计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210300308

基于云服务器辅助的多方隐私交集计算协议

王 勤 魏立斐 刘纪海 张 蕾
上海海洋大学信息学院 上海 201306
(qinwang97@foxmail.com)

摘 要 隐私集合交集(Private Set Intersection,PSI)技术允许私有集合数据持有方联合计算出集合交集而不泄露交集外的任何隐私信息。作为安全多方计算中的重要密码学工具,该技术已被广泛应用于人工智能和数据挖掘的安全领域。随着多源数据共享时代的到来,大多数PSI协议主要解决两方隐私集合交集问题,一般无法直接推广到多方隐私交集计算场景。文中设计了基于云服务器辅助的多方隐私交集计算协议,能将部分计算和通信外包给不可信云服务器而又不会泄露任何隐私数据,通过使用不经意伪随机函数、秘密共享和键值对打包方法使得协议更高效。通过模拟范例证明了协议在半诚实模型下能够安全地计算多方隐私集合交集,所有参与方和云服务器都无法窃取额外数据。与现有方案相比,所提协议受限制更少,适用范围更广。

关键词:隐私集合交集;安全多方计算;云计算;不可信云服务器;隐私计算
中图法分类号 TP309

Private Set Intersection Protocols Among Multi-party with Cloud Server Aided

WANG Qin,WEI Li-fei,LIU Ji-hai and ZHANG Lei
College of Information Technology,Shanghai Ocean University,Shanghai 201306,China

Abstract Private set intersection (PSI) is a secure multi-party computation technique that allows several parties,who each hold a set of private items,to compute the intersection of those private sets without revealing additional information. PSI has been widely used in the field of artificial intelligence security and data mining security. With the advent of the multi-source data sharing era,most PSI protocols mainly solve the problem of two-party privacy set intersection,which can not be directly extended to multi-party privacy intersection computing scenarios. This paper designs a multi-party privacy intersection protocol with the help of cloud servers,which can outsource a part of the computation and communication to untrusted cloud server without disclosing any privacy data. This paper makes the protocol more efficient by using the methods of oblivious pseudo-random functions,secret sharing and key-value pair packing. It proves that the PSI protocol can be secure in the semi-honest model and all participants and cloud servers can not obtain the additional data. Compared with the existing scheme,the proposed protocol has the merit of less restricted and more applicable in application scenarios.

Keywords Private set intersection,Secure multi-party computation,Cloud computing,Untrusted cloud server,Privacy computing

1 引言

隐私集合交集(PSI)技术允许私有集合数据持有方联合计算出集合交集而不泄露除交集外的任何隐私信息[1-2]。作为安全多方计算中的重要密码学工具,PSI技术是许多密码协议的基础模块,也被广泛应用于数据挖掘、人工智能和数据共享的安全领域,如带隐私保护的数据挖掘[3-4]、隐私通讯录查找[5]、新冠接触者追踪[6]等。现有的大部分PSI实现协议都致力于解决两方隐私交集计算问题,协议已有非常高的效率,但是此类PSI协议一般无法直接推广到多方隐私交集计算场景。随着多源数据共享时代的到来,支持多方参与的PSI协议的适用范围更广,与传统两方PSI协议相比其能产

到稿日期:2021-03-31 返修日期:2021-05-23
基金项目:国家重点研发计划海洋环境安全保障专项资助(2016YFC1403200);国家自然科学基金(61972241,61802248);上海市自然科学基金项目(18ZR1417300);上海市高等学校青年骨干教师国内访问学者项目(A1-2007-00-000503);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 Key Research and Development Program(2016YFC1403200),National Natural Science Foundation of China(61972241,61802248),Natural Science Foundation of Shanghai(18ZR1417300),Domestic Visiting Scholar Project of Shanghai Young Backbone Teachers in Colleges and Universities(A1-2007-00-000503) and 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:魏立斐(Lfwei@shou.edu.cn)

生更多的数据共享机会。目前只有少数的 PSI 协议适用于多方参与的场景[7-9]。在现实应用场景中,需要保密计算交集的参与方通常不止两个,如社交软件中的私有联系人查找功能常被要求查找多个用户的共同好友。

现有的大部分 PSI 实现协议都致力于解决两方隐私交集计算问题,如早期的 PSI 方案通常会将私有集合求解问题转为哈希值集合求解问题,虽然效率高,但该方法并不能完全保证数据隐私,在碰撞攻击下无法保证数据安全。现有安全多方计算领域存在如混淆电路等多种能安全地计算任何功能函数的通用方法,但由于使用混淆电路需要对每一比特位进行电路门运算,并且电路门数量多,因此通用方法与特定的隐私问题求解方案相比效率较低。文献[10]基于多项式的特殊性质和同态加密提出了一个在半诚实下安全的 PSI 方案,但该方案中构造的多项式次数较高,在密文状态下完成高阶多项式计算较为困难。为解决该问题,文献[10]在后续研究工作中使用哈希函数将双方集合元素映射到长度为 β 的二维容器,以构建朴素哈希表,使多项式次数大幅下降。文献[11]基于不经意传输拓展协议构造不经意伪随机函数(Oblivious Pseudo-Random Function,OPRF),高效地完成了隐私相等性测试并提出了一个 PSI 协议。文献[12-13]在文献[11]的基础上引入布谷鸟哈希映射技术[13],提出了一个高效的 PSI 协议,计算复杂度达到 $O(m)$,其中 m 为集合元素数量。文献[14]在文献[12]的基础上使用了一种新的不经意传输拓展协议,用于构造不经意伪随机函数以及 PSI 方案,其通信复杂度能达到 $O(m)$。文献[15]针对集合计算算法难以嵌入密文搜索等问题,提出利用秘密共享和离散对数构造 PSI 协议。文献[16]避免使用同态加密或混淆电路,基于不经意传输协议实现了一系列隐私集合相关的计算函数,计算复杂度较低。文献[17]构造了一个适用于处理有理数域集合元素的 PSI 方案。上述方案仅限于求解两方参与者之间的隐私集合交集,但不能直接推广到多方隐私集合交集的求解。

