数不一致导致每个桶的编码数量不同, 但是其总的编码数量是固定的, 故设计了 1 个提供大小为 N 并进一步隐藏每个桶中编码点数量的原语称为 B-OPPRF. 通过 B-OPPRF 协议完成隐私成员测试: P_1 作为发送者, 在 1 个包含 b 个 OPPRF 独立实例的 B-OPPRF 实例中, 在第 j 个 OPPRF 实例中将所有编程输出设置为单个随机值 t_j. 然后 P_0 评估第 j 个 OPPRF 的 $T_0[j]$, 如果 $T_0[j]$等于 $T_1[j]$则 P_0 和 P_1 持有相等值. 最后通过电路比较 $F_k(x)$, 因此该电路只需要对每个桶进行 1 次比较计算. 通过 B-OPPRF 将桶的比较次数从$O(\log n)$减少到 1, 但 OPPRF 在创建提示时使用多项式插值, 导致计算复杂度增加. Chandran 等人[55]基于上述 B-OPPRF 设计了 1 个严格的 RB-OPPRF. 通过使用 1 个具有 3 个 Hash 函数的布谷鸟 Hash 结构取代多项式插值结构完成 B-OPPRF 操作, 将每个桶的比较次数从$O(\log n)$减少到 3, 同时只产生线性计算开销.

Mohassel 等人[80]、Song 等人[81]不使用电路构造情况下也实现了交集结果上执行对称函数(交集基数、交集和)且不泄露交集信息的电路 PSI. Mohassel 等人[80]将协议设定在三方之间, 通过秘密共享和不经意交换网络实现输入数据和输出数据秘密共享. P_2 构造布谷鸟 Hash 表 T, 利用 P_1 的元素和$hash_k$将表 T 映射为 k 个 1 维表, 其间通过不经意交换网络保护数据隐私, 将每个桶的比较次数下降到 1. Song 等人[81]采用布谷鸟 Hash 和朴素 Hash 降低比较次数, 双方执行 OPRF 操作: P_2 输入 y 输出$f_k(y)$, P_1 输出密钥 k. P_1 本地生成$f_k(x_i)$和随机值 r 构造盲化多项式$P(x) = r + \prod_{i=1}^{|X|}(x - f_k(x_i))$, 以实现电路 PSI 的构造, 最后将等值比较问题转化为比特串汉明距离问题, 实现 OT 构造等值比较协议, 相较于 GMW 电路等值比较具有更小的通信量.

3.4 布隆过滤器

布隆过滤器[82]是一种对集合进行编码的数据结构, 其利用若干独立分布的 Hash 函数将集合中的每个元素映射到二进制数组中最终得到 1 个轻量级 1 维数组, 是有效进行集合相等性测试的工具. 未保护隐私的 BF 构造 PSI 思想如下: 参与方 P_1 和 P_2 分别拥有集合 S_1 和 S_2, 参与方 P_1 依据$hash_k$构造 1 维数组 bf_1, 将 bf_1 的所有位设置为 0, 对每个$x \in S_1$, 计算 $hash_k(x)$作为 bf_1 的索引并将对应位设置为 1. P_2 为元素 y 进行相等性测试, 即$hash_k$ (y)对应的 bf_1 均为 1, 则 y 属于 S_1. BF 存在一定的错误率, 其与 Hash 个数、BF 长度、集合大小有关. 文献[83]对其进行详细的分析并给出最佳 BF 构造参数. 基于布隆过滤器构造 PSI 的论文进展如图 3 所示:

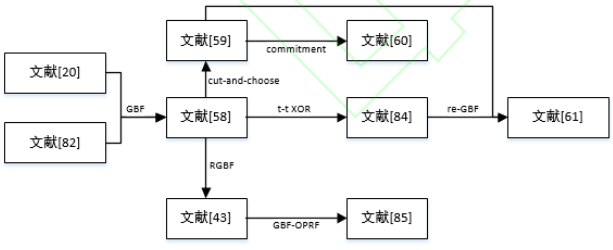

Fig. 3 The progress of BF based PSI protocols

图 3 基于 BF 的 PSI 论文进展

Dong 等人[58]首先通过 BF 构建 PSI: 双方根据其私有集构造 BF. 双方执行 OT 协议, P_2 获得 P_1 中 BF 为 1 的索引. 最后 P_2 通过计算 2 个 BF 上的逐位与操作, 得到交集结果. 但其泄露了 P_1 的 BF 中为 1 但不属于交集的额外信息. 由此进而提出混淆布隆过滤器 GBF, 将元素 x 首先拆分为 k 个共享值并通过$hash_k$映射得到 k 个 BF 索引, 并将 k 个共享值随机填充, 将 BF 中的每 1 比特信息改为 1 个随机字符串. GBF 插入元素 x 步骤: 计算$hash_k$, 共用 GBF 中已有字符串, 剩下的位置通过异或得到共享字符串并填充入对应位置. 基于 GBF 的隐私集合交集: P_2 根据私有集构造 BF, P_1 根据私有集构造gbf_1和随机填充gbf_2. P_2 测试元素 x 是否属于 P_1 的私有集合执行如下: 双方执行 m 个 1-out-of-2 OT(m 为 BF 长度), P_1 输入消息对(gbf_1,gbf_2), P_2 输入 BF 作为选择位, P_2 只能获得到 $gbf[h_i(x)]$, $(i \in [1,k])$. 然后 P_2 将 k 个 $gbf[h_i(x)]$进行重组与 x 比较即可完成测试.

Rindal 等人[59]通过生成比所需的 BF 比特数略多的 1-out-of-2 OT 构建 Cut-and-Choose 技术以此实现抵抗恶意敌手的 PSI 协议: P_1 从 OT 中随机抽取一小部分, 让 P_2 证明适当数量的 OT 中使用了选择位 0. 然后再进入在线阶段, 执行离线阶段未使用的 OT 协议.

Pinkas 等人[43]基于随机 OT 扩展设计了 1 个不经意伪随机发生器(oblivious pseudo random generator, OPRG)以完成 GBF 的构建: 双方基于私有集构建 BF, 分别为 bf_x 和 bf_y, 将 bf_x 和 bf_y 的每一个对应比特 b_x,b_y 作为 OPRG 的输入. OPRG 产生 1 个随机字符串发送给$b_t = 1(t \in \{x,y\})$的参与方构建出 gbf_x 和 gbf_y. 对集合 X 的每一个元素 x_i, P_1 计算 $m_1[i] = gbf_x[h_1(x_i)] \oplus gbf_x[h_2(x_i)] \oplus \ldots \oplus gbf_x[h_k(x_i)]$. P_1 将 m_1 打乱并发送给 P_2, P_2 通过检查是否存在 1 个 i 使得 $m_1[i] = gbf_y[h_1(y_i)] \oplus gbf_y[h_2(x_i)] \oplus \ldots \oplus gbf_y[h_k(y_i)]$判断元素 y_i 是否为交集. 相比于 GBF, OPRG 具有更小的通信量和计算量.

