低带宽场景下防合谋多方隐私集合并集协议

张恩[1,2]，王梦涛[1]，郑东[1,3]，禹勇[4]，黄昱晨[1]
（1.河南师范大学计算机与信息工程学院，河南 新乡 453007；2.河南省教育人工智能与个性化学习重点实验室，河南 新乡 453007；
3.西安邮电大学无线网络安全技术国家工程实验室，陕西 西安 710061；4.陕西师范大学计算机科学学院，陕西 西安 710062）

摘 要：针对现存的多方隐私集合并集（MPSU）协议存在交互轮数多以及通信开销大等问题，使其无法在低带宽场景中得以有效应用，设计了一种基于不经意键值存储和门限同态加密技术的不经意匹配置换方法，并运用该方法提出了一种半诚实模型下的多方隐私集合并集协议。该协议允许N个参与方共同计算所有集合的并集，且不会泄露任何其他的信息，具有通信轮数少、能抵御$N-1$个参与方的合谋、通信开销低等优势，比现有最先进的多方隐私集合并集的通信开销降低了65%左右。

关键词：多方隐私集合并集；低带宽场景；不经意键值存储；半诚实模型；门限同态加密
中图分类号：TP309
文献标志码：A
DOI：10.11959/j.issn.1000−436x.2025020

Anti-collusion multi-party private set union protocol in low-bandwidth scenarios

ZHANG En[1,2], WANG Mengtao[1], ZHENG Dong[1,3], YU Yong[4], HUANG Yuchen[1]
1. College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China
2. Key Laboratory of Artificial Intelligence and Personalized Learning in Education of Henan Province, Xinxiang 453007, China
3. National Engineering Laboratory of Wireless Network Security Technology, Xi'an University of Posts and Telecommunications, Xi'an 710061, China
4. College of Computer Science, Shaanxi Normal University, Xi'an 710062, China

Abstract: Aiming at the problems of the existing multi-party private set union (MPSU) protocols, such as a large number of interaction rounds and excessive communication overhead, which prevented them from being effectively applied in low-bandwidth scenarios, an oblivious matching permutation method based on oblivious key-value store and threshold homomorphic encryption technologies was designed, and a multi-party private set union protocol under a semi-honest model was proposed through this method. This protocol allowed N participants to jointly calculate the union of all sets and would not leak any other information. It mainly has the advantages of a small number of communication rounds, the ability to resist the collusion of $N-1$ participants, and low communication overhead. Its communication overhead is reduced by about 65% compared to the existing most advanced multi-party private set union.

Keywords: multi-party private set union, low-bandwidth scenario, oblivious key-value store, semi-honest model, threshold homomorphic encryption

0 引言

在当今数字化时代，隐私计算技术作为保障数据安全流通的有效方式，已逐渐成为促进数据要素跨域流通和应用的核心技术。多方隐私集合并集（MPSU, multi-party private set union）作为隐私计算的关键组成部分，为诸多实际应用场景提供了不

收稿日期：2024–09–03；修回日期：2025–01–06
通信作者：张恩，zhangenzdrj@163.com
基金项目：国家自然科学基金资助项目（No.62372157）
Foundation Item：The National Natural Science Foundation of China (No.62372157)

可或缺的技术支撑。多方隐私集合并集协议是指 N 个参与方 $P_1,\cdots,P_N$ 各自持有一个私有集合 $X_i, i \in [1,N]$，希望获得所有集合的并集 $\bigcup_{i=1}^{N} X_i$，且不泄露其他任何信息，其功能如图1所示。该协议在网络风险评估[1-2]、隐私保护数据聚合[3]和Private-ID[4]等场景下被广泛应用。例如，在网络安全领域，信号较差且通信带宽较低的情况下，多个企业为了降低其基础设施中的潜在漏洞，希望协同计算所有的IP黑名单。企业的IP黑名单都源于其私有的检测策略，以明文形式交换IP黑名单会导致恶意企业通过分析相同的IP地址推断其他企业的检测策略，从而通过规避检测策略来发起攻击。此外，在医疗健康领域MPSU协议也得到了广泛的应用。例如，一个社会服务组织需要确定哪些癌症患者有资格享受社会福利，其需要来自多家医院的患者数据来获得所有癌症患者的集合，然后确定哪些人有资格享受社会福利。该组织可以使用MPSU协议安全地计算来自各家医院的所有癌症患者的并集，而不需要向其他方透露任何原始数据。随后，该组织可以通过对获得的集合与有资格享受社会福利的患者进行隐私集合交集（PSI, private set intersection）计算来获得最终结果，同时保护患者的隐私。

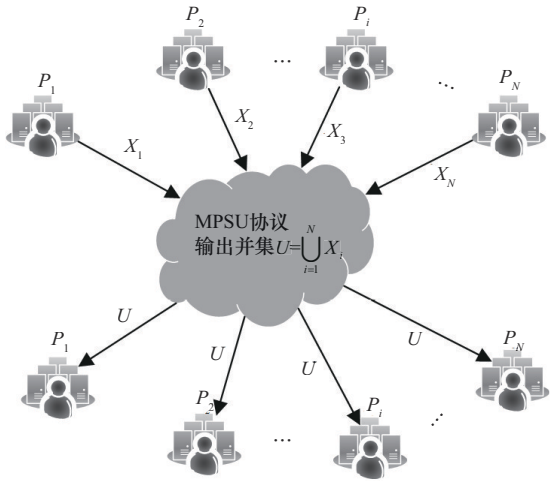

图1　MPSU的功能

近年来，隐私集合操作技术飞速发展，特别是隐私集合交集。目前两方隐私集合交集协议[5-15]中除了使用少数不经意传输（OT, oblivious transfer）[16-17]操作外，主要依赖于对称密钥操作，实现了高效的性能。同时随着零共享等技术的提出，多方隐私集合交集协议[18-26]在高效性方面取得了显著成就。然而，在过去的十年里，隐私集合并集（PSU, private set union）协议却很少受人关注，导致现有的PSU协议[27-38]大多只适用于两方场景下安全的计算集合的并集。在多方隐私集合并集协议[39-45]中，随着参与方数量或数据集规模的增加，其通信开销呈指数级增长，严重限制了其在大规模场景中的应用，尤其是在低带宽场景下。本文旨在解决多方隐私集合并集通信开销较大的问题，通过引入新的密码学原语和优化算法，在保证数据隐私的前提下，降低通信复杂度，提高协议的执行效率，以克服现有技术在低带宽场景下的局限性。本文的主要贡献包括3个方面。

1) 设计了一种新的不经意匹配置换方法，并基于该方法提出了一种多方隐私集合并集协议，解决了在多方场景下计算隐私集合并集隐私泄露的问题，且能防止 $N-1$ 方合谋。

2) 在本文协议中，以牺牲少量计算开销为代价，有效地降低了协议的通信开销，适用于带宽受限的场景。与此同时，该协议的离线阶段耗时较短，且减少了在线时间的通信轮数。

3) 在三方且集合大小为 $n=2^{20}$ 的情况下，本文协议通信开销为357.8 MB，在带宽为200 Mbit的情况下，运行时间为27.033 s；文献[44]协议的通信开销为2 655 MB，在带宽为200 Mbit的情况下，运行时间为61.595 s。在十方且集合大小为 $n=2^{16}$ 的情况下，本文协议通信开销为557.035 MB，在带宽为200 Mbit的情况下，运行时间为19.803 5 s；文献[44]协议的通信开销为2 627.3 MB，在带宽为200 Mbit的情况下，运行时间为55.168 s。

1　相关工作

现有的PSU协议可分为两大类，一类是使用算术电路，布尔电路或者混淆电路等技术的通用PSU协议[28-29]，此类协议具有较高的可拓展性，适用于各种场景，但在效率方面表现相对较低；另一类是根据特定场景而设计的专有协议，通常具有较高的效率，但可拓展性较差。