云计算技术为使用者提供了灵活、经济的计算资源,利用云服务器辅助计算隐私集合交集是一个可行的办法,然而,云服务器为用户带来便利的同时也引发了新的安全问题,不可信云服务器可能会违反数据隐私保密协定以窃取隐私数据。因此,不少基于云服务器的隐私交集方案应运而生。文献[18-20]构造了一系列适用于云计算环境的 PSI 协议,允许两方客户端利用云服务器的计算能力和存储空间辅助计算双方的隐私集合交集,而不向云服务器泄露集合中的任何敏感信息,但仅限于解决两方隐私交集计算问题。文献[7]和文献[8]分别提出了支持多方参与的隐私集合交集协议,但其计算与通信均由各参与方完成,参与方设备有限时则无法安全地利用云服务辅助计算。文献[9]基于哥德尔编码和同态加密,设计了一种适用于云环境的多方隐私集合计算协议,但该协议仅限于有限范围内计算多方隐私集合交并集,当输入范围未知时,不能保证得到正确的计算结果,若参与者将集合中的元素划分为不同的所属范围,则会泄露过多集合元素信息。在云计算环境下,现有隐私交集协议无法安全有效地处理多方隐私集合的交集问题。

因此,本文利用秘密共享、不经意伪随机函数和键值对打包技术,提出了一个适用于云环境的多方隐私交集计算协议,并通过哈希映射技术提升了协议的性能。在该协议中,假设各参与方均与云服务器不合谋,组织方将集合信息保密地外包给不可信云,由算力强大且通信廉价的不可信云服务器与各参与方进行交互计算,云服务器在计算过程中无法窃取各参与方的任何隐私数据。最终,组织方、参与方得到集合交集而对交集外的元素一无所知,云服务器在整个过程中也无法获得任何隐私集合信息。最后,本文实例化上述多方 PSI 计算协议,并分别嵌入基于多项式插值的键值对打包技术和基于混淆布隆过滤器的键值对打包技术对该协议进行测试与性能评估。本文的贡献如下:

(1)提出了一个适用于云计算环境的多方隐私交集计算方案,并利用不可信云的计算能力和通信能力辅助计算多方隐私集合的交集。

(2)选用两种具有代表性的键值对打包技术作为交集协议的子模块,实现的协议分别表现为计算速度快和通信量小。

(3)与现有适用于云环境的多方隐私交集协议进行比较,在仿真实验环境下测试协议在各阶段的性能表现,并给出了真实实验数据。

2 密码学原语

本节将介绍文中涉及的基本密码学原语和符号,$[n]$表示整数集$\{1,2,\cdots,n\}$,集合$\overline{X}$代表$\{X_1,X_2,\cdots,X_n\}$,κ 代表计算安全参数,λ 代表统计安全参数。

2.1 半诚实模型的安全性定义

半诚实参与者在协议执行的过程中能够正确地履行协议指令,但会记录下协议执行过程中所能收集到的一切信息,并试图利用计算过程中存储的中间信息推测额外的敏感信息。在密码学中,针对各种安全要求有着不同的安全性证明方法,目前半诚实模型下的安全多方计算协议普遍采用模拟范例进行证明,将引入可信第三方的理想安全多方计算协议的安全性与实际构造的协议的安全性进行对比,若实际协议未泄露更多信息,则说明构造协议安全[21]。本文方案涉及的参与方均假定为半诚实参与方,为证明协议的安全性,需要各参与方在仅知道己方输入和应得输出的情况下模拟出己方视图,并且在计算上不可区分。

本文提出的协议由 n 个参与方$P_1,P_2,\cdots,P_n$和一个云辅助服务器 S 构成,其中P_n为参与方中的组织者,参与方P_n持有集合X_i,计算协议可以表示为 $\pi:\perp\times X_1\times X_2\cdots\times X_n\rightarrow\perp\times f_{|\cap|}\times f_{|\cap|}\cdots\times f_{|\cap|}$。其中,$\perp$指空集,$f_{|\cap|}$为交集结果,即$X_1\cap X_2\cap\cdots\cap X_n$。协议对云服务器的输出为$\perp$,对各参与方的输出为$f_{|\cap|}$。当参与方$P_i$执行输入值为$\overline{X}$的协议$\pi$时,其视图可以表示为 $VIEW_i^{\pi}(\overline{X})=(X_i,r_i,M_i^1,\cdots,M_i^j)$,$r_i$为协议执行期间产生的随机数,$M_i^j$为$P_i$收到的第$j$条信息。

定义 1 在半诚实模型下令 f 表示上述确定性函数,如果存在概率多项式时间的模拟器Sim_{P_i}和Sim_s使得式(1)、

式(2)成立，则证明 π 能安全地计算 f。符号 $\overset{c}{\equiv}$ 表示计算上不可区分，参与方 $P_i \in P_1, P_2, \cdots, P_n$。

$$Sim_{P_i}(X_i, f_{|\cap|}) \overset{c}{\equiv} VIEW_{P_i}^{\pi}(\overline{X}) \tag{1}$$

$$Sim_s(\perp, \perp) \overset{c}{\equiv} VIEW_S^{\pi}(\overline{X}) \tag{2}$$

2.2　秘密共享

秘密共享(secret sharing)是密码学中的一个基本原语，指将秘密以适当方式拆分，拆分后的份额由不同参与方持有，单个参与方无法恢复出秘密信息，当且仅当足够数量的参与者协作时才能恢复出原始秘密值。文献[22]基于异或操作⊕(XOR)提出了一个高效的秘密共享方案，该方案产生 $n-1$ 个随机比特串 $s_1, s_2, \cdots, s_{n-1}$ 作为共享值，并逐位异或得到共享值 $s_n = s_1 \oplus \cdots \oplus s_{n-1} \oplus s$，当拥有秘密值 $s_1, s_2, \cdots, s_n$ 时可恢复出原始秘密 $s = s_1 \oplus s_2 \oplus \cdots \oplus s_n$，而缺少任何共享值 s_i 均无法解出秘密 s。