Inbar 等人[84]通过秘密共享将文献[58]扩展为多方 PSI 协议: 每个参与方 P_i 通过$hash_k$本地生成 GBF 和 BF. 然后参与方 P_i 选择 t 个随机字符串共享 GBF, 将字符串分发给其他参与方, 以完成在所有各方之间(t, t) 异或秘密共享其 GBF. P_i 将其接收到的所有 GBF 的共享字符串异或后, 得到新的 GBF 再将其发送给 P_0. P_0 对所有的 GBF 进行异或再和本地 BF 异或得到交集. Zhang 等人[60]使用星型拓扑结构[46]和 Cut-and-Choose[59]技术实现多方恶意安全的 PSI. 协议通过离线-在线阶段的方式进一步优化, 将多数繁重的计算、通信放在预计算阶段.

Karakoç 等人[85]继承文献[54]的设计思想, 通过布谷鸟 Hash 算法构建 2 维表, 再对每个桶运行 PSM 协议, 执行比较协议并完成具有可计算对称函数的电路 PSI. 文献[84]通过修改文献[58]的结构, 设计了基于 BF 的 PSM: 显著降低了通信量, 且不再使用电路进行等值比较, 而是将 PSM 通过控制参与方只输入 1 个元素得到 PET, 执行 PET 使得等值测试不消耗任何通信量.

Efraim 等人[61]改进了恶意安全的 2 方 PSI 协议[59]结合半诚实安全的多方 PSI 协议[84]构造出高效的恶意安全的多方 PSI 协议. 首先为解决直接结合协议造成通信量随参与方数量的增加而指数增长的问题对恶意 2 方协议[59]进行了改进: 通过 P_2 和 P_1 执行 k-out-of-N OT, 使得 P_2 获得 P_1 的 GBF G_1 的适当部分. 然后引入重随机化 GBF 的概念, P_2 重随机化 GBF 得到 reGBF. 双方再次执行 k-out-of-N OT, 其中 P_2 和 P_1 角色互换, 让 P_1 获得 P_2 的 reGBF 适当部分, 该过程避免了直接发送 GBF 而泄露 P_1 不在交集的元素信息. 最后 P_2 通过比较 reGBF 和自身 GBF 得到隐私集合交集. 为将 2 方协议推广到多方协议, 首先让 P_0 与每一方 P_i 执行 2 次 k-out-of-N OT 协议, OT 执行之前先执行 Cut-and-Choose 技术以抵抗恶意敌手, 然后 P_0 将其得到的所有 GBF 进行异或操作得到 GBF G_0. 同时 P_i 与 P_0 交互得到的 GBF 和自身 GBF 异或得到 G_i. 最后为克服 P_i 直接将 G_i 发送给 P_0 使得 P_0 可以获得与每一方的交集问题, 所有参与方需执行 t-t 秘密共享 G_i, 并将得到的份额求和再发给 P_0, 求得多方交集.

4 新型场景的隐私集合交集

4.1 非平衡 PSI

传统 PSI 的大部分方案往往要求参与方集合的大小相等或相近, 并且假设了双方具有相似的计算和存储能力, 称之为平衡 PSI. 然而, 在某些实际应用中, 如隐私联系人发现, 客户拥有几百到几千个联系人集合, 而服务方拥有百万甚至千万级的用户集合. 它们的集合大小不平衡, 技术能力和存储空间也相差甚远. 使用平衡 PSI 协议, 它们的通信开销和计算开销均与较大集合成线性关系, 导致客户端需承受巨大的存储与计算开销. 这激发了对不平衡 PSI 的研究.

目前, 只有少数研究考虑了集合大小不对称的情况: Chen 等人[86]使用同态加密将大小为 N 的集合的通信量减少到 $O(\log N)$. 接收方加密集合并将其发送给发送方, 发送方通过计算适当的比较电路来计算同态加密数据的交集. 使用同态乘法将输出压缩到更小的尺寸, 并发送回接收方进行解密. 在协议中, 接收方只执行相对较轻的计算, 当接收方的计算能力有限时（如移动设备）, 该协议十分有效. 在此基础上, Chen 等人[87]使用 OPRF 预处理阶段来实现比文献[86]更高的性能和安全性, 实现了针对恶意客户端和恶意服务器的安全性. 并将集合求交扩展到带标签 PSI 的特殊用例, 其对于相交元素传输相关联的标签. 协议的优点是它们的通信复杂度是次线性的, 而不是服务器集合大小的线性, 然而, 协议的缺点是以重复的高计算开销作为代价.

Kiss 等人[88]描述了一种将$O(N)$通信量放在预处理阶段的方法, 其中服务器为每个元素 x 发送包含 $AES(k,x)$的大型 Bloom 过滤器. 各方使用 Yao 协议来对客户端的每个元素上的 AES 进行模糊评估, 客户端可测试 Bloom 过滤器的成员资格. 即服务器只能对其数据执行 1 次操作, 并将结果发送给客户端, 客户端将在未来的执行中使用它们来计算交集. Resende 等人[89]设计了一个更节省空间的布谷鸟过滤器用于服务器预处理阶段发送大消息, 以节省存储和通信成本. 为了进一步减少通信, 发送方将集合 X 转发给接收方之前对其进行压缩. 虽然这种压缩确实减少了通信, 但在高压缩率下产生误报, 接收方以不可忽略的概率输出 $Y\setminus X$ 中的元素.

Resende 等人[90]通过基于排名和选择的商过滤器(rank and select based quotient filter, RSQF)来减少协议交换的数据量, 相比其他数据结构具有更小的存储空间, 支持删除、插入、查找、合并等操作, 当集合达到最大容量时可重构过滤器, 而无需从头开始生成新的过滤器. 在隐私联系人查找等应用场景具有重要意义. 该文将 RSQF 数据结构引入 PSI 协议中, 通过放宽常数时间执行加速椭圆曲线提高协议计算性能. 离线阶段, 服务器对元素使用插入操作生成 RSQF, 并将 RSQF 发送给客户端. 在线阶段, 客户端和服务端进行交互以掩盖客户端元素, 客户端通过查找操作判断它的元素是否属于 RSQF, 从而得到集合的交集.

在最近的研究中, Lv 等人[91]研究了在不平衡场景下计算集合交集的基数, 利用 Bloom 过滤器构造低通信复杂度计算非平衡私有集合交集基数. 接收方不需要用低功耗设备加密发送方的数据集, 当接收方的数据集比发送方的数据集小得多时, 协议实现了高效率.