Zhang等[3]提出了2种线性计算和通信复杂性的PSU协议。第一种是基于对称密钥加密和通用2PC技术。第二种是基于可重新随机化的公钥加密。第一种接收方首先选择一个字符串 s，使用对

称密钥k将s加密n次得到n个密文c_i，使用不经意键值存储（OKVS, oblivious key-value store）编码将(y_i,c_i)编码为数据结构D并将其发给发送方，发送方执行解码算法$\mathrm{Decode}(D,x_i)$得到n个密文d_i，然后发送方将d_i作为输入，则接收方将s和对称密钥k作为输入调用通用2PC技术，接收方输出向量$\boldsymbol{b}\in\{0,1\}^n$，如果$x_i\in Y$，则接收方获得的$b_i$值为0，否则获得的$b_i$值为1。最后双方调用$n$个OT实例，让接收方得到除自己集合外的非交集元素，获得所有集合的并集。第二种则是将对称加密换成可重新随机化公钥加密，从而避免了使用效率较低的安全多方计算，但上述2种协议仅适用于两方半诚实场景。

Kissner等[39]提出了PSU协议，该协议使用多项式和同态加密技术实现，主要思想是将每个参与方集合的元素插入一个多项式f_i中，多项式的根就是集合中每个元素，此时参与方在半同态下计算多项式累乘$g=\prod_{i=1}^{N}f_i$，对于每个参与方的集合元素x_j^i，都满足等式$\prod_{i=1}^{N}f_i(x_j^i)=0$，参与方在得到多项式累乘$g$后，先执行缩减步骤减少根的度数，再解密计算出多项式的所有根，即可获得所有参与方的并集。由于插值多项式的限制，该协议在多参与方和大规模数据的场景下无法进行高效的计算。

Frikken[40]提出了一种通信复杂度与数据集合大小呈线性关系的PSU协议，该协议的主要思想是接收方使用插值多项式将自己的集合元素$y\in Y$插入多项式f中，然后通过同态加密技术将多项式f加密并发给发送方，发送方得到加密的多项式后，根据自己的集合元素$x\in X$计算$(\mathrm{Enc}(f(x)),x\,\mathrm{Enc}(f(x)))$，并将其发送给接收方。接收方收到消息后将其解密，如果$x\in Y$，则$f(x)=0$，接收方解密得到的结果为$(0,0)$。如果$x\notin Y$，则$f(x)\neq 0$，接收解密计算$y=xf(x)^{-1}f(x)$并得到非交集元素。多次的多项式插值计算会导致同态计算的电路深度增加，使该协议在多参与方和大规模数据的场景下无法进行高效的计算。

Seo等[42]提出了一种基于秘密共享和逆洛朗级数的MPSU协议。其核心思想是，如果2个集合X和Y分别用多项式f_X和f_Y表示，则并集$X\cup Y$可以用f_X和f_Y的最小公倍数表示，记为$\mathrm{lcm}(f_X,f_Y)$。此时，$\frac{1}{f_X}+\frac{1}{f_Y}=\frac{q(x)}{\mathrm{lcm}(f_X,f_Y)}$，因此计算$\frac{1}{f_X}+\frac{1}{f_Y}$就足够了。该协议虽然实现了恒定轮数的通信，但高次多项式的运算会导致计算和通信复杂度较高，在多参与方和大规模数据的场景下无法进行高效的计算。

Gong等[43]提出了一种基于布隆过滤器（BF, Bloom filter）和同态加密的MPSU协议。他们观察到，如果BF没有冲突，那么对存储在其中的每个元素，至少有一个位置仅由其自身映射。利用这一特性，他们首先构造了一个存储并集的BF，然后检查BF中的每个位置是否仅由一个元素映射。如果是这样，他们就可以找出那个元素。由于BF的长度与统计安全参数和并集集合大小有关，因此该协议需要大量的同态计算操作，计算开销太大，在多参与方和大规模数据的场景下无法进行高效的计算。

Liu等[44]提出了一种称为多查询秘密共享私有成员资格测试（mq-ssPMT, multi-query secret-shared private membership test）的新技术，并将该技术和不经意传输结合构造了一种新的高效MPSU协议。该协议可以在多参与方和大规模数据的场景下进行高效的计算。但由于该协议需要进行多轮的交互，通信开销相对较高。

综上所述，PSU在多方的场景下由于计算开销和通信开销较大，在网络带宽较低的情况下，拥有大数据集的多参与方无法高效计算隐私集合并集。如果用现有的两方PSU协议来实现多方隐私集合并集协议，不仅会泄露各个参与方之间的交集信息，同时还会泄露部分参与方的私有元素。例如，现有N个参与方$P_1,\cdots,P_N$，各个参与方的私有集合用$X_i=\{x_1^i,\cdots,x_n^i\}_{i\in[1,N]}$表示，$n$为私有元素的数量。$N-1$个参与方$P_2,\cdots,P_N$与$P_1$通过执行两方PSU协议后输出$N-1$个并集$X'_{i-1}$，$P_1$能通过对比$N-1$个并集获得除自己元素外的$P_2,\cdots,P_N$的交集元素，此时$P_1$还能学习到并集中某些元素是来自哪个参与方的，隐私信息泄露问题如图2所示。

2 预备知识

本文设计的协议主要基于不经意键值存储和同态加密，下面介绍相关概念和基础知识。

2.1　多方隐私集合并集

多方隐私集合并集是指N个参与方$P_{i,i\in[1,N]}$各自拥有大小为n的集合X_i，希望共同计算所有集合的并集，且不会泄露任何其他的信息，其MPSU的理想功能如下。

参数：有N个参与方$P_i, i\in[1,N]$，各自拥有大小为n的集合X_i。

功能（$F_{\text{MPSU}}^{P_1,\cdots,P_N}$）：1) 对于$i\in[1,N]$，等待参与方$P_i$输入集合$X_i=\{x_1^i,\cdots,x_n^i\}$；2) 参与方$P_1$输出并集$U=\bigcup_{i=1}^{n}X_i$。

2.2　安全模型

本文实现的是在半诚实模型下安全的多方隐私集合并集协议，在该模型中所有参与方执行的计算都必须完全遵守协议的要求，不能偏离协议。但是好奇的参与方可以根据执行协议时所获得的信息去推理除结果之外的信息。此外，由于本文协议涉及多个参与方，需考虑合谋情况，这意味着敌手可以腐败多个参与方，并结合他们所得到的信息推断出更多信息。

对于N个参与方，其中腐败的参与方集合为$C=\{i_1,\cdots,i_q\}\subseteq[N]\overset{\text{def}}{=}\{1,\cdots,N\}$，令$F_C(X_1,\cdots,X_N)$为$F_{i_1}(X_1,\cdots,X_N),\cdots,F_{i_q}(X_1,\cdots,X_N)$。对于$C=\{i_1,\cdots,i_q\}$的视图为$\text{View}(\bar{X})=(C,\text{View}_{i_1}^{\Pi}(\bar{X}),\cdots,\text{View}_{i_q}^{\Pi}(\bar{X}))$，其中$\bar{X}=(X_1,\cdots,X_N)$为参与方的输入。

定义 1　设$F:(\{0,1\}^*)^N\to(\{0,1\}^*)^N$是一个确定性函数，令$\Pi$为安全计算$F$的多方协议，如果存在概率多项式时间算法Sim，对于任意的$C\subseteq[N]$满足

$$\left\{\text{Sim}(C,(X_{i_1},\cdots,X_{i_q}),F_C(\bar{X}))\right\}_{\bar{X}\in(\{0,1\}^*)^N}\overset{c}{\equiv}\text{View}_C^{\Pi}(\bar{X})_{\bar{X}\in(\{0,1\}^*)^N}$$