2.3　不经意伪随机函数

不经意伪随机函数(OPRF)由接收者和发送者两方参与，其允许接收者输入元素 x 得到 $F(k,x)$，发送者得到 OPRF 密钥 k，其中 F 表示伪随机函数，x 为接收方的输入值。文献[12]提出了一种以不经意传输拓展协议为基础的高效 OPRF 协议，参与双方仅需少量计算和通信就能得到大量 OPRF 实例。该 OPRF 结构中，OPRF 密钥为 (s,k)，s 为发送者选择的随机值，k 为协议执行后发送方得到的密钥值且 $k = t \oplus [C(x) \wedge s]$，其中 x 为接收方输入值，运算符 $\wedge$ 表示逐比特与运算，$C(\cdot)$ 表示汉明距离最小为 κ 的随机函数，t 为输入 x 后获得的 OPRF 输出值，由接收方接收且 $F(k,x) = t$。文献[6]使用满足异或同态性质的线性编码函数 $C(\cdot)$ 构造 OPRF，并证明了该方案的安全性，若使用满足 XOR 同态性的 $C(\cdot)$ 构造 OPRF，则 $[C(x_1) \wedge s] \oplus [C(x_2) \wedge s] = [C(x \oplus y) \wedge s]$ 成立，于是 OPRF 满足 XOR 同态性：

$$k_1 \oplus k_1 = (t_1 \oplus t_1) \oplus [C(x_1 \oplus x_2) \wedge s]$$

本文方案均使用上述满足 XOR 同态性，且仅需少量计算和通信就能得到大量实例的 OPRF 方案。

2.4　哈希映射算法

布谷鸟哈希算法[13]可将集合中的元素映射到一个密集的哈希表中，现已被广泛应用在隐私交集计算领域。基础布谷鸟哈希算法由一个长度为 β 的容器 $B[1 \cdots \beta]$、容错存储空间 Stash 和 k 个哈希函数 $h_1, h_2, \cdots, h_k: \{0,1\}^* \to [\beta]$ 组成，详细构建方法参见文献[13]。文献[5]通过实验证明，合理选择参数 k 和 β 能使布谷鸟哈希表的构建成功率大于 $(1-2^{-\lambda})$ 且不再需要容错空间 Stash。

使用布谷鸟哈希算法提升隐私集合交集计算性能时需要构建朴素哈希表。朴素哈希表可以理解为一个行容量为 η、共有 β 行的二维容器，选用上述相同的 k 个哈希函数 $h_1, h_2, \cdots, h_k: \{0,1\}^* \to [\beta]$，将集合 Y 映射至朴素哈希表中，元素 $y \in Y$ 若未遇地址碰撞，朴素哈希表中将有 k 行存有元素 y。将元素 x 映射到布谷鸟哈希表中，若存在 $x \in Y$，则必定会有元素 x 出现在使用相同哈希函数且由集合 Y 映射的朴素哈希表的同一行中。根据上述优良性质，行与行之间求解交集能大幅提升协议性能。

2.5　键值对打包技术

合理选择打包结构能使协议适用于不同的实际场景，应当根据实际需要选择合适的打包结构。本文选用计算上表现优秀的基于混淆布隆过滤器的键值对打包技术和在通信上表现优秀的基于多项式插值的键值对打包技术来打包键值对。

(1)基于混淆布隆过滤器的键值对打包方法

混淆布隆过滤器[23](Garbled Bloom Filters，GBF)可以由字符串数组 $GBF[\tau]$ 构建，将集合 $S = \{(x_1, y_1), (x_2, y_2), \cdots, (x_\eta, y_\eta)\}$ 插入 $GBF[\tau]$，需要使用到一组哈希函数 $H = \{h_1, h_2, \cdots, h_t | h_i: \{0,1\}^* \to [\tau]\}$，构建细节参见文献[23]，$(x,y)$ 在布隆过滤器中以 $y = \sum_{i=1}^{t} GBF[h_i(x)]$ 的形式存在。键值对打包技术主要有 $pack()$ 和 $unpack()$ 方法，以混淆布隆过滤器打包键值对为例：

1) $pack(S) \to \Pi$：将字符串数组 $GBF[\tau]$ 初始化为 $\perp$，对于点 $(x_i, y_i) \in S$ 计算出插入位置为空的索引集合 $U = \{h_i(x) | GBF[h_i(x)] = \perp, i \in [t]\}$，选取使得 $y = \sum_{i=1}^{t} GBF[h_i(x)]$ 成立的随机数赋值给 $\{GBF[j], j \in U\}$。所有键值对插入完成后对 $GBF[\tau]$ 中依然为空的位置填入随机数，Π 在该结构中指最终产生的字符串数组 $GBF[\tau]$。

2) $unpack(\Pi, x) \to v$：计算 $v = \sum_{i=1}^{t} GBF[h_i(x)]$ 即可得到解包结果。

(2)基于多项式插值的键值对打包方法[14]