4.2 云辅助 PSI

随着云技术的发展, 基于云服务器的 PSI 协议逐渐成为热点. 云服务器具有高计算能力和存储容量, 为现有的 PSI 协议提供了成熟的优化方法, 但又产生了数据外包的隐私泄露问题. 基于云辅助的 PSI 协议可分为数据加密和数据盲化 2 种.

kerschbaum[92]提出服务器代理其中一个客户端 A 与另一个客户端 B 利用单向函数实现 PSI 操作, 且代理是一次性. 之后, kerschbaum[93]提出将 2 个客户端的 BF 通过同态加密外包给云服务器而不是对数据本身加密外包给服务器, 服务器进行 PSI 操作. 客户端只需本地保留集合元素以验证得到的交集 BF 里包含的本地元素. Liu 等人[94]直接利用对称和非对称加

密方案对数据集加密并外包给云服务器, 云服务器每进行一次 PSI 操作时, 客户端都需下载加密向量, 且会向服务器泄露交集基数. Kamara 等人[95]基于伪随机密钥对集合元素进行盲化处理再发送给云服务器, 但交集计算任务仍在客户端中进行, 服务器只为某一个客户端重新编码集合来维护计算隐私性. Qiu 等人[96]将隐私集合求交计算完全委托给服务器, 但需要确保服务器可信, 因为云服务器可不经客户端同意进行 PSI 计算并得到交集基数. 以上基于云服务器的 PSI 协议并不能完成云服务器的理想功能, 如数据集仍需本地存储, 数据需反复上传下载, PSI 计算不能完全委托给云服务器等.

Abadi 等人[47]提出 O-PSI, 利用点值多项式和 Paillier 同态加密实现将隐私集合求交操作完全委托给服务器, 无需本地维护集合, 客户只需上传 1 次外包数据集即可应用到多次隐私集合交集计算的理想云服务器功能, 但是其方案基于公钥密码操作, 过于昂贵. Abadi 等人[97]提出基于 Hash 表和点值多项式表示集合构造了一种即具有云服务器理想功能又无需公钥操作的高效 PSI 方案, 称为 EO-PSI. 但各方之间需事先建立安全通道, 否则攻击者可以窃听隐私信息. Kavousi 等人[98]提出无需安全通道的改进协议. O-PSI 理想云服务器功能构造如下: 客户端 A, B 为伪随机函数选取随机密钥 k. 构造多项式点值对 $\{(x_1,y_1),(x_2,y_2),\dots,(x_n,y_n)\}$表示集合, 将由 y_i 组成的向量$\boldsymbol{Y}=(y_1,y_2,\dots,y_n)$通过随机值$r_i=f(k,i)$进行盲化, 得到向量$\boldsymbol{v}=(y_1r_1,\ y_2r_2,\dots,y_nr_n)$发送给服务端. 客户端若同意云服务器计算交集, 则客户端 B 使用 Paillier 加密盲化值, 发送给客户端 A, A 再对其加密发生给服务器. 最后服务器利用加法同态性质进行 PSI 操作得到向量 T 将其发送给客户端 B, 客户端利用私钥解密得到由 $y_i(A)$和 $y_i(B)$组成的 z_i, 最后通过点值对(x_i,z_i)进行多项式插值得到交集元素.

Li 等人[99]在云计算环境下设计了多方隐私集合求并(求交)方案. 通过输入集合域已知的 1-r 编码(0-r 编码)将求并(求交)集合表示为向量, 借助哥德尔编码将向量编码为自然数 x, 利用同态加密得到密文 $E(x)$以防止泄露明文, 利用模运算对密文进行秘密共享以防止云服务器合谋, 利用模运算性质对所有密文相乘再通过算术基本定理展开得到并(交)集集合, 但方案限制了输入集合的范围以及不能进行 2 方计算.

Abadi 等人[100]提出支持外包数据集有效更新和可扩展的多方隐私集合求交协议 Feather, 允许客户端只上传 1 次私有数据集即可无限次委托计算, 无需公钥密码操作使其具有高效率和高可伸缩性. 协议由 3 部分组成, 设置阶段: 客户端构造 Hash 表并为每个桶生成 1 个多项式、BF 和标签, 利用伪随机函数盲化 BF, 并随机排列桶、盲化 BF 和标签. 客户端保留标签到桶的映射和排列, 向云发送置换的桶、BF 和标签. 更新阶段: 客户端利用 Hash 函数确定更改集合元素所在桶, 计算桶的标签将其发送给云, 云将标签对应的桶和盲化 BF 发送给客户端, 客户端利用密钥恢复桶的内容对其进行重新编码然后将更新的桶和过滤器发送给云服务器. PSI 计算阶段是基于文献[97]的改进.

Debnath 等人[101]利用 BF、乘法同态加密、零知识证明-数据签名结合技术设计了一种抵抗恶意敌手的 O-PSI 结构. 协议实现了标准模型安全且取得了较低的线性复杂度. 协议包括一组客户端 $\{C_1,C_2,\dots,C_n\}$和 1 个云服务器 S. 如果客户端 C_i 想要学习和 C_j 的交集. C_i 通过签名消息对请求 C_j 许可, 如 C_j 许可则将签名消息对发送给 S. S 计算结果集并将结果集和从 C_j 收到的签名消息对发送给 C_i. 最后 C_i 在本地计算交集结果.

Ali 等人[102]基于属性基加密的思想提出属性基私有集合求交 (AB-PSI) 方案, 可提供细粒度的访问控制, 并允许数据拥有者通过定义访问控制策略控制其外包数据集的访问权限, 实现数据拥有者不参与的情况下完成 PSI 操作, 云服务器通过隐私集合交集访问权限和盲化后数据集计算出相应的访问权限. 如果请求交集的客户端能得到一个有效的访问权限, 则可计算出其数据集和请求数据集的交集. Shi 等人[103]基于密钥策略属性加密(key-policy attribute-based encryption, KP-ABE)和 PSI 相结合提出一个新的概念: 基于委托密钥策略属性的外包加密数据集交集(KP-ABPSI), 以实现隐私集合交集计算完全委托给云服务器和对 PSI 的细粒度访问控制. 集合元素 d 被加密为 2 部分, 第 1 部分和元素 d 本身相关, 进行加密盲化; 第 2 部分和访问控制策略相对应的属性集相关. 数据用户生成与第 2 部分相关的令牌, 一旦令牌满足访问控制权限, 云服务器就能得到 d 被加密的第 1 部分, 最后进行 PSI 操作将结果返回给数据用户, 数据用户对其解密得到交集元素.

4.3 阈值 PSI

阈值 PSI 指当交集的基数大于或等于门限值时, 接收方才能获得隐私集合交集. 如网约顺风车, 在不泄露陌生人路径的情况下如何共享双方的公共路径是该场景的重点问题.