则协议Π在半诚实敌手存在下是安全的。

2.3　门限同态加密

门限全同态加密（TFHE, threshold fully homomorphic encryption）[46-47]是同态加密中的一个特例，通常用于多密钥同态加密方案中，以实现门限解密或门限操作的功能。在门限同态加密系统中，私钥被分割成多个份额，并分发给不同的参与者。只有当达到一定数量（门限值）的参与者合作并提供他们各自的份额时，才能正确地执行解密或特定的同态操作。本文协议仅用到了多密钥这个性质，多密钥的性质是把私钥分割成多个份额私钥并分发给各个参与方，参与方可对密文数据进行分析处理，处理后的结果由所有参与方联合解密，其相对于传统的（单密钥）全同态加密，更加适用于多参

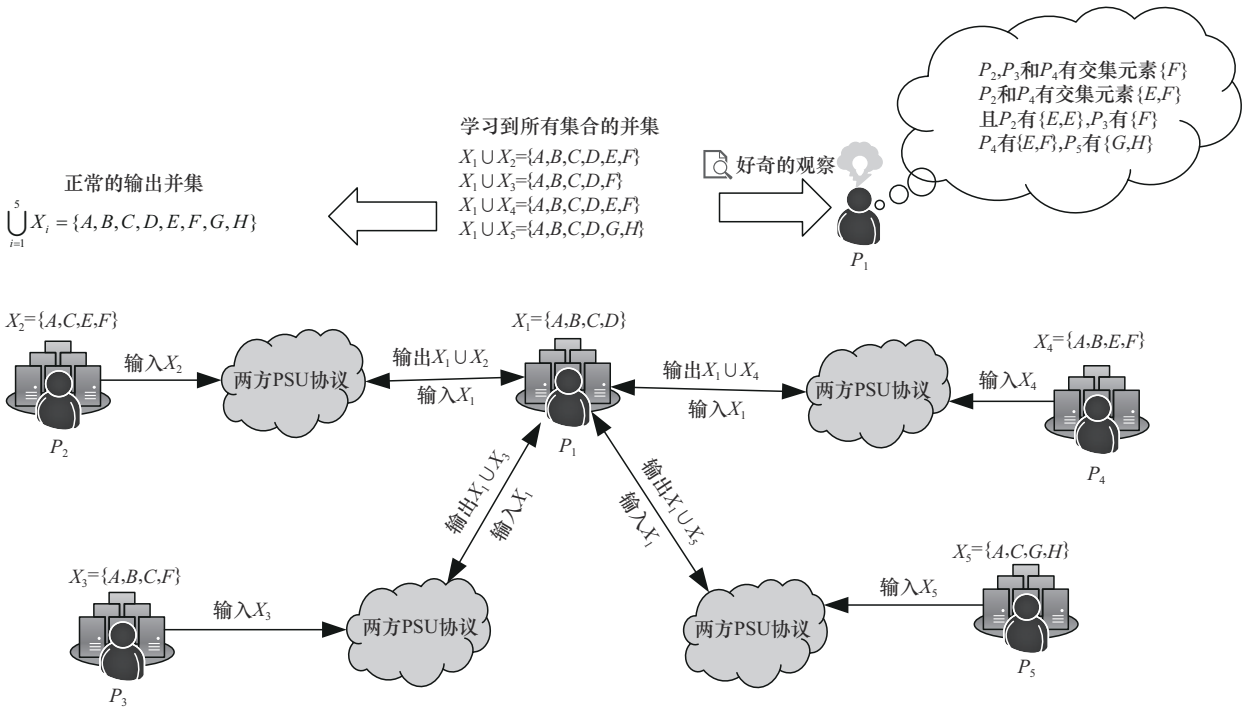

图2　隐私信息泄露问题

与方数据的隐私保护和处理。门限全同态加密与一般的全同态加密不同，一般全同态加密的密钥生成是一个算法，而门限全同态加密的密钥生成是一个N方的协议，下面是根据一些常见设置定义的协议。

TFHE.Keygen(setup)：每一方输入相同公共参数pp。当协议结束时，每一方P_i输出一个公钥pk，一个公共评估密钥evk和一个密钥份额sk_i。

TFHE.Enc(pk,m)：输入公钥pk和明文$m \in M$，输出密文$c \in C$。

TFHE.PartyDec(pk,sk_i,c)：输入密钥份额sk_i、密文$c \in C$，部分解密算法输出部分解密份额c_i'。

TFHE.FinDec(pk,$c_1',\cdots,c_N'$)：输入一个公钥pk和重复执行N次上述步骤得到的N个部分解密的值c_i'，最终解密算法输出一个明文$m \in M$。

TFHE.Eval(pk,f,$c_1,\cdots,c_N$)：输入公钥pk、电路f和一组密文$c_1,\cdots,c_N$，评估算法计算$c^* \leftarrow$TFHE.Eval(pk,f,$c_1,\cdots,c_N$)并输出c^*。

正确性。对于$i \in [1,N]$，首先执行密钥生成协议$(\mathrm{pk},\mathrm{sk}_i) \leftarrow$TFHE.Keygen(setup)，其次对$N$个明文$m_i$进行加密得到$N$个密文$c_i \leftarrow$TFHE.Enc(pk,$m_i$)，最后调用评估算法得到结果$c^* \leftarrow$TFHE.Eval(pk,$f$,$c_1,\cdots,c_N$)都满足TFHE.FinDec(pk,{TFHE.PartyDec(c^*,sk_i)}$_{i\in[1,N]}$)$=f(m_1,\cdots,m_N)$。

同态性。存在同态加法运算，使得给出任意m_1和m_2，任意$c_1 \leftarrow \mathrm{Enc}(m_1)$和$c_2 \leftarrow \mathrm{Enc}(m_2)$，满足FinDec(Enc($m_1$)+Enc($m_2$))=FinDec(Enc($m_1+m_2$))$=m_1+m_2$。

存在同态乘法运算，使得给出任意m_1和m_2，任意$c_1 \leftarrow \mathrm{Enc}(m_1)$和$c_2 \leftarrow \mathrm{Enc}(m_2)$，满足FinDec(Enc($m_1$)Enc($m_2$))=FinDec(Enc($m_1m_2$))$=m_1m_2$。

2.4 不经意键值存储

不经意键值存储[10,13-14]是一种能将一组键K映射到相应的一组值V并且能将这种映射关系隐藏的数据结构。

定义2 键值存储（KVS, key-value stores）由一组K键、一组V值和一组函数H进行参数化，并由$\mathrm{Encode}_H(\cdot)$和$\mathrm{Decode}_H(\cdot)$算法组成。

$\mathrm{Encode}_H(\{(x_1,v_1),\cdots,(x_n,v_n)\})$：将一组$(x_i,v_i)_{i\in[1,n]}$键值对作为该算法的输入，以压倒性的概率输出一个数据结构P，极小的概率出现错误终止。

$\mathrm{Decode}_H(P,y)$：将数据结构P和一个值y作为输入，$\exists i \in [1,n]$，使得$x_i=y$，则输出一个值v_i，否则输出一个随机数。

正确性。对于所有不同的键$A \subseteq K \times V$，有

$$(x,v) \in A \text{ 和 } \perp \neq P \leftarrow \mathrm{Encode}_H(A) \Rightarrow \mathrm{Decode}_H(P,x)=v$$

定义3 对于所有不同的$\{x_1^0,\cdots,x_n^0\}$和所有不同的$\{x_1^1,\cdots,x_n^1\}$，如果Encode对于$(x_1^0,\cdots,x_n^0)$或$(x_1^1,\cdots,x_n^1)$没有输出$\perp$，而输出的$R(x_1^0,\cdots,x_n^0)$与$R(x_1^1,\cdots,x_n^1)$是计算不可区分的，则KVS是一个OKVS，其中