该方法将集合 $S = \{(x_1, y_1), (x_2, y_2), \cdots, (x_\eta, y_\eta)\}$ 视为点集，可构造成一个唯一的 $(\eta-1)$ 次的多项式。$pack(S) \to \Pi$ 表示以点 $(x_1, y_1), (x_2, y_2), \cdots, (x_\eta, y_\eta)$ 构建多项式 Π。$unpack(\Pi, x) \to v$ 表示将 x 代入多项式 Π，计算结果为 v。与基于混淆布隆过滤器的键值对打包方法相同，由多项式插值打包键值对可使得接收多项式的一方无法判断多项式是否被插入了某一键值对，多项式在传输时仅需传输系数，通信量较小，但插值高次多项式非常昂贵，适用于 η 不大的情况。

3　云环境多方隐私集合计算解决方案

3.1　问题描述

基于云服务器辅助的多方隐私集合交集的实际场景可以假定为参与方 $P_1, P_2, \cdots, P_n$ 分别持有集合 $X_1, X_2, \cdots, X_n$，任意选择其中一个参与方作为组织方，为简化方案描述，本文假设选定的组织方为 P_n 且各集合 X_i 中含有的元素个数相同，即 X_i 中都存在 m 个元素 $\{x_{i,1}, x_{i,2}, \cdots, x_{i,m}\}$，$P_1, P_2, \cdots, P_n$ 希望通过云服务器辅助计算出隐私集合的交集 $X_1 \cap X_2 \cap \cdots \cap X_n$，而又不希望泄露除交集以外的任何隐私信息给其他参与方，且不能泄露任何信息给不可信云。假定各参与方不与不可信云合谋，本节将在半诚实模型下对该实际场景提出具体解决方案，方案系统模型如图 1 所示。1)参与方 $P_1, P_2, \cdots, P_n$ 将各自持有的私有集合经加密操作后发送给不可信云服务器；2)云服务器在无法得到任何信息的情况下计算出集合

交集结果，并将其发送给组织方P_n；3）由P_n解出交集并向其他参与方公布交集集合。

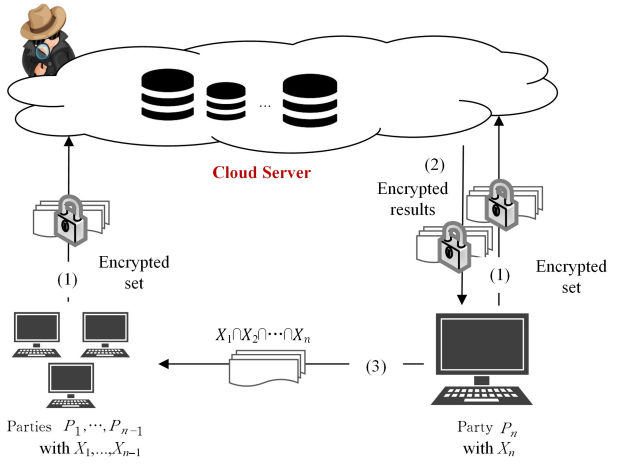


图1 系统模型

Fig. 1 System model

3.2 协议设计

为帮助读者理解本协议，先介绍P_n仅有一个元素x且无云服务器辅助时的基本解决思路。现假定P_n仅有元素x，而$P_1, P_2, \cdots, P_{n-1}$分别持有集合$X_1, X_2, \cdots, X_{n-1}$。$P_n$随机生成$n-1$个随机数$r_1, r_2, \cdots, r_{n-1}$并分别发送给$P_1, P_2, \cdots, P_{n-1}$，随后$P_n$与各参与方$P_i \in P_1, P_2, \cdots, P_{n-1}$执行OPRF协议，$P_n$输入元素$x$得到OPRF结果$t_i$，$P_i$得到OPRF密钥$k_i$。参与方$P_i \in P_1, P_2, \cdots, P_{n-1}$计算键值对集合$S_i = \{(F(k_i, x_{i,1}), r_i), (F(k_i, x_{i,2}), r_i), \cdots, (F(k_i, x_{i,m}), r_i)\}$并使用打包方法得到$\Pi_i \leftarrow pack(S_i)$，将$\Pi_i$发送给$P_n$。$P_n$计算$unpack(\Pi_i, t_i) \rightarrow v_i$可以得到$\{v_1, v_2, \cdots, v_{n-1}\}$。根据键值对打包方法和OPRF的正确性可知，若集合S_i中存在键值对$(F(k_i, x), r_i)$，则有$unpack(\Pi_i, t_i) = r_i$。于是，若$r_1 \oplus r_2 \oplus \cdots \oplus r_{n-1} = v_1 \oplus v_2 \oplus \cdots \oplus v_{n-1}$，则元素$x$为$P_1, P_2, \cdots, P_n$所共有。

上述方案虽然可以通过执行m次来计算出$P_1, P_2, \cdots, P_n$的交集，但P_n能通过对比v_i与r_i得知P_i是否持有元素x，不符合多方隐私交集的计算预期。并且，P_n的通信量和计算量过高，不满足现实应用场景。通过云服务器辅助计算可解决以上两个问题，若将P_n需要执行的OPRF和$unpack()$外包给云服务器，则P_n不再需要与$P_1, P_2, \cdots, P_{n-1}$反复通信，$v_1 \oplus v_2 \oplus \cdots \oplus v_{n-1}$由云服务器计算后再发送给$P_n$，可使得$P_n$仅能判断$x$是否为共有元素，无法得到更多信息。将$P_n$的集合$X_n$经异或秘密共享后分别发送给不可信云$S$和$P_1, P_2, \cdots, P_{n-1}$，可以满足将计算外包又不泄露隐私集合数据。此时，$P_1, P_2, \cdots, P_{n-1}$每轮都需要将数量为$|X_i|$的键值对进行打包，若使用多项式插值打包键值对，则$P_i$需要生成$m$个阶数为$|X_i|-1$的高阶多项式；若引入布谷鸟哈希技术和朴素哈希技术在行与行之间执行上述协议，可使得P_i仅需生成β个阶数为η的多项式。其中，β为布谷鸟哈希表长度，η为朴素哈希表每行最多能容纳元素的个数。本文完整协议的设计如下。