Hallgren 等人[104]首次引入 TPSI 解决该问题, 通过构造门限密钥封装机制, 实现交集的基数与阈值隐私比较, 从而得到 TPSI. Zhao 等人[105]基于不经意多项式评估构建了 1 个加密私有集合交集基数(ePSI-CA)协议, 通过外包扩展使协议更接近于现实世界的

模型. 以上 TPSI 协议首先计算交集基数, 再比较其与阈值 t 的关系考虑是否执行 PSI 协议, 其通信复杂度依赖于集合的大小。

Ghosh 等人[106] 提出了第 1 个通信复杂度仅依赖门限 t 的 PSI 协议, 其通过测试两集合是否足够相似而非计算交集基数。其基于较弱假设，计算效率高但通信复杂度为$O(t^2)$, 具体思想如下：双方将集合各自编码为多项式, 将两多项式相除, 消掉相同项得到阶数更低的有理函数, 通过比较有理函数阶数和阈值来判断是否执行 PSI 协议. Badrinarayanan 等人[107]设计了 2 种多方 TPSI 协议, 其通信复杂度随集合差的增大而增加, 当集合差值明显小于集合大小时, 该协议具有次线性通信复杂度. Branco 等人[108]基于门限加法同态加密方案提出了一种新的允许 N 方检查其输入集的交集是否大于 $N-t$ 的 TPSI 方案, 该协议通信复杂度为$O(NT^2)$.

Mahdavi 等人[109]介绍了另一种 TPSI 场景: 多方分别持有 1 个集合, 希望了解哪些元素至少出现在 t 个集合中, 而其它信息不被泄露, 称为超阈值多方 PSI(OT-MP-PSI), 通过构造不经意伪随机秘密共享(oblivious pseudo-random secret sharing, OPR-SS)实现 OT-MP-PSI 协议, 其通信复杂度为 $O(nmk)$. 共享阶段: 密钥持有方持有密钥 k 和值 S, 以及一组参与者 P_i 持有输入集合 S_i. 每个参与方 P_i 输入 S_i 和随机值 r_i, 得到共享集合. 由于具有 OPRF 的安全性, 保证参与方 P_i 不知道密钥 k, 密钥持有者不知道集合 S_i. 重构阶段: 参与方将共享集合构建为 Hash 表并发送给重建者, 重建者对所有的参与方选择 t 个执行拉格朗日插值验证每一行的元素是否大于等于阈值 t, 然后将其发送给参与方. 此过程重建者无法知道元素 $S(i)$, 但参与方可以知道自己的哪些集合元素至少出现在其他 t-1 个集合中.

Zhang 等人[110]基于 GBF 和秘密共享构建了一个新的 TPSI, 其通信复杂度为 $O(\lambda m)$, 通过设计一种新的 GBF 来建立集合元素和秘密共享之间的关系, 并使用其作为 TPSI 的门限检测, 并通过秘密关系方案确定$|X \cap Y|$和门限 t 的关系, 只有当交集中有足够的元素时接收者才能重构秘密共享方案的多项式以获得交集. 结合 Reed-Solomon 编码算法改进了秘密共享的重构阶段, 通过忽略错误的共享而避免计算所有可能的共享组合重构秘密的可行方法.

4.4 多方 PSI

隐私集合交集目前存在的高效协议大多只针对 2 方设置, 多方 PSI 的高效协议并没有引起多大关注. 这可能与各方之间不可避免的通信而导致极大的通信成本有关. 目前关于多方设置的 PSI 协议大多采用 2 种网络模型: 一种是星型拓扑网络结构减少双方之间的中间交流, 但给指定方带来了很高的工作量. 另一种是星型-路径网络结构其使每一方(除指定方)的通信量和计算复杂度仅取决于自身输入集大小. 还可能与如何保证只能获得所有参与方的集合交集, 而不能获得部分参与方的集合交集有关. 目前多方 PSI 协议主要通过秘密共享解决该问题, 使得只能秘密重构交集元素. Kolesnikov 等人[63]通过指定方与各方执行 OPPRF 协议实现交集元素的零共享, Hazay 等人[46]通过构造门限同态加密方案实现元素的零共享. 零共享指如果所有各方都持有相同的值 x, 共享异或得到 0, 否则得到 1 个随机值. Inbar 等人[84]、Efraim 等人[61]利用 GBF 和 OT 实现元素秘密共享. Kavousi 等人[74]不再对集合元素进行秘密共享而是对密钥进行秘密共享, 通过路径结构, 逐一得到由密钥 k 加密的 OPRF 值, 指定方对 OPRF 值进行比较得到交集结果. 为了对上述提到的多方协议性能有更加深刻的认识, 从通信量(指定方, 其他方)、计算量、安全模型、设计思想及隐藏技术、网络结构等方面总结如表 3 所示：

Table 3 Complexity Analysis for Multiparty PSI

表 3 多方 PSI 复杂性分析

协议	通信量		计算量		安全模型	技术	网络结构
	指定方	其他方	指定方	其他方			
文献[63]	$O(tn\lambda)$	$O(tn\lambda)$	$O(tk)$	$O(tk)$	半诚实	OPRF+OT	星型
文献[63]	$O(tn\lambda)$	$O(n\lambda)$	$O(tk)$	$O(k)$	强半诚实	OPRF+OT	星型
文献[46]	$O(tn\lambda)$	$O(n\lambda)$	$O(tn\log n)$	$O(n)$	恶意	OPE+HE	星型
文献[84]	$O(tn\lambda k)$	$O(tn\lambda k)$	$O(tn\lambda k)$	$O(tn\lambda k)$	半诚实	BF+OT	星型
文献[84]	$O(\log(t)n\lambda k)$	$O(\log(t)n\lambda k)$	$O(tn\lambda k)$	$O(tn\lambda k)$	强半诚实	BF+OT	星型
文献[61]	$O(tnk(\log(nk)+k))$	$O(tnk(\log(nk)+k))$	$O(tnk(\log(nk)+k))$	$O(tnk(\log(nk)+k))$	恶意	BF+OT	星型
文献[74]	$O(tn\lambda)$	$O(n\lambda k)$	$O(tn\lambda)$	$O(n\lambda k)$	半诚实	OPRF+OT	星型-路径

t:参与方数量；*n*:集合大小；*k*:Hash 函数个数；*λ*:比特长度

5 总结

综上所述, 随着安全多方技术研究的逐步深入, PSI 作为安全多方计算的一种重要应用, 已被广泛应用于隐私计算, 具有重要的理论和实践意义. 首先介绍 PSI 协议的密码技术、敌手模型、安全证明、编程框架, 其次系统总结了传统 PSI 协议的密码框架, 随后介绍 PSI 协议中集合元素比较技术, 进一步地详细阐述了适应新型应用场景的 PSI 方案. 随着密态数据的隐私计算技术进一步深入, 传统的 PSI 在安全性、高效性、适用性、可扩展性等方面受到了巨大的挑战. 因此, 未来的主要研究方向建议如下:

1) 考虑威胁性更高的场景, 设计不同安全性需求的 PSI 协议. 在目前主流 PSI 协议设计中, 通常只考虑半诚实的敌手. 然而在协议执行过程中可能会有黑客等侵入、篡改甚至伪造数据, 因此需将半诚实模型推广到恶意模型, 协议需要保证在恶意攻击下仍然使得数据具有一致性与可用性.