$R(x_1,\cdots,x_n)$:
for $i \in [1,n]$: do $v_i \leftarrow V$
return Encode($\{(x_1,v_1),\cdots,(x_n,v_n)\}$)

3 多方隐私集合并集基础协议

现有最先进的两方PSU协议是Zhang等[3]提出的基于可重新随机化公钥加密的两方隐私集合并集协议，该协议实现的功能为持有一个私有集合Y的接收方R与持有一个私有集合X的发送方S交互，并且接收方R可以获得并集$X \cup Y$。其具体的实现思路为，接收方R选择一个字符串s，使用公钥pk将s加密n次得到n个密文$c_{i,i\in[1,n]}$，并使用OKVS的编码算法Encode将(y_i,c_i)编码为数据结构D并将其发给发送方S。发送方S调用OKVS的解码算法Decode得到n个密文$d_{i,i\in[1,n]}$，然后将$d_{i,i\in[1,n]}$重新随机化后发送给接收方R。接收方R收到$d_{i,i\in[1,n]}$后使用私钥sk解密并判断是否与字符串s相等，如果相等则将该元素对应的标签值b_i设置为0，否则设置为1。最后双方执行不经意传输协议，接收方R得到两方的非交集元素并输出并集。

如果直接将该方案扩展为多方，参与方P_1作为接收方R首先选择一个字符串s，并使用公钥pk将s加密n次得到n个密文$c_j,j \in [1,n]$，其次使用OKVS的编码算法Encode将(x_j^1,c_j)编码为数据结构D_1并将其发送给参与方$P_2,\cdots,P_N$。$P_2,\cdots,P_N$执行解码算法Decode(D_1,x_j^i)得到n个密文$d_j^i,i \in [2,N],j \in [1,n]$，并将$d_j^i$重新随机化后发送给$P_1$。$P_1$解密后对比是否等于字符串$s$，输出$N-1$个向量$\boldsymbol{b}_i$，随后再跟$P_2,\cdots,P_N$执行不经意传输协议得到非交集元素输出所有集合的并集，此时同样会出现如图2中交集信息泄露的问题。假设上述交集信息泄露的问题已经解决，P_1已经获得了$N-1$个无交集的选择向量$\boldsymbol{b}_i$，此时P_1与$P_2,\cdots,P_N$分别调用不

经意传输协议来获取非交集元素，尽管如此P_1也了解到了某个非交集元素是来自哪个参与方的。这种信息的泄露在多方隐私集合并集协议中也是不被允许的，所以本文摒弃了以不经意传输的方法来获取非交集元素。

假设对于$i\in[2,N],i<m\leqslant N$，参与方P_m的集合元素x_j^m与P_i的集合有交集时，给该元素设置一个对应的标签b_j^m，并将该值设置为0，否则将其设置为随机值r。这样P_m就可以计算$(b_j^m,b_j^m x_j^m)$并发送给P_1，如果为交集元素时P_1得到的值被隐藏为$(0,0x_j^m)$。如果直接让P_m拿到明文的b_j^m，P_m就直接得到了交集信息。在此步骤中引入同态加密技术就能完美解决，它不仅能隐藏b_j^m的值，同样可以在密文下计算$(b_j^m,b_j^m x_j^m)$。解决上述问题后，现在的关键问题是P_m如何得到元素对应的b_j^m。在该问题的解决上，本文借鉴了 Zhang 等[3]的基于可重新随机化的公钥加密实现的两方隐私集合并集协议。具体做法是对于$i\in[1,N-1],j\in[1,n]$，P_i使用 OKVS 编码将$(x_j^i,\mathrm{Enc}(\mathrm{pk},0))$编码为数据结构$D_i$并将其发送给$P_m$，其中$i<m\leqslant N$。$P_m$收到数据结构$D_i$并解码得到$d_j^{m,i}=\mathrm{Decode}(\{(D_i,x_j^m)\})$，并计算标签$b_j^m=\prod_{i=1}^{m-1}d_j^{m,i}$。最后对于$i\in[2,N]$，参与方$P_i$计算$\{(b_j^i,b_j^i x_j^i)\}$发送给$P_1$，$P_1$解密后得到所有集合的并集。

此时P_1能了解到某个元素是来从哪个参与方获得的，为此协议中采用的同态加密选用门限同态加密。对于$i\in[2,N]$，参与方P_i计算$\{(b_j^i,b_j^i x_j^i)\}$后先发给P_N，P_N打乱顺序进行部分解密后发给P_{N-1}，对于$i=N-1,N-2,\cdots,2$，P_i打乱键值对集合后再使用私钥sk_i进行部分解密后将其发送给P_{i-1}，最终P_1输出并集集合。该协议的主要思想是对于$m\in[2,N]$，将参与方P_m集合中与参与方$P_1,\cdots,P_{m-1}$集合中的交集元素在密文的情况下置换为0，称该方法为不经意匹配置换，最终将计算后的结果发送给P_1，P_1解密所有的结果即可得出所有集合的并集，即$\bigcup_{i=1}^{N}X_i=X_1\cup(X_2\backslash X_1)\bigcup\cdots(X_N\backslash X_{N-1}\cdots\backslash X_1)$。

3.1　多方隐私集合并集的基础协议构造

下面是多方隐私集合并集的基础协议构造，协议符号说明如表 1 所示，三方 PSU 协议流程如图 3 所示。

表 1　　MPSU 协议符号说明

符号	说明
$P_i,i\in[1,N]$	参与方P_i
$X_i,i\in[1,N]$	参与方P_i的私有数据集
D_i	编码得到的数据结构
$x_j^i,i\in[1,N],j\in[1,n]$	参与方P_i的第j个元素
$d_j^{m,i},i\in[1,N],j\in[1,n]$	P_m用D_i解码元素x_j^i的结果
$\tilde{d}_j^i,i\in[1,N],j\in[1,n]$	P_i重新随机化$d_j^{i,m}$的结果

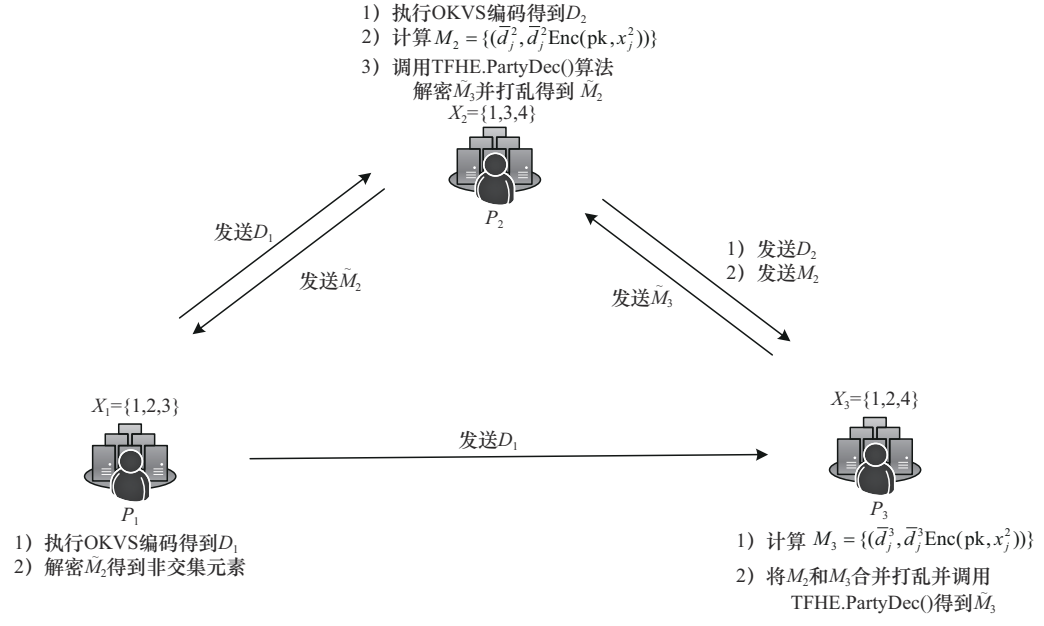

图3　三方 PSU 协议流程

协议1 多方隐私集合并集基础协议

公共参数：N个参与方$P_1,\cdots,P_N$，每个参与方拥有一个集合$X_i=\{x_1^i,\cdots,x_n^i\}$；一个OKVS协议包括(Encode,Decode)算法；一个门限同态加密方案。