协议1 基于云服务器辅助的多方隐私集合交集计算协议

参与者：参与方$P_1, P_2, \cdots, P_{n-1}$，组织方$P_n$，云服务器$S$

输入：$P_1, P_2, \cdots, P_n$分别输入$X_1, X_2, \cdots, X_n$

输出：$X_1 \cap X_2 \cap \cdots \cap X_n$

（1）组织方P_n通过布谷鸟哈希算法将集合X_n映射至布谷鸟哈希表$B[\beta]$，随后选取$n-1$个随机数种子$seed_1, seed_2, \cdots, seed_{n-1}$分别发送给$P_1, P_2, \cdots, P_{n-1}$。

（2）参与方$P_i \in P_1, P_2, \cdots, P_{n-1}$计算$R_i = \{r_{i,1}, r_{i,2}, \cdots, r_{i,\beta}\} \leftarrow PRG(seed_i)$，组织方$P_n$计算$R_n = \{r_{n,1}, r_{n,2}, \cdots, r_{n,\beta}\}$，其中$r_{n,j} = r_{1,j} \oplus r_{2,j} \oplus \cdots \oplus r_{n-1,j}$。

（3）参与方$P_i \in P_1, P_2, \cdots, P_{n-1}$通过朴素哈希映射算法将集合$X_i$分别存入朴素哈希表$B_i[\beta]$中，$B_i[b]$表示存入参与方$P_i$的朴素哈希表$B_i[\beta]$中第$b$行的所有元素，$x_{i,b,l}$表示$B_i[b]$的第$l$个元素。

（4）组织方P_n随机生成秘密共享种子$share1$，对于$B[\beta]$中第j行的元素$x_{n,j}$，通过种子$share1$生成随机数$x^1_{n,j}$，计算$x^2_{n,j} = x_{n,j} \oplus x^1_{n,j}$。将共享值集合$Share2 = \{x^2_{n,1}, x^2_{n,2}, \cdots, x^2_{n,\beta}\}$发送给云服务器$S$，将秘密共享种子$share1$发送给$P_1, P_2, \cdots, P_{n-1}$，并计算出秘密共享集合$Share1 = \{x^1_{n,1}, x^1_{n,2}, \cdots, x^1_{n,\beta}\}$。

（5）云服务器S输入$share2$与参与方$P_i \in P_1, P_2, \cdots, P_{n-1}$执行OPRF协议，对于所有$b \in \beta$，云服务器$S$得到OPRF结果$t_b$，$P_i$得到OPRF密钥$k_{i,b}$。

（6）各参与方$P_i \in P_1, P_2, \cdots, P_{n-1}$对所有$b \in \beta$，计算$x_{i,b,l} \in B_i[b]$的OPRF结果$u_{i,b,l} = F(k_{i,b}, x_{i,b,l} \oplus x^1_{n,b})$，并与随机数$r_{i,b}$组成键值对集合$S_{i,b} = \{(H(u_{i,b,1}), r_{i,b}), (H(u_{i,b,2}), r_{i,b}), \cdots\}$，将键值对打包结果$\Pi_{i,b} \leftarrow pack(S_{i,b})$发送至云服务器$S$。

（7）云服务器S计算$v_{i,b} = unpack(\Pi_{i,b}, t_b)$，可得到集合$V_1, V_2, \cdots, V_{n-1}$，其中$V_i = \{v_{i,1}, v_{i,2}, \cdots, v_{i,\beta}\}$。随后逐行计算$V_n = V_1 \oplus V_2 \oplus \cdots \oplus V_{n-1}$，将$V_n = \{v_{n,1}, v_{n,2}, \cdots, v_{n,\beta}\}$发送给组织方$P_n$。

组织方P_n解出$O = \{b \mid v_{n,b} = r_{n,b}, r_{n,b} \in R_n, v_{n,b} \in V_n\}$，$X_1 \cap X_2 \cdots \cap X_n = \{B[\beta] \mid b \in O\}$，由组织方$P_n$向$P_1, P_2, \cdots, P_{n-1}$公开集合交集。

3.3 正确性分析

有元素$x \in X_n$，经布谷鸟哈希映射存储于$B[h_a(x)]$，现假定$P_1, P_2, \cdots, P_{n-1}$均持有元素$y = x$，于是$h_a(x) = h_a(y)$，根据朴素哈希表的映射规则，$B_i[h_a(y)]$中必然存有元素$y$。后续仅分析第$h_a(y)$行，$P_n$任取随机数$share1$并执行$share2 = x \oplus share1$，将$share2$发送至云服务器$S$，$share1$发送至$P_1, P_2, \cdots, P_{n-1}$，并选取随机数$r_1, r_2, \cdots, r_{n-1}$分别发送至$P_1, P_2, \cdots, P_{n-1}$，云服务器$S$输入$share2$与参与方$P_i \in P_1, P_2, \cdots, P_{n-1}$执行OPRF协议，$S$得到OPRF结果$t_i$，$P_i$得到OPRF密钥$k_i = t_i \oplus [C(share2) \wedge s]$。$P_i$计算$u_i = F(k_i, y \oplus share1)$。此时，存在关系：$u_i = k_i \oplus [C(y \oplus share1) \wedge s]$，若$y = x$，则$y \oplus share1 = share2$，$u_i = t_i$；若$y \neq x$，则$y \oplus share1 \neq$