2) 考虑更多的参与方, 从而增加适用范围. 传统的 PSI 协议一般只有 2 个参与方(即发送方和接收方), 而 2 方 PSI 协议所使用的技术在一般无法简单推广至构建多方 PSI 协议, 且会导致部分隐私数据的泄露, 多方 PSI 协议允许多个参与方共同计算所有参与方的交集, 这使得问题的难度进行了提升, 也增加了 PSI 协议的适用范围.

3) 面向新型场景构建 PSI 协议. 在某些特定场景中, 通过 PSI 协议得到的交集元素仍然是敏感信息, 如电子医疗中的病患数据、基因测序中的序列数据等, 需要对现有 PSI 协议进行改造, 允许参与方在交集保密的情况下对交集中的元素进行各种函数的运算, 如计数、求和等.

4) 提高现有方案的效率. 目前大部分的 PSI 方案都是复杂的密码学操作, 如基于全同态加密、不经意传输、混淆电路、公钥加密等, 计算或通信开销随数据集大小呈现线性增长. 目前 PSI 协议在百万数据集下的运算速度尚能接受, 但当数据集大小达到百亿数量级时, 传统的 PSI 方案效率会大幅度下降, 迫切需要构建面向海量数据的高效 PSI 协议, 实现数据的共享.

参 考 文 献

[1] Yao A. Protocols for secure computations[C]//Proc of the 23rd Annual Symp on Foundations of Computer Science(SFCS 1982). Piscataway,NJ: IEEE, 1982: 160-164

[2] Duong T, Phan D, Trieu N. Catalic: Delegated psi cardinality with applications to contact tracing[C]// Proc of the 26th Int Conf on the Theory and Application of Cryptology and Information Security. Berlin:Springer, 2020: 870-899

[3] Demmler D, Rindal P, Rosulek M, et al. PIR-PSI: Scaling private contact discovery[J]. Proceedings on Privacy Enhancing Technologies, 2018, 2018(4): 159-178

[4] Lv Siyi, Ye Jinhui, Yin Sijie, et al. Unbalanced private set intersection cardinality protocol with low communication cost[J]. Future Generation Computer Systems, 2020, 102: 1054-1061

[5] Shen Liyan, Chen Xiaojun, Wang Dakui, et al. Efficient and private set intersection of human genomes[C]//Proc of IEEE Int Conf on Bioinformatics and Biomedicine (BIBM'18). Piscataway, NJ:IEEE, 2018: 761-764

[6] Meadows C. A more efficient cryptographic matchmaking protocol for use in the absence of a continuously available third party[C]//Proc of the 7th IEEE Symp on Security and Privacy. Los Alamitos, CA : IEEE Computer Society, 1986: 134-134

[7] Huberman B, Franklin M, Hogg T. Enhancing privacy and trust in electronic communities[C]//Proc of the 1st ACM Conf on Electronic Commerce. New York:ACM, 1999: 78-86

[8] Freedman M, Nissim K, Pinkas B. Efficient private matching and set intersection[C/OL]// Proc of the 23rd Int Conf on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2004[2020-10-16]. https://link.springer.com/content/pdf/10.1007/978-3-540-24676-3_1.pdf

[9] Shen Liyan, Chen Xiaojun, Shi Jinqiao, et al. Survey on private preserving set intersection technology[J]. Journal of Computer Research and Development, 2017, 54(10):2153-2169(in Chinese)
（申立艳,陈小军,时金桥,等.隐私保护集合交集计算技术研究综述[J].计算机研究与发展,2017,54(10):2153-2169）

[10] Shamir A. How to share a secret[J]. Communications of the ACM, 1979, 22(11): 612-613

[11] Blakley G. Safeguarding cryptographic keys[C]//Proc of American Federation of Int Processing Socicties National Computer Conf. Los Alamitos, CA: IEEE Computer Society, 1979: 313-313

[12] Jia Xingxing, Wang Daoshun, Nie Daxin, et al. A new threshold changeable secret sharing scheme based on the Chinese remainder theorem[J]. Information Sciences, 2019, 473: 13-30

[13] Paillier P. Public-key cryptosystems based on composite degree residuosity classes[C]//Proc of the 18th Int Conf on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 1999: 223-238

[14] Rivest R, Shamir A, Adleman L. A method for obtaining digital

signatures and public-key cryptosystems[J]. Communications of the ACM, 1978, 21(2): 120-126

[15] ElGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms[J]. IEEE Transactions on Information Theory, 1985, 31(4): 469-472

[16] Gentry C. A fully homomorphic encryption scheme[D]. Stanford, CA: Stanford University. 2009

[17] Even S, Goldreich O, Lempel A. A randomized protocol for signing contracts[J]. Communications of the ACM, 1985, 28(6): 637-647

[18] Crépeau C. Equivalence between two flavours of oblivious transfers[C]// Proc of the 7th Conf on the Theory and Application of Cryptographic Techniques. Berlin: Springer, 1987: 350-354

[19] Beaver D. Correlated pseudo randomness and the complexity of private computations [C]//Proc of the 28th Annual ACM Symp on Theory of Computing. New York: ACM, 1996: 479-488

[20] Ishai Y, Kilian J, Nissim K, et al. Extending oblivious transfers efficiently[C]//Proc of the 23rd Annual Int Cryptology Conf. Berlin: Springer, 2003: 145-161

[21] Kolesnikov V, Kumaresan R. Improved OT extension for transferring short secrets[C]// Proc of the 33rd Annual in Cryptology (CRYPTO 2013). Berlin: Springer, 2013: 54-70

[22] Asharov G, Lindell Y, Schneider T, et al. More efficient oblivious transfer and extensions for faster secure computation[C]//Proc of the 20th ACM SIGSAC Conf on Computer & Communications Security. New York: ACM, 2013: 535-548

[23] Boyle E, Couteau G, Gilboa N, et al. Efficient pseudorandom correlation generators: Silent OT extension and more[C]//Proc of the 39th Annual Int Cryptology Conf. Berlin: Springer, 2019: 489-518

[24] Schoppmann P, Gascón A, Reichert L, et al. Distributed vector-OLE: Improved constructions and implementation[C]//Proc of the 26th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2019: 1055-1072

[25] Weng Chenkai, Yang Kang, Katz J, et al. Wolverine: Fast, scalable, and communication-efficient zero-knowledge proofs for boolean and arithmetic circuits[R/OL]. Cryptology ePrint Archive, 2020[2021-05-10]. https://eprint. iacr. org/2020/925

[26] Yang Kang, Weng Chenkai, Lan Xiao, et al. Ferret: Fast extension for correlated ot with small communication[C]//Proc of the 27th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2020: 1607-1626

[27] Micali S, Goldreich O, Wigderson A. How to play any mental game[C]//Proc of the 19th ACM Symp on Theory of Computing. New York: ACM, 1987: 218-229.