协议离线阶段。

对于$i\in[1,N]$，P_i共同选取TFHE的参数setup，然后执行TFHE.Keygen(setup)算法，P_i获得公共公钥pk和私有的部分私钥sk_i。

协议在线阶段。

1) 对于$1\leqslant i<m\leqslant N,j\in[1,n]$，$P_i$调用OKVS的编码Encode将键值对$\{(x_j^i,\mathrm{Enc}(\mathrm{pk},0))\}$编码为数据结构$D_i=\mathrm{Encode}(\{(x_j^i,\mathrm{Enc}(\mathrm{pk},0))\})$，并将数据结构$D_i$发送给$P_m$。

2) 对于$1\leqslant i<m\leqslant N,j\in[1,n]$，$P_m$收到$P_i$发送的$D_i$解码算法得到$d_j^{m,i}=\mathrm{Decode}(\{(D_i,x_j^m)\})$，计算$(\bar{d}_j^m,\bar{d}_j^m\,\mathrm{Enc}(\mathrm{pk},x_j^m))=(\prod_{i=1}^{m-1}d_j^{m,i},x_j^m\prod_{i=1}^{m-1}d_j^{m,i})$。

3) 对于$i\in[2,N-1],j\in[1,n]$，P_i将集合$M_i=\{(\bar{d}_j^i,x_j^i\bar{d}_j^i)\}$发送给$P_N$，其中$P_2$需要做一次密文运算（做一次密文运算不改变真实值）。

4) 对于$i\in[2,N],j\in[1,n]$，P_N将所有$M_i=\{(\bar{d}_j^i,x_j^i\bar{d}_j^i)\}$合并为一个集合$\tilde{M}$，并将数组打乱后，调用TFHE.PartyDec()算法进行部分解密后得到$\tilde{M}_N$并发送给P_{N-1}。

5) 对于$i=N-1,N-2,\cdots,2$，P_i将$\tilde{M}_{i+1}$打乱后，调用TFHE.PartyDec()算法进行部分解密后得到$\tilde{M}_i$，并将$\tilde{M}_i$发送给P_{i-1}。

6) P_1收到$\tilde{M}_2$使用私钥sk_1进行最终解密后，如果是交集元素则得到的键值对(0,0)，否则得到键值对(r,rx_j^i)，其中r是随机数，计算可得到非交集元素x_j^i，其中$i\in[2,N],j\in[1,n]$。

3.2 安全证明

定理1 在不经意键值存储和同态加密的混合模式下，$\prod_{\mathrm{MPSU}}^{P_1,\cdots,P_N}$实现了针对半诚实敌手$\{P_1,\cdots,P_N\}$的任何子集的破坏，都能安全地计算$F_{\mathrm{MPSU}}^{P_1,\cdots,P_N}$功能。

证明 首先证明MPSU协议的正确性，即执行协议能得到正确的结果，然后证明该协议在半诚实环境下的安全性。

正确性。$P_1,\cdots,P_N$执行多方隐私集合并集协议，最终P_1获得键值对集合$\{(\bar{d}_j^i,\bar{d}_j^ix_j^i)\}$并解密计算，如果是交集元素则得到键值对(0,0)，否则得到键值对(r,rx_j^i)，其中r是随机数，计算可得到非交集元素x_j^i。

首先分析当$i\in[1,N-1]$时，P_i执行OKVS编码算法的正确性。P_i共同调用门限同态加密方案算法输入公共参数多项式空间$\mathcal{R}=\dfrac{\mathbb{Z}[X]}{X^n+1}$，以及系数模$q$的多项式空间$\mathcal{R}_q=\dfrac{\mathbb{Z}_q^n[X]}{X^n+1}$，输出公共私钥pk和分享私钥$\mathrm{sk}_i$。其最终的私钥$\mathrm{sk}=s\in\mathbb{Z}_q^n$，公钥$\mathrm{pk}=(a,b)=(a_j,a_js+te_j)$，其中$a\in\mathbb{Z}_q^n$，$t$为明文模数，$e$为一个系数满足高斯分布的多项式。使用pk对明文$m=0$加密$n$次得到$n$组密文$(a_j^i,a_j^is+m+te_j^i)$，然后将$(x_j^i,a_j^i)$和$(x_j^i,a_j^is+m+te_j^i)$作为键值执行OKVS编码算法得到

$$\begin{aligned}D_i&=(D_i',D_i'')=\\&(\mathrm{Encode}(\{(x_j^i,a_j^i)\}),\\&\mathrm{Encode}(\{(x_j^i,a_j^is+m+te_j)\}))\end{aligned}\tag{1}$$

根据文献[13]中给出的编码失败的概率为$\varepsilon=\Pr[\mathrm{Encode}(\{(k_i,v_i)\})=\perp]=2^{-29.355}$，随后$P_i$将数据结构$D_i$发给$P_m$，其中$2\leqslant i<m\leqslant N$。$P_m$收到$P_i$发送的$D_i$后，解码算法得到

$$\begin{aligned}&d_j^{m,i}=(\{d_1^{m,i\prime},\cdots,d_n^{m,i\prime}\},\{d_1^{m,i\prime\prime},\cdots,d_n^{m,i\prime\prime}\})=\\&(\mathrm{Decode}(D_i',d_j^{m,1}),\mathrm{Decode}(D_i'',d_j^{i,1}))=\\&(a_j^{m,i},a_j^{m,i}s+m_j^{m,i}+te_j^{m,i})\end{aligned}\tag{2}$$

并计算

$$(\bar{d}_j^m,\bar{d}_j^m\,\mathrm{Enc}(\mathrm{pk},x_j^m))_{j\in[1,n]}=(\prod_{i=1}^{m-1}d_j^{m,i},x_j^m\prod_{i=1}^{m-1}d_j^{m,i})\tag{3}$$

最后经过$P_N,\cdots,P_2$打乱和部分解密后将结果发给P_1，P_1收到键值对$\{(\bar{d}_j^i,\bar{d}_j^ix_j^i)\},i\in[2,N],j\in[1,n]$用私钥$\mathrm{sk}_1$解密后，如果是交集元素则得到键值对(0,0)，否则得到键值对(r,rx_j^i)，其中r是随机数。其正确性分为交集元素和非交集元素2种情况。

1) 交集元素的情况。如果$\exists x_j^m\in X_i,1\leqslant i<m\leqslant N,j\in[1,n]$，则$\prod_{i=1}^{m-1}\tilde{m}_j^{m,i}=0$。

$$\bar{d}_j^m=(\tilde{a}_j^m,\tilde{a}_j^ms+\prod_{i=1}^{m-1}\tilde{m}_j^{m,i}+t\tilde{e}_j^m)\tag{4}$$

$$\bar{d}_j^mx_j^m=(\tilde{a}_j^m,\tilde{a}_j^ms+x_j^i\prod_{i=1}^{m-1}\tilde{m}_j^{m,i}+t\tilde{e}_j^m)\tag{5}$$