$share2$，$u_i \neq t_i$。P_i打包键值对$(H(u_i), r_i)$至Π_i，将结果Π_i发送至云服务器S。云服务器S计算$v_i = unpack(\Pi_i, H(t_i))$；当$u_i = t_i$时，$v_i = r_i$，当$u_i \neq t_i$时，$v_i$为无信息附带的随机数。云服务器$S$异或解出结果值$v = v_1 \oplus v_2 \oplus \cdots \oplus v_{n-1}$，并将结果值$v$发送给组织方$P_n$。显然，在$P_1, P_2, \cdots, P_{n-1}$均持有元素$y = x$的假设下，$v = r_1 \oplus r_2 \oplus \cdots \oplus r_{n-1}$成立。若存在任意$P_i$无元素$x$，则$u_i \neq t_i$，于是$v_i \neq r_i$，$v \neq r_1 \oplus r_2 \oplus \cdots \oplus r_{n-1}$。

3.4　安全性分析

定理 1　以秘密共享的安全性和 OPRF 的不可区分性为基础，协议 1 在半诚实模型下保密地完成了多方隐私集合交集的计算。

证明：分别模拟半诚实参与方$P_i \in P_1, P_2, \cdots, P_n$和半诚实云服务器$S$，以证明在半诚实模型下协议 1 能够在保护$P_1, P_2, \cdots, P_n$数据隐私的基础上计算集合交集。

(1)半诚实参与方P_n

组织方P_n的视图$VIEW^{\pi}_{P_n}(\overline{X}) = \{X_n, (share, seed_1, seed_2, \cdots, seed_{n-1}), (v_{n,1}, v_{n,2}, \cdots, v_{n,\beta})\}$，模拟器$Sim_{P_n}$按照协议 1 模拟云服务器$S$和参与方$P_i \in P_1, P_2, \cdots, P_{n-1}$与组织方$P_n$交互。

Sim_{P_n}随机生成$P_i \in P_1, P_2, \cdots, P_{n-1}$的模拟数据集$X_i'$，并按照朴素哈希映射法将集合元素存入朴素哈希表$B_i'[\beta]$，使其中每一行$b \in \beta$，$Sim_{P_n}$模拟执行 OPRF 协议，得到 OPRF 结果$t_b'$和 OPRF 密钥$k'_{i,b}$，计算$u'_{i,b,l} = F(k'_{i,b}, x'_{i,b,l} \oplus x^1_{n,b})$，生成键值对集合$S'_{i,b} = \{(H(u'_{i,b,l}), r_{i,b}), (H(u'_{i,b,2}), r_{i,b}), \cdots\}$，得到$\Pi'_{i,b} \leftarrow pack(S'_{i,b})$，计算$v'_{i,b} = unpack(\Pi'_{i,b}, t_b')$，得到集合$V_1', V_2', \cdots, V'_{n-1}$，其中$V_i' = \{v'_{i,1}, v'_{i,2}, \cdots, v'_{i,\beta}\}$。随后逐行计算$V_n' = V_1' \oplus V_2' \oplus \cdots \oplus V'_{n-1}$，$Sim_{P_n}$输出$V_n = \{v'_{n,1}, v'_{n,2}, \cdots, v'_{n,\beta}\}$。

X_i'与真实集合X_i均为等长任选集合且 OPRF 协议结果存在伪随机性。因此，在上述过程中无法区分$u'_{i,b,l} = F(k'_{i,b}, x'_{i,b,l} \oplus x^1_{n,b})$与$u_{i,b,l} = F(k_{i,b}, x_{i,b,l} \oplus x^1_{n,b})$，无法区分$S'_{i,b}$与$S_{i,b}$、$\Pi'_{i,b}$与$\Pi_{i,b}$、$v'_{i,b}$与$v_{i,b}$，最终可得$V_n'$与$V_n$不可区分。即$Sim_{P_n}(X_n, f_{|\cap|}) \overset{c}{\equiv} VIEW^{\pi}_{P_n}(\overline{X})$成立。

(2)半诚实参与方$P_i \in P_1, P_2, \cdots, P_{n-1}$

参与方$P_i \in P_1, P_2, \cdots, P_{n-1}$视图$VIEW^{\pi}_{P_n}(\overline{X}) = \{X_i, \perp, seed_i, share, \{k_{i,1}, k_{i,2}, \cdots, k_{i,\beta}\}, f_{|\cap|}\}$，模拟器$Sim_{P_i}$按照协议 1 模拟云服务器$S$和参与方$P_n$与$P_i$进行交互。按照协议 1 的要求，$seed_i'$和$share'$的值由模拟器$Sim_{P_i}$随机生成并发送给$P_i$，因此参与方$P_i$无法区分$seed_i'$与$seed_i$，无法区分$share'$与$share$。而$\{k'_{i,1}, k'_{i,2}, \cdots, k'_{i,\beta}\}$为 OPRF 密钥值，根据 OPRF 对参与双方均有不可区分性，因此$\{k'_{i,1}, k'_{i,2}, \cdots, k'_{i,\beta}\}$与$\{k_{i,1}, k_{i,2}, \cdots, k_{i,\beta}\}$不可区分，不同数据集执行协议 1 得出的交集结果不同，结果集$f_{|\cap|}$与$f'_{|\cap|}$不可区分。因此，$Sim_{P_i}(X_i, f_{|\cap|}) \overset{c}{\equiv} VIEW^{\pi}_{P_i}(\overline{X})$成立。