[28] Kolesnikov V. Gate evaluation secret sharing and secure one-round two-party computation[C]//Proc of the 11th Int Conf on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2005: 136-155

[29] Beaver D, Micali S, Rogaway P. The round complexity of secure protocols[C]//Proc of the 2nd Annual ACM Symp on Theory of Computing. New York: ACM, 1990: 503-513

[30] Naor M, Pinkas B, Sumner R. Privacy preserving auctions and mechanism design[C]//Proc of the 1st ACM Conf on Electronic Commerce. New York: ACM, 1999: 129-139

[31] Kolesnikov V, Schneider T. Improved garbled circuit: Free XOR gates and applications[C]// Proc of the 35th Int Colloquium on Automata, Languages, and Programming. Berlin: Springer, 2008: 486-498

[32] Pagh R, Rodler F F. Cuckoo hashing[J]. Journal of Algorithms, 2004, 51(2): 122-144

[33] Pinkas B, Schneider T, Zohner M. Scalable private set intersection based on OT extension[J]. ACM Transactions on Privacy and Security (TOPS), 2018, 21(2): 1-35

[34] Arbitman Y, Naor M, Segev G. Backyard cuckoo hashing: Constant worst-case operations with a succinct representation[C]//Proc of the 51st IEEE Annual Symp on Foundations of Computer Science. Piscataway, NJ: IEEE, 2010: 787-796

[35] Schneider T. Engineering Secure Two-Party Computation Protocols[M]. Berlin: Springer, 2011:5-27

[36] Hastings M, Hemenway B, Noble D, et al. Sok: General purpose compilers for secure multi-party computation[C]//Proc of the 40th IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2019: 1220-1237

[37] Demmler D, Schneider T, Zohner M. ABY-A framework for efficient mixed-protocol secure two-party computation[C/OL]//Proc of the 22nd Network and Distributed System Security Symp. Reston, VA: Internet Society, 2015[2020-11-13].https://www.ndss-symposium.org/ndss2015/

[38] Wang Xiao, Malozemoff A J, Katz J. EMP-toolkit: Efficient multiparty computation toolkit[CP/OL]. [2021-05-20].https://github.com/emp-toolkit,

[39] Zahur S, Evans D. Obliv-C: A language for extensible data-oblivious computation[J/OL]. IACR Cryptology ePrint Archive, 2015[2020-11-30]. https://eprint.iacr.org/2015/1153

[40] Bogdanov D, Laur S, Willemson J. Sharemind: A framework for fast privacy-preserving computations[G]//LNCS 5283: Proc of the 27th European Symp on Research in Computer Security. Berlin: Springer, 2008: 192-206

[41] Zhang Yihua, Steele A, Blanton M. PICCO: A general-purpose compiler for private distributed computation[C]//Proc of the 20th ACM SIGSAC Conf on Computer & Communications Security. New York: ACM, 2013: 813-826

[42] De Cristofaro E, Tsudik G. Experimenting with fast private set intersection[C]// Proc of Int Conf on Trust and Trustworthy Computing. Berlin: Springer, 2012: 55-73

[43] Pinkas B, Schneider T, Zohner M. Faster private set intersection based on OT extension[C]//Proc of the 23rd USENIX Security Symp.

Berkeley. CA: USENIX Association, 2014: 797-812

[44] Freedman M J, Hazay C, Nissim K, et al. Efficient set intersection with simulation-based security[J]. Journal of Cryptology, 2016, 29(1): 115-155

[45] Kissner L, Song D. Privacy-preserving set operations[C]//Proc of the 25th Annual Int Cryptology Conf. Berlin: Springer, 2005: 241-257

[46] Hazay C, Venkitasubramaniam M. Scalable multi-party private set-intersection[C]//Proc of the 20th IACR International Workshop on Public Key Cryptography. Berlin: Springer, 2017: 175-203

[47] Abadi A, Terzis S, Dong C. O-PSI: Delegated private set intersection on outsourced datasets[C]//Proc of the 27th IFIP Int Information Security and Privacy Conf. Berlin: Springer, 2015: 3-17

[48] Jarecki S, Liu Xiaomin. Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection[G]//LNCS 5444: Proc of the 6th Theory of Cryptography Conf. Berlin: Springer, 2009: 577-594

[49] Dou Jiawei, Liu Xuhong, Wang Wenli et al. Efficient and secure calculation of two-party sets in the field of rational numbers[J].Chinese Journal of Computers,2020,43(8):1397-1413(in Chinese)
(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[J].计算机学报,2020,43(8):1397-1413)

[50] Damgård I, Pastro V, Smart N, et al. Multiparty computation from somewhat homomorphic encryption[C]//Proc of the 32nd Annual Cryptology Conf. Berlin: Springer, 2012: 643-662

[51] Huang Y, Evans D, Katz J. Private set intersection: Are garbled circuits better than custom protocols[C/OL]//Proc of the 19th Network and Distributed System Security Symp. Reston, VA: ISOC, 2012[2020-10-21].http://www.cs.virginia.edu/~evans/pubs/ndss2012/

[52] Pinkas B, Schneider T, Segev G, et al. Phasing: Private set intersection using permutation-based hashing[C]// Proc of the 24th USENIX Security Symp. Berkeley, CA: USENIX Association, 2015: 515-530

[53] Pinkas B, Schneider T, Weinert C, et al. Efficient circuit-based PSI via cuckoo hashing[C]// Proc of the 38th Annual Int Conf on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2018: 125-157

[54] Pinkas B, Schneider T, Tkachenko O, et al. Efficient circuit-based psi with linear communication[C]// Proc of the 39th Annual Int Conf on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2019: 122-153

[55] Chandran N, Gupta D, Shah A. Circuit-PSI with linear complexity via relaxed batch OPPRF[R/OL]. IACR Cryptology ePrint Archive, 2021[2021-03-25]. https://eprint.iacr.org/2021/034

[56] Keller M, Orsini E, Scholl P. Actively secure ot extension with optimal overhead[G]//LNCS 9215: Proc of the 35th Annual Cryptology Conf. Berlin: Springer, 2015: 724-741.