使用 sk 解密 $\bar{d}_j^i$ 和 $\bar{d}_j^i x_j^i$ 的结果为

$$\mathrm{Dec}(\mathrm{sk},\bar{d}_j^i)=[\tilde{a}_j^m s+\prod_{i=1}^{m-1}\tilde{m}_j^{m,i}+t\tilde{e}_j^m-\tilde{a}_j^m s]_t=$$
$$[\prod_{i=1}^{m-1}\tilde{m}_j^{m,i}+t\tilde{e}_j^m]_t=\prod_{i=1}^{m-1}\tilde{m}_j^{m,i}=0 \tag{6}$$

$$\mathrm{Dec}(\mathrm{sk},d_j^i x_j^i)=[\tilde{a}_j^m s+x_j^m\prod_{i=1}^{m-1}\tilde{m}_j^{m,i}+t\tilde{e}_j^m-\tilde{a}_j^m s]_t=$$
$$[x_j^m\prod_{i=1}^{m-1}\tilde{m}_j^{m,i}+t\tilde{e}_j^m]_t=x_j^m\prod_{i=1}^{m-1}\tilde{m}_j^{m,i}=0 \tag{7}$$

其中，$[]_t$ 为取模计算。

2) 非交集元素的情况。如果 $\forall x_j^m\notin X_i,1\leqslant i<m\leqslant N,j\in[1,n]$，则 $\prod_{i=1}^{m-1}\tilde{m}_j^{m,i}=r_j^{m,i}$，用 sk 解密 $\bar{d}_j^i$ 和 $\bar{d}_j^i x_j^i$ 的结果为

$$\mathrm{Dec}(\mathrm{sk},\bar{d}_j^i)=[\tilde{a}_j^i s+\prod_{i=1}^{m-1}\tilde{m}_j^{m,i}+t\tilde{e}_j^m-\tilde{a}_j^m s]_t=$$
$$[\prod_{i=1}^{m-1}\tilde{m}_j^{m,i}+t\tilde{e}_j^m]_t=\prod_{i=1}^{m-1}\tilde{m}_j^{m,i}=r_j^{m,i} \tag{8}$$

$$\mathrm{Dec}(\mathrm{sk},d_j^i x_j^i)=[a_j^m s+x_j^m\prod_{i=1}^{m-1}\tilde{m}_j^{m,i}+te_j^m-a_j^m s]_t=$$
$$[x_j^m\prod_{i=1}^{m-1}\tilde{m}_j^{m,i}+te_j^m]_t=x_j^m r_j^{m,i} \tag{9}$$

其中，$[]_t$ 为取模计算，随后可得到非交集元素为

$$x=\frac{\mathrm{Dec}(\mathrm{sk},\bar{d}_j^m x_j^m)}{\mathrm{Dec}(\mathrm{sk},\bar{d}_j^m)}=\frac{x_j^m r_j^{m,i}}{r_j^{m,i}}=x_j^m \tag{10}$$

安全性。证明多方隐私集合并集协议在不同腐败方下的安全性，首先本文设腐败方集合为集合 C，构建了腐败方集合的模拟器 $\mathrm{Sim}_{\mathrm{MPSU}}^C$，用来模拟腐败方集合 C 的视图。

首先假设 P_N 为诚实方，腐败方集合为 $C\subset\{P_1,\cdots,P_{N-1}\}$。

1) 对于所有的腐败方 $P_i\in C$，模拟器 $\mathrm{Sim}_{\mathrm{MPSU}}^C$ 随机采样数据结构 $D_k,k\in[1,i-1]$，满足如果 P_i 的元素 $x_j^i\in X_k$ 时，$\mathrm{Decode}(D_k,x_j^i)=\mathrm{Enc}(\mathrm{pk},0)$，然后 $\mathrm{Sim}_{\mathrm{MPSU}}^C$ 将所有的 $D_k,k\in[1,i-1]$ 添加到视图中。

2) 对于所有的腐败方 $P_i\in C$，模拟器 $\mathrm{Sim}_{\mathrm{MPSU}}^C$ 随机采样 $\tilde{M}_{i+1}=\{(\bar{d}_j^i,x_j^i\bar{d}_j^i)\}$，满足如果 P_i 的集合元素 $x_j^i\in X_k,k\in[1,i-1]$ 时，$(\bar{d}_j^i,x_j^i\bar{d}_j^i)=(\mathrm{Enc}(\mathrm{pk},0),x_j^i\,\mathrm{Enc}(\mathrm{pk},0))$，然后 $\mathrm{Sim}_{\mathrm{MPSU}}^C$ 将所有的 $D_k,k\in[1,i-1]$ 添加到视图中。

$\mathrm{Sim}_{\mathrm{MPSU}}^C$ 无法像诚实方 P_N 一样计算键值对 $\{(\bar{d}_j^N,\bar{d}_j^N x_j^N)\}$，因为 $\mathrm{Sim}_{\mathrm{MPSU}}^{P_1}$ 没有诚实方 $P_{i,i\in[2,N]}$ 的输入集合 X_N。由于不经意键值存储的不经意性，即用同一个数据结构解码不同的值得到的值是不可区分的，用不同数据结构解码相同的值得到的值是不可区分的。所以 $\mathrm{Sim}_{\mathrm{MPSU}}^C$ 视图中的 $\{(\bar{d}_j^N,\bar{d}_j^N x_j^N)\}$ 与真实协议执行过程产生的视图是计算不可区分的。同时，由于 $\{(\bar{d}_j^N,\bar{d}_j^N x_j^N)\}$ 键值对集合是公钥 pk 加密得到的，即使腐败方 $P_i\in C$ 全部进行合谋，没有密钥 sk_N 也无法完全解密。由于全同态加密方案是 IND-CPA 安全的，从而保证了模拟中的视图在计算上与真实协议中的视图无法区分。因此真实视图与模拟视图是计算不可区分的，即

$$\{\mathrm{Sim}_{\mathrm{MPSU}}^C(\mathrm{pk},X_i)\}\overset{c}{\equiv}\{\mathrm{View}_{\Pi}(\mathrm{pk},X_i)\} \tag{11}$$

其中，View 为真实协议执行中根据输入执行的算法得到的结果。

然后假设 P_N 为腐败方，诚实方 P_h 为 $P_1,\cdots,P_{N-1}$ 的其中一个。

1) 对于所有的腐败方 $P_i\in C$，模拟器 $\mathrm{Sim}_{\mathrm{MPSU}}^C$ 随机采样数据结构 $D_k,k\in[1,i-1]$，满足如果 P_i 的元素 $x_j^i\in X_k$ 时，$\mathrm{Decode}(D_k,x_j^i)=\mathrm{Enc}(\mathrm{pk},0)$，然后 $\mathrm{Sim}_{\mathrm{MPSU}}^C$ 将所有的 $D_k,k\in[1,i-1]$ 添加到视图中。

2) 对于腐败方 P_N，模拟器 $\mathrm{Sim}_{\mathrm{MPSU}}^C$ 随机采样 $M_i=\{(\bar{d}_j^i,x_j^i\bar{d}_j^i)\}$，其中 $D_i,i\in[2,N-1]$ 满足如果 P_i 的集合元素 $x_j^i\in X_k,k\in[1,i-1]$ 时，$(\bar{d}_j^i,x_j^i\bar{d}_j^i)=(\mathrm{Enc}(\mathrm{pk},0),x_j^i\,\mathrm{Enc}(\mathrm{pk},0))$，然后 $\mathrm{Sim}_{\mathrm{MPSU}}^C$ 将所有的 $D_k,k\in[1,i-1]$ 添加到视图中。