(3)半诚实云服务器S

云服务器S视图$VIEW^{\pi}_{S}(\overline{X}) = \{\perp, \perp, Share2, \{t_b \mid b \in \beta\}, \{\Pi(i,b) \mid i \in [n-1], b \in \beta\}\}$，模拟器$Sim_s$按照协议 1 模拟参与方$P_i \in P_1, P_2, \cdots, P_n$与云服务器$S$进行交互。模拟器$Sim_s$随机生成集合$X_n'$，并产生秘密共享集合$Share2'$，其为$Sim_s$随机生成的集合$X_n'$秘密共享的结果，故云服务器$S$不可区分$Share2'$与真实$Share2$。在证明$Sim_{P_n}(X_n, f_{|\cap|}) \overset{c}{\equiv} VIEW^{\pi}_{P_n}(\overline{X})$成立的过程中证明了$\Pi'_{i,b}$与$\Pi_{i,b}$是不可区分的。同理，模拟器$Sim_s$产生的$\Pi'_{i,b}$与真实打包结果$\Pi_{i,b}$也是不可区分的。根据$Share2'$与真实$Share2$不可区分以及 OPRF 的不可区分性，$P_i \in P_1, P_2, \cdots, P_{n-1}$与云服务器$S$生成的 OPRF 结果$t'_{i,b}$与$t_{i,b}$也满足不可区分性。因此，$Sim_s(\perp, \perp) \overset{c}{\equiv} VIEW^{\pi}_{S}(\overline{X})$成立。

综上，协议 1 在半诚实模型下能够保证各参与方在不泄露任何集合信息的前提下计算出集合交集。

4　实施与分析

4.1　实验环境与实施参数

实验过程中参与方$P_i \in \{P_1, P_2, \cdots, P_n\}$和云服务器$S$均选用配置为 Intel i7-8750H 2.20 GHz CPU，16 GB RAM 的 Ubuntu 18.04 服务器执行协议。为提升协议实施性能，在实施过程中使用到了文献[12]设计的 OPRF 技术源码和 NTL 库[24]提供的高效多项式操作。实验中，各方持有集合中的元素为 128 bit 随机串，安全参数$\kappa = 128$，$\lambda = 40$。根据文献[5]，本文按照布谷鸟哈希映射算法的步骤选取 3 个不同的哈希函数，将组织方集合X_n中的元素映射至容量为$\beta = 1.5\,\mathrm{m}$的布谷鸟哈希表。按照文献[23]，本文构造的混淆布隆过滤器的大小为$58N$，选取 31 个哈希函数计算元素存储位置，N表示待插入元素的数量，失败概率为$(1 - e^{\frac{-31}{58}})^{31}$，接近于$2^{-\lambda}$。

4.2　性能分析与对比

本文分别对各参与方持有元素数量为$\{2^{14}, 2^{16}, 2^{18}\}$的 3 种情况进行了仿真实验测试。在实施过程中，参与方P_n的计算量极小，主要执行布谷鸟哈希映射和秘密共享，通信量主要包括分发秘密共享结果、接收云服务器的计算结果和向其他参与方公布的交集结果。参与方$P_i \in P_1, P_2, \cdots, P_{n-1}$的计算量主要包括构建朴素哈希映射、OPRF 和键值对$pack()$，通信量主要包括 OPRF、发送$pack()$结果Π和接收交集结果。云服务器S的计算量主要包括分别与$P_i \in P_1, P_2, \cdots, P_{n-1}$生成 OPRF 实例和大量$unpack()$操作的总和。总耗时为参与方并行计算环境下协议执行完毕所需的时间，总通信量则表示协议执行过程中实际使用信道传输数据量的总和。基于混淆布隆过滤器(GBF)和多项式插值(Poly)构造的两种代表性的键值对打包方法实施本文协议，方案总通信量和计算耗时与参与方人数呈线性递增，当参与方人数$n = 32$时，协议各参与方的性能表现如表 1 所列。当各参与方集合大小为2^{16}时，参与方人数对协议总计算时间和总通信开销的影响如表 2 所列。在实际实施过程中，参与方$P_1, P_2, \cdots, P_{n-1}$的计算并行进行，参与方$P_i \in P_1, P_2, \cdots, P_{n-1}$计算环节的总耗时与耗时最久的参与方的运行时间一致，表 1 选取耗时最长的参与方

P_i进行统计，表 2 中运行时长指$P_1, P_2, \cdots, P_{n-1}$并行环境下协议实施从开始到得到交集所用的总时间。

表 1 协议各参与方的性能评测

Table 1 Performance evaluation of our protocol

Parameters		Running Time/s			Communication Cost/MB		
Set size		2^{14}	2^{16}	2^{18}	2^{14}	2^{16}	2^{18}
P_n		0.003	0.013	0.063	8.61	34	136.00
S	Poly	**4.216**	**16.398**	**73.300**	**53.15**	**212.56**	**850.55**
	GBF	4.540	17.369	73.532	496.76	1987.31	7949.55
P_i	Poly	6.760	29.677	120.578	**1.94**	**7.76**	**31.05**
	GBF	**0.069**	**0.273**	**1.178**	16.25	65.01	260.05
Total	Poly	10.980	46.089	193.941	**61.00**	**243.56**	**974.55**
	GBF	**4.613**	**17.656**	**74.773**	504.61	2018.31	8073.55

表 2 不同参与人数下协议性能对比

Table 2 Performance comparison of our protocols with different number of parties

Number Parties n		2	4	8	16	32
Running Time/s	Poly	29.135	30.196	32.320	36.566	46.089
	GBF	**0.863**	**2.021**	**4.337**	**8.970**	**17.656**
Communication Cost/MB	Poly	**10.76**	**26.28**	**57.32**	**119.40**	**243.56**
	GBF	68.01	198.03	458.07	978.15	2018.31

参与方P_n作为组织方将原本需要执行的大部分计算都外包给了不可信云服务，在不考虑云服务器计算与通信能力的前提下，应该以参与方P_i的通信能力和计算能力来选择方案中键值对的打包方法，若P_i处于网络宽松但算力不足的情况下，应选择使用混淆布隆过滤器打包键值对；若P_i处于通信紧张但算力充足的情况下，则可选用多项式插值打包键值对。