[57] Orrù M, Orsini E, Scholl P. Actively secure 1-out-of-n ot extension with application to private set intersection[C]//Proc of Cryptographers' Track at the RSA Conf. Berlin: Springer, 2017: 381-396

[58] Dong Changyu, Chen Liqun, Wen Zikai. When private set intersection meets big data: An efficient and scalable protocol[C]//Proc of the 20th ACM SIGSAC Conf on Computer & Communications Security. New York: ACM, 2013: 789-800

[59] Rindal P, Rosulek M. Improved private set intersection against malicious adversaries[C]// Proc of the 27th Annual Int Conf on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2017: 235-259

[60] Zhang En, Liu Fenghao, Lai Qiqi,et al. Efficient multi-party private set intersection against malicious adversaries[C]//Proc of the 27th ACM SIGSAC Conf on Cloud Computing Security Workshop. New York: ACM, 2019: 93-104

[61] Efraim A B, Nissenbaum O, Omri E, et al. Psimple: Practical multiparty maliciously-secure private set intersection[R/OL]. IACR Cryptology ePrint Archive, 2021[2021-05-13]. https://eprint. iacr. org/2021/122

[62] Kolesnikov V, Kumaresan R, Rosulek M, et al. Efficient batched oblivious PRF with applications to private set intersection[C]//Proc of the 23rd ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2016: 818-829

[63] Kolesnikov V, Matania N, Pinkas B, et al. Practical multi-party private set intersection from symmetric-key techniques[C]//Proc of the 24th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2017: 1257-1272

[64] Pinkas B, Rosulek M, Trieu N, et al. Spot-light: Lightweight private set intersection from sparse ot extension[C]// Proc of the 39th Annual Int Cryptology Conf. Berlin: Springer, 2019: 401-431

[65] Chase M, Miao Peihan. Private set intersection in the internet setting from lightweight oblivious PRF[C]// Proc of the 40th Annual Int Cryptology Conf. Berlin: Springer, 2020: 34-63

[66] Pinkas B, Rosulek M, Trieu N, et al. PSI from PaXoS: fast, malicious private set intersection[C]// Proc of the 39th Annual Int Conf on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2020: 739-767

[67] Rindal P, Schoppmann P. VOLE-PSI: Fast OPRF and circuit-PSI from vector-OLE[R/OL]. IACR Cryptology ePrint Archive, 2021[2021-05-09]. https://eprint.iacr.org/2021/266

[68] Naor M, Reingold O. Number-theoretic constructions of efficient pseudo-random functions[C]//Proc of the 38th Annual Symp on Foundations of Computer Science. Piscataway, NJ: IEEE, 1997: 458-467

[69] Hazay C, Lindell Y. Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries[C]// Proc of the 5th Theory of Cryptography Conf. Berlin: Springer, 2008: 155-175

[70] Dodis Y, Yampolskiy A. A verifiable random function with short proofs and keys[C]// Proc of the 8th Int Workshop on Public Key Cryptography.

Berlin: Springer , 2005: 416-431

[71] Debnath S K, Dutta R. Towards fair mutual private set intersection with linear complexity[J]. Security and Communication Networks, 2016, 9(11): 1589-1612

[72] Jarecki S, Liu Xiaomin. Fast secure computation of set intersection[C]// Proc of the 7th Int Conf on Security and Cryptography for Networks. Berlin: Springer, 2010: 418-435

[73] Sanders P, Egner S, Korst J. Fast concurrent access to parallel disks[J]. Algorithmica, 2003, 35(1): 21-55

[74] Kavousi A, Mohajeri J, Salmasizadeh M. Efficient scalable multi-party private set intersection using oblivious PRF[R/OL]. IACR Cryptology ePrint Archivw, 2021[2021-04-25]. https://eprint.iacr.org/2021/484

[75] Hemenway Falk B, Noble D, Ostrovsky R. Private set intersection with linear communication from general assumptions[C]//Proc of the 18th ACM Workshop on Privacy in the Electronic Society. New York: ACM, 2019: 14-25

[76] Chen Zhenhua, Li Shundong, Wang Daoshun et al. The non-encrypted method securely calculates the set containment relationship [J]. Journal of Computer Research and Development, 2017, 54(7):1549-1556(in Chinese)
（陈振华,李顺东,王道顺,等.非加密方法安全计算集合包含关系[J].计算机研究与发展,2017,54(7):1549-1556）

[77] Zhou Sufang, Li Shundong, Guo Yiwen et al. Efficient calculation of the intersection of confidential sets[J]. Chinese Journal of Computers, 2018, 41(2): 464-480(in Chinese)
（周素芳,李顺东,郭奕旻,等.保密集合相交问题的高效计算[J].计算机学报,2018,41(2):464-480）

[78] Ruan Ou, Mao Hao. Efficient private set intersection using point-value polynomial representation[J/OL]. Security and Communication Networks, 2020[2020-11-04]. https://www.hindawi.com/journals/scn/2020/8890677

[79] Ciampi M, Orlandi C. Combining private set-intersection with secure two-party computation[C]// Proc of the 15th Int Conf on Security and Cryptography for Networks. Berlin: Springer, 2018: 464-482

[80] Mohassel P, Rindal P, Rosulek M. Fast database joins and PSI for secret shared data[C]//Proc of the 27th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2020: 1271-1287

[81] Song Xiangfu, Gai Min, Zhao Shengnan, et al. Privacy-preserving statistics protocol for set-based computation[J]. Journal of Computer Research and Development, 2020, 57(10): 2221-2231(in Chinese)
（宋祥福, 盖敏, 赵圣楠, 等. 面向集合计算的隐私保护统计协议[J]. 计算机研究与发展, 2020, 57(10): 2221-2231)

[82] Bloom B H. Space/time trade-offs in Hash coding with allowable errors[J]. Communications of the ACM, 1970, 13(7): 422-426

[83] Grandi F. On the analysis of Bloom filters[J]. Information Processing Letters, 2018, 129: 35-39

[84] Inbar R, Omri E, Pinkas B. Efficient scalable multiparty private set-intersection via garbled Bloom filters[C]//Proc of the 15th Int Conf on Security and Cryptography for Networks. Berlin: Springer, 2018: 235-252

[85] Karakoç F, Küpçü A. Linear complexity private set intersection for secure two-party protocols[C]// Proc of the 16th Int Conf on Cryptology and Network Security. Berlin: Springer, 2020: 409-429

[86] Chen Hao, Laine K, Rindal P. Fast private set intersection from homomorphic encryption[C]//Proc of the 24th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2017: 1243-1255