$\mathrm{Sim}_{\mathrm{MPSU}}^C$ 无法像诚实方 P_h 一样计算键值对 $\{(\bar{d}_j^h,\bar{d}_j^h x_j^h)\}$，因为 $\mathrm{Sim}_{\mathrm{MPSU}}^{P_1}$ 没有诚实方 P_h 的输入集合 X_h。由于不经意键值存储的不经意性，即用同一个数据结构解码不同的值得到的值是不可区分的，用不同数据结构解码相同的值得到的值是不可区分的。所以 $\mathrm{Sim}_{\mathrm{MPSU}}^C$ 的视图中的 $\{(\bar{d}_j^h,\bar{d}_j^h x_j^h)\}$ 与真实协议执行过程产生的视图是计算不可区分的。同时，由于 $\{(\bar{d}_j^h,\bar{d}_j^h x_j^h)\}$ 键值对集合是公钥 pk 加密得到的，即使腐败方 $P_i\in C$ 全部进行合谋，没有密钥 sk_h 也无法完全解密，由于全同态加密方案是 IND-CPA 安全的，从而保证了模拟中的视图在计算上与真实协议中的视图无法区分。因此，真实视图与模拟视图是计算不可区分的，即

$$\{\mathrm{Sim}^{C}_{\mathrm{MPSU}}(\mathrm{pk},X_i)\} \overset{c}{\equiv} \{\mathrm{View}_{\Pi}(\mathrm{pk},X_i)\} \quad (12)$$

其中，View为真实协议执行中根据输入执行的算法得到的结果。

4 协议性能与比较

本文协议使用C++实现，并在ubuntu 20.04系统上进行实验，其中CPU为E5-2630@2.20GHz，内存为128 GB。在本文协议中，门限同态加密[46-47]使用openFHE库进行实例化，OKVS方案使用文献[14]进行实例化。下面对本文多方隐私集合并集协议通信和计算开销等方面的性能进行评估。本文协议设置计算安全参数$\kappa=128$，统计安全参数$\lambda=40$。

4.1 计算和通信复杂度

本节主要分析协议的计算复杂度。首先，在多方隐私集合并集协议中，离线阶段计算多个0的密文值的计算复杂度为$O((N-1)n)$，在线阶段OKVS编码和解码算法阶段的计算复杂度分别为$O(N(n+\lambda))$和$O(Nn)$，在密文下的同态乘法的计算复杂度为$O\left(\frac{(1+N)Nn}{2}\right)$，打乱解密的计算复杂度为$O(Nn)$。其次，在多方隐私集合并集协议中，发送OKVS编码得到的数据结构的总通信复杂度为$O\left(\frac{N(N-1)n\kappa}{2}\right)$，参与方$P_2,\cdots,P_N$将数据发给$P_N$的通信复杂度为$O((N-1)n\kappa)$，打乱和部分解密的通信复杂度为$O((N^2-2N+1)n\kappa)$。

4.2 实验数据

本文所提多方隐私集合并集协议的运行时间与参与方数量及参与方的集合大小有关，在这里本文对于不同参与方数量$N=3,4,5,7,10$，分别测试了集合大小为$n=2^8,2^{10},2^{12},2^{14},2^{16},2^{18},2^{20}$的通信开销和运行时间，在相同的环境下复现了文献[44]和文献[45]协议并与本文协议进行了对比。从实验数据中能看出，本文协议与文献[44]协议相比，通信开销要优于文献[44]协议，如当三方集合大小为2^{20}时，通信开销降低了65%左右。本文协议与文献[45]相比，在运行时间上要优于文献[45]协议，尤其是在集合较小的场景下，优势更明显。MPSU的通信开销如表2所示，在200 Mbit带宽下MPSU的运行时间如表3所示。

表2 MPSU的通信开销

参与方人数	协议	通信开销/MB						
		$n=2^8$	$n=2^{10}$	$n=2^{12}$	$n=2^{14}$	$n=2^{16}$	$n=2^{18}$	$n=2^{20}$
3	文献[44]	0.83	2.79	10.61	41.79	166.74	665.28	2 655
	文献[45]	0.111	0.426	1.690	6.788	27.87	112.7	455.6
	本文协议	0.089	0.324	1.248	5.655	22.465	89.585	357.837
4	文献[44]	1.68	5.68	21.6	85.06	339.52	1 354.56	5 408
	文献[45]	0.204	0.791	3.145	12.81	51.69	208.8	843.7
	本文协议	0.262	0.967	3.751	17.565	69.937	279.176	1 115.7
5	文献[44]	2.82	9.55	36.37	143.3	571.85	2 281.6	9 106
	文献[45]	0.323	1.257	5.007	20.36	82.11	331.5	1 339
	本文协议	0.448	1.655	6.419	30.110	119.895	478.627	1 912.80
7	文献[44]	5.98	20.28	77.28	304.53	1 215.25	4 848.96	19 353
	文献[45]	0.634	2.489	10.03	40.31	162.5	655.4	—
	本文协议	0.959	3.545	13.757	64.608	257.282	1 027.12	4 104.89
10	文献[44]	12.9	43.83	167.05	658.3	2 627.3	10 483.2	—
	文献[45]	1.288	5.086	20.48	82.39	332.0	—	—
	本文协议	2.075	7.670	29.767	139.88	557.035	2 223.83	8 887.61

表3　在 200 Mbit 带宽下 MPSU 的运行时间

参与方人数	协议	运行时间/s						
		$n=2^8$	$n=2^{10}$	$n=2^{12}$	$n=2^{14}$	$n=2^{16}$	$n=2^{18}$	$n=2^{20}$
3	文献[44]	0.149 1	0.185 3	0.336 8	0.732 7	3.019 9	14.657 1	61.595
	文献[45]	2.166	2.332	3.157	3.734	4.444	9.705	33.10
	本文协议	0.028	0.073	0.101	0.396	1.993	6.442	27.033
4	文献[44]	0.209 4	0.226 7	0.346 4	0.997	6.603 5	29.298 6	119.696
	文献[45]	2.969	3.298	3.976	4.618	6.507	17.10	59.21
	本文协议	0.048	0.124	0.173	0.719	3.637	11.87	55.436
5	文献[44]	0.249 1	0.253 8	0.380 8	2.201 5	11.593 7	48.853 2	197.83
	文献[45]	3.733	4.471	4.800	5.521	8.938	25.95	95.40
	本文协议	0.068	0.175	0.245	1.042	6.081	18.413	53.838
7	文献[44]	0.354 5	0.43	0.938	5.622 3	25.198 3	103.562 9	—
	文献[45]	5.381	6.207	6.644	8.164	17.67	56.894	—
	本文协议	0.108	0.277	0.389	3.689	9.968	37.725	—
10	文献[44]	0.610 2	0.784	2.795 4	13.182 8	55.168 2	—	—
	文献[45]	7.780	8.032	8.635	9.203	14.54	38.31	—
	本文协议	0.189	0.490	1.606	4.659	19.803	—	—

5　结束语

本文针对现有多方隐私集合并集协议交互轮数多以及通信开销过大等问题，设计了一种通信轮数少与通信开销小的多方隐私集合并集协议，解决了多方隐私集合并集协议无法适用于带宽受限的场景下的问题。通过算法分析与实验结果表明，本文协议与现有的协议相比具有更好的性能。

参考文献：

[1] HOGAN K, LUTHER N, SCHEAR N, et al. Secure multiparty computation for cooperative cyber risk assessment[C]//Proceedings of the 2016 IEEE Cybersecurity Development (SecDev). Piscataway: IEEE Press, 2016: 75-76.

[2] RAMANATHAN S, MIRKOVIC J, YU M L. BLAG: improving the accuracy of blacklists[C]//Proceedings of 2020 Network and Distributed System Security Symposium. Piscataway: IEEE Press, 2020: 1-6.

[3] ZHANG C, CHEN Y, LIU W R, et al. Linear private set union from multi-query reverse private membership test[C]//Proceedings of the 32nd USENIX Conference on Security Symposium. Berkeley: USENIX Association, 2023: 337-354.

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

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

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

[7] RINDAL P, ROSULEK M. Malicious-secure private set intersection via dual execution[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2017: 1229-1242.

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

[9] 张恩, 耿魁, 金伟, 等. 抗隐蔽敌手的云外包秘密共享方案[J]. 通信学报, 2017, 38(5): 57-65.
ZHANG E, GENG K, JIN W, et al. Cloud outsourcing secret sharing scheme against covert adversaries[J]. Journal on Communications, 2017, 38(5): 57-65.