现有适用于云平台的多方隐私交集协议较少，文献[9]提出了一种适用于云环境下解决多方隐私集合计算问题的方案，但该方案对输入域存在限制，输入域过大时存在计算问题和安全隐患，对云服务器数量和参与方人数也有一定限制。文献[9]与本文方案的对比结果如表 3 所列。

表 3 云环境下多方 PSI 计算协议对比

Table 3 Comparison of multi-party PSI protocols with cloud aided

协议	输入域限制	云服务器数量	参与方数量
文献[9]	yes	大于等于 3	大于等于 3
Ours	no	大于等于 1	大于等于 2

结束语 隐私交集计算是安全多方计算协议的基础模块，并已应用于现实众多领域中。本文设计了一种适用于云计算环境的多方隐私交集计算协议，使得持有隐私集合的多方在不泄露任何集合信息的前提下，利用不可信云服务器的计算能力和通信能力辅助参与方完成隐私集合交集计算，并选用基于不经意传输拓展协议的 OPRF 技术、键值对打包技术和秘密共享技术设计隐私集合交集协议，配合使用布谷鸟哈希技术和朴素哈希映射技术以大幅提升协议性能，与现有方案相比，本文方案使用限制更少，有更广的应用范围。本文的下一步工作将研究如何把参与方的数据安全地外包给不可信云，以实现处理后的隐私数据重复利用。

参 考 文 献

[1] SHEN L Y, CHENG X J, SHI J Q, et al. A review of the research on privacy protection set intersection computing technology[J]. Computer Research and Development, 2017, 54(10): 2153-2169.

[2] CUI H R, LIU T Y, YU Y. Overview of the development status of set intersection computing protocol with privacy protection [J]. Information Security and Communication Confidentiality, 2019(3): 48-67.

[3] YUNG M. From mental poker to core business: Why and how to deploy secure computation protocols? [C]// Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. New York: Association for Computing Machinery, 2015: 1-2.

[4] AGGARWAL C C, YU P S. Privacy-preserving data mining: models and algorithms [M]. Springer Science & Business Media, 2008.

[5] DEMMLER D, RINDAL P, ROSULEK M, et al. PIR-PSI: Scaling Private Contact Discovery[J]. Proceedings on Privacy Enhancing Technologies, 2018(4): 159-178.

[6] DUONG T, PHAN D H, TRIEU N. Catalic: Delegated PSI Cardinality with Applications to Contact Tracing[C]// International Conference on the Theory and Application of Cryptology and Information Security. Cham: Springer, 2020: 870-899.

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

[8] ZHANG E, LIU F H, LAI Q, et al. Efficient Multi-Party Private Set Intersection Against Malicious Adversaries[C]// Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop. New York: Association for Computing Machinery, 2019: 93-104.

[9] LI S D, ZHOU S F, GUO Y M, et al. Collective Privacy Computing in Cloud Environment [J]. Journal of Software, 2016, 27(6): 1549-1565.

[10] FREEDMAN M J,NISSIM K,PINKAS B. Efficient private matching and set intersection[C]// International Conference on the Theory and Applications of Cryptographic Techniques. Berlin,Heidelberg:Springer,2004:1-19.

[11] PINKAS B,SCHNEIDER T,ZOHNER M. Faster private set intersection based on OT extension[C]// Proceedings of the 23rd USENIX Security Symposium. San Diego:{USENIX} Association,2014:797-812.

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

[13] PINKAS B,SCHNEIDER T,SEGEV G,et al. Phasing:Private set intersection using permutation-based hashing[C]// Proceedings of the 24th USENIX Security Symposium USENIX Association. 2015:515-530.

[14] PINKAS B,ROSULEK M,TRIEU N,et al. Spot-light:Lightweight private set intersection from sparse ot extension[C]// Annual International Cryptology Conference. Cham:Springer,2019:401-431.

[15] CHEN Z H,LI S D,HUANG Q,et al. Non-encrypted method securely calculates two sets of relations[J]. Journal of Software,2018,29(2):473-482.

[16] SONG X F,GAI M,ZHAO S N,et al. Privacy protection statistical protocol for ensemble computing[J]. Computer Research and Development,2020,57(10):2221-2231.

[17] DOU J W,LIU X H,WANG W L. Efficient and secure calculation of two-party sets in the field of rational numbers[J]. Chinese Journal of Computers,2020,43(8):1397-1413.

[18] ABADI A,TERZIS S,DONG C. O-PSI:delegated private set intersection on outsourced datasets[C]// IFIP International Information Security and Privacy Conference. Cham:Springer,2015:3-17.

[19] TAJIMA A,SATO H,YAMANA H. Outsourced private set intersection cardinality with fully homomorphic encryption[C]// 2018 6th International Conference on Multimedia Computing and Systems (ICMCS). IEEE,2018:1-8.

[20] 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.

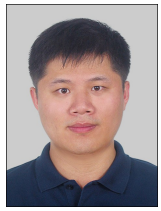
[21] GOLDREICH O. Foundations of cryptography:volume 2,basic applications[M]. Cambridge University Press,2009.

[22] SCHNEIER B. Applied cryptography:protocols,algorithms,and source code in C[M]. John Wiley & Sons,2007.

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

[24] VICTOR S. NTL:A Library for doing Number Theory[EB/QL]. https://libntl.org/.

WANG Qin,born in 1996,master candidate,is a student member of China Computer Federation. His main research interests include information security and secure computation.



WEI Li-fei,born in 1982,Ph.D,associate professor,master supervisor,is a senior member of China Computer Federation. His main research interests include information security,privacy preserving and cryptography.