[87] Chen Hao, Huang Zhicong, Laine K, et al. Labeled PSI from fully homomorphic encryption with malicious security[C]//Proc of the 25th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2018: 1223-1237

[88] Kiss Á, Liu Jian, Schneider T, et al. Private set intersection for unequal set sizes with mobile applications[R/OL]. IACR Cryptology ePrint Archive, 2017[2020-10-19]. https://eprint.iacr.org/2017/670

[89] Resende A C D, Aranha D F. Faster unbalanced private set intersection[C]// Proc of the 22nd Int Conf on Financial Cryptography and Data Security. Berlin: Springer, 2018: 203-221

[90] Resende A C D, de Freitas Aranha D. Faster unbalanced private set intersection in the semi-honest setting[J]. Journal of Cryptographic Engineering, 2021, 11(1): 21-38

[91] Lv Siyi, Ye Jinhui, Yin Sijie, et al. Unbalanced private set intersection cardinality protocol with low communication cost[J]. Future Generation Computer Systems, 2020, 102: 1054-1061

[92] Kerschbaum F. Collusion-resistant outsourcing of private set intersection[C]//Proc of the 27th Annual ACM Symp on Applied Computing. New York: ACM, 2012: 1451-1456

[93] Kerschbaum F. Outsourced private set intersection using homomorphic encryption[C]//Proc of the 7th ACM Symp on Information, Computer and Communications Security. New York: ACM, 2012: 85-86

[94] Liu Fang, Ng W, Zhang Wei, et al. Encrypted set intersection protocol for outsourced datasets[C]//Proc of the 2nd IEEE Int Conf on Cloud Engineering(IC2E). Piscataway, NJ: IEEE, 2014: 135-140

[95] Kamara S, Mohassel P, Raykova M, et al. Scaling private set intersection to billion-element sets[C]//Proc of the 18th Int Conf on Financial Cryptography and Data Security. Berlin: Springer, 2014: 195-215

[96] Qiu Shuo, Liu Jiqiang, Shi Yanfeng, et al. Identity-based private matching over outsourced encrypted datasets[J]. IEEE Transactions on Cloud Computing, 2015, 6(3): 747-759

[97] Abadi A, Terzis S, Metere R, et al. Efficient delegated private set intersection on outsourced private datasets[J]. IEEE Transactions on Dependable and Secure Computing, 2017, 16(4): 608-624

[98] Kavousi A, Mohajeri J, Salmasizadeh M. Improved secure efficient delegated private set intersection[C/OL]//Proc of the 28th Iranian Conf

on Electrical Engineering (ICEE).Piscataway, NJ: IEEE, 2020[2021-03-08]. www.icee.ir/en

[99] Li Shundong, Zhou Sufang, Guo Yimin et al. Secure set computing in cloud environment[J].Journal of Software, 2016,27(6):1549−1565 (in Chinese)
（李顺东, 周素芳, 郭奕旻, 等. 云环境下集合隐私计算[J]. 软件学报, 2016,27(6): 1549-1565)

[100] Abadi A, Terzis S, Dong C. Feather: Lightweight multi-party updatable delegated private set intersection[R/OL]. IACR Cryptology ePrint Archive, 2020[2021-01-06]. https://eprint.iacr.org/2020/407

[101] Debnath S K, Sakurai K, Dey K, et al. Secure outsourced private set intersection with linear complexity[C/OL]//Proc of the 6th IEEE Conf on Dependable and Secure Computing (DSC). Piscataway, NJ: IEEE, 2021[2021-07-25]. https://ieeexplore.ieee.org/xpl/conhome/9346211/proceeding

[102] Ali M, Mohajeri J, Sadeghi M R, et al. Attribute-based fine-grained access control for outscored private set intersection computation[J]. Information Sciences, 2020, 536: 222-243

[103] Shi Yanfeng, Shuo Qiu. Delegated key-policy attribute-based set intersection over outsourced encrypted data sets for CloudIoT[J/OL]. Security and Communication Networks, 2021[2021-05-30]. https://www.hindawi.com/journals/scn/2021/5595243/

[104] Hallgren P, Orlandi C, Sabelfeld A. Privatepool: Privacy-preserving ridesharing[C]//Proc of the 30th IEEE Computer Security Foundations Symp (CSF). Piscataway, NJ: IEEE, 2017: 276-291

[105] Zhao Yongjun, Chow S. Can you find the one for me?[C]//Proc of the 18th Workshop on Privacy in the Electronic Society. New York: ACM, 2018: 54-65

[106] Ghosh S, Simkin M. The communication complexity of threshold private set intersection[C]//Proc of the 39th Annual Int Cryptology Conf. Berlin: Springer, 2019: 3-29

[107] Badrinarayanan S, Miao P, Rindal P. Multi-party threshold private set intersection with sublinear communication[R/OL]. IACR Cryptology ePrint Archive, 2021[2021-03-07]. https://eprint.iacr.org/2020/600

[108] Branco P, Döttling N, Pu Sihang. Multiparty cardinality testing for threshold private set intersection[C]//Proc of the 24th IACR Int Conf on Public-Key Cryptography. Berlin: Springer, 2021: 32-60.

[109] Mahdavi R A, Humphries T, Kacsmar B, et al. Practical over-threshold multi-party private set intersection[C]//Proc of the 27th Annual Computer Security Applications Conf. New York: ACM, 2020: 772-783

[110] Zhang En, Chang Jian, Li Yu. Efficient threshold private set inters ection[J]. IEEE Access, 2021, 9: 6560-6570

Wei Lifei, born in 1982. PhD. Associate professor, Master supervisor and senior member of CCF. His main research interests include information security, privacy preserving and cryptography.
魏立斐，1982 年生. 博士，副教授，硕士生导师，CCF 高级会员. 主要研究方向为信息安全、隐私保护、密码学.

Liu jihai, born in 1998. Master candiadate. Student member of CCF. His main research interests include secure computation and information security.
刘纪海，1998 年生. 硕士研究生，CCF 学生会员. 主要研究方向为安全计算、信息安全.

Zhang Lei, born in 1983. PhD. Assistant professor. Her main research interests include applied cryptography, big data security and access control(Lzhang@shou.edu.cn).
张蕾，1983 年生. 博士，讲师. 主要研究方向为应用密码学、大数据安全、访问控制.

Wang Qin, born in 1996. Master candiadate. Student member of CCF. His main research interests include information security and secure computation(913377391@qq.com).
王勤，1996 年生. 硕士研究生，CCF 学生会员. 主要研究方向为信息安全、安全计算.

He Chongde, born in 1997. Master candiadate. His main research interests include cryptography and information security(15513097134@163.com).

贺崇德，1997 年生．硕士研究生．主要研究方向为密码学、信息安全.