[10] PINKAS B, ROSULEK M, TRIEU N, et al. PSI from PaXoS: fast, malicious private set intersection[C]//Advances in Cryptology-EUROCRYPT 2020. Berlin: Springer, 2020: 739-767.

[11] CHASE M, MIAO P H. Private set intersection in the Internet setting from lightweight oblivious PRF[C]//Advances in Cryptology-CRYPTO 2020. Berlin: Springer, 2020: 34-63.

[12] RINDAL P, SCHOPPMANN P. VOLE-PSI: fast OPRF and circuit-PSI from vector-OLE[C]//Advances in Cryptology-EUROCRYPT 2021. Berlin: Springer, 2021: 901-930.

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

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

[15] 魏立斐, 王勤, 张蕾, 等. 半可信云服务器辅助的高效隐私交集计算协议[J]. 软件学报, 2023, 34(2): 932-944.
WEI L F, WANG Q, ZHANG L, et al. Efficient private set intersection protocols with semi-trusted cloud server aided[J]. Journal of Software, 2023, 34(2): 932-944.

[16] ISHAI Y, KILIAN J, NISSIM K, et al. Extending oblivious transfers efficiently[C]//Advances in Cryptology-CRYPTO 2003. Berlin: Springer, 2003: 145-161.

[17] COUTEAU G, RINDAL P, RAGHURAMAN S. Silver: silent VOLE and oblivious transfer from hardness of decoding structured LDPC codes[C]//Advances in Cryptology-CRYPTO 2021. Berlin: Springer, 2021: 502-534.

[18] 张蕾, 贺崇德, 魏立斐. 高效且恶意安全的三方小集合隐私交集计算协议[J]. 计算机研究与发展, 2022, 59(10): 2286-2298.
ZHANG L, HE C D, WEI L F. Efficient and malicious secure three-party private set intersection computation protocols for small sets[J]. Journal of Computer Research and Development, 2022, 59(10): 2286-2298.

[19] 张恩, 裴瑶瑶, 杜蛟. 基于RLWE的密文策略属性代理重加密[J]. 通信学报, 2018, 39(11): 129-137.
ZHANG E, PEI Y Y, DU J. RLWE-based ciphertext-policy attribute proxy re-encryption[J]. Journal on Communications, 2018, 39(11): 129-137.

[20] 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.
SONG X F, GAI M, ZHAO S N, et al. Privacy-preserving statistics protocol for set-based computation[J]. Journal of Computer Research and Development, 2020, 57(10): 2221-2231.

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

[22] ZHANG E, LIU F H, LAI Q 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: ACM Press, 2019: 93-104.

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

[24] 张恩, 秦磊勇, 杨刃林, 等. 基于弹性秘密共享的多方门限隐私集合交集协议[J]. 软件学报, 2023, 34(11): 5424-5441.
ZHANG E, QIN L Y, YANG R L, et al. Multi-party threshold private set intersection protocol based on robust secret sharing[J]. Journal of Software, 2023, 34(11): 5424-5441.

[25] 魏立斐, 刘纪海, 张蕾, 等. 双云辅助的超阈值多方隐私集合交集计算协议[J]. 软件学报, 2023, 34(11): 5442-5456.
WEI L F, LIU J H, ZHANG L, et al. Two cloud-assisted over-threshold multi-party private set intersection calculation protocol[J]. Journal of Software, 2023, 34(11): 5442-5456.

[26] LIU F H, ZHANG E, QIN L Y. Efficient multiparty probabilistic threshold private set intersection[C]//Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2023: 2188-2201.

[27] FENSKE E, MANI A, JOHNSON A, et al. Distributed measurement with private set-union cardinality[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2017: 2295-2312.

[28] DONG C Y, LOUKIDES G. Approximating private set union/intersection cardinality with logarithmic complexity[J]. IEEE Transactions on Information Forensics and Security, 2017, 12(11): 2792-2806.

[29] BLANTON M, AGUIAR E. Private and oblivious set and multiset operations[J]. International Journal of Information Security, 2016, 15(5): 493-518.

[30] KOLESNIKOV V, ROSULEK M, TRIEU N, et al. Scalable private set union from symmetric-key techniques[C]//Advances in Cryptology-ASIACRYPT 2019. Berlin: Springer, 2019: 636-666.

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

[32] JIA Y X, SUN S F, ZHOU H S, et al. Shuffle-based private set union: faster and more secure[C]//31st USENIX Security Symposium. Berkeley: USENIX Association, 2022: 2947-2964.

[33] TU B B, CHEN Y, LIU Q, et al. Fast unbalanced private set union from fully homomorphic encryption[C]//Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2023: 2959-2973.

[34] HAZAY C, NISSIM K. Efficient set operations in the presence of malicious adversaries[J]. Journal of Cryptology, 2012, 25(3): 383-433.

[35] MANY D, BURKHART M, DIMITROPOULOS X. Fast private set operations with SEPIA[R]. 2012.

[36] DAVIDSON A, CID C. An efficient toolkit for computing private set operations[C]//Information Security and Privacy. Berlin: Springer, 2017: 261-278.

[37] CANETTI R, PANETH O, PAPADOPOULOS D, et al. Verifiable set operations over outsourced databases[C]//Public-Key Cryptography-PKC 2014. Berlin: Springer, 2014: 113-130.

[38] SHISHIDO K, MIYAJI A. Efficient and quasi-accurate multiparty private set union[C]//Proceedings of the 2018 IEEE International Conference on Smart Computing (SMARTCOMP). Piscataway: IEEE Press, 2018: 309-314.

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

[40] FRIKKEN K. Privacy-preserving set union[C]//Applied Cryptography and Network Security. Berlin: Springer, 2007: 237-252.

[41] HONG J, KIM J W, KIM J, et al. Constant-round privacy preserving

multiset union[J]. Bulletin of the Korean Mathematical Society, 2013, 50(6): 1799-1816.

[42] SEO J H, CHEON J H, KATZ J. Constant-round multi-party private set union using reversed Laurent series[C]//Public Key Cryptography-PKC 2012. Berlin: Springer, 2012: 398-412.

[43] GONG X H, HUA Q S, JIN H. Nearly optimal protocols for computing multi-party private set union[C]//Proceedings of the 2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS). Piscataway: IEEE Press, 2022: 1-10.

[44] LIU X, GAO Y. Scalable multi-party private set union from multi-query secret-shared private membership test[C]//Advances in Cryptology-ASIACRYPT 2023. Singapore: Springer, 2023: 237-271.

[45] DONG M L, CHEN Y, ZHANG C, et al. Breaking free: efficient multi-party private set union without non-collusion assumptions[J]. arXiv Preprint, arXiv: 2406.07011, 2024.

[46] BADAWI A A, BATES J, BERGAMASCHI F, et al. OpenFHE: open-source fully homomorphic encryption library[C]//Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography. New York: ACM Press, 2022: 53-63.

[47] ASHAROV G, JAIN A, LÓPEZ-ALT A, et al. Multiparty computation with low communication, computation and interaction via threshold FHE[C]//Advances in Cryptology-EUROCRYPT 2012. Berlin: Springer, 2012: 483-501.

[作者简介]

张恩（1974– ），男，河南新乡人，博士，河南师范大学教授、硕士生导师，主要研究方向为网络安全、密码协议设计、隐私保护。

王梦涛（1999– ），男，河南周口人，河南师范大学硕士生，主要研究方向为密码协议。

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

禹勇（1980– ），男，山东泰安人，博士，陕西师范大学教授、博士生导师，主要研究方向为公钥密码理论及应用、区块链安全、数据安全与隐私保护。

黄昱晨（1999– ），男，河南商丘人，河南师范大学硕士生，主要研究方向为密码协议。