布尔电路上保护隐私集合并集运算的研究与实现

孙茂华*①　胡　磊①　朱洪亮②　李　祺②
①(首都经济贸易大学信息学院　北京　100070)
②(北京邮电大学计算机学院　北京　100876)

摘　要：隐私保护技术是当前信息安全领域的研究热点。然而，现阶段集合并集运算中的隐私保护技术侧重理论研究，在实验模型的开发上较为欠缺。针对该问题，该文首先设计了保护隐私的集合合并运算电路、去重电路和混淆电路，并应用 YAO 氏通用混淆电路估值技术提出了一种布尔电路上保护隐私的集合并集协议。然后，该文使用模拟器视图仿真法证明了协议的安全性。最后，基于 MightBeEvil 中的 YAO 氏混淆电路估值框架，开发了该文理论方案对应的实验模型。实验结果表明，在安全计算稀疏集合的并集时，所提算法效率优于当前布尔电路上的其他算法。

关键词：安全多方计算；YAO 氏混淆电路技术；保护隐私的集合并集运算
中图分类号：TP309　　**文献标识码**：A　　**文章编号**：1009-5896(2016)06-1412-07
DOI: 10.11999/JEIT150911

Research and Implementation of Privacy Preserving Set Union in Boolean Circuits

SUN Maohua①　HU Lei①　ZHU Hongliang②　LI Qi②
①(*Information School*, *Capital University of Economics and Business*, *Beijin*g 100070, *China*)
②(*School of Computer*, *Beijing University of Posts and Telecommunications*, *Beijing* 100876, *China*)

Abstract: Privacy-preserving technology is the focus of information security area. Unfortunately, rare implementation of private set union protocol is developed. To solve the issue above, a novel private set union protocol based on the YAO's garbled circuit technology is presented. The specially designed circuits include the private set merge circuit, the private set filter circuit and the private set confusion circuit. Then, the security of the novel protocol is proven in semi-honest model. Finally, a prototype of the protocol is built based on the MightBeEvil framework. The simulation results show that this protocol is more efficient than the existing one when evaluating the union of sparse sets in a privacy-preserving manner.

Key words: Secure multi-party computation; YAO's garbled circuit technology; Private set operation

1　引言

保护隐私的集合运算是安全多方计算的一个重要研究分支，是近几年来国内外的研究热点。保护隐私集合运算的函数表现形式有算术电路和布尔电路两种。根据数据结构的不同，算术电路上保护隐私的集合运算分为基于茫然多项式估值的方案、基于茫然伪随机函数评估的方案和基于布隆过滤器的方案。例如，2005 年 CRYPTO 大会上，文献[1]使用茫然多项式估值实现了半诚实模型下保护隐私的集合并集运算协议(Private Set Union, PSU)，但是该协议会泄漏交集信息；2007 年 ACNS 大会上，文献[2]提出了恶意模型下保护隐私的并集运算协议，该协议的计算复杂度为 nk^2 模乘操作，通信复杂度为 $O(kn^2+k^2n^2)$；2012 年 PKC 大会上，文献[3]使用茫然有理函数表示集合并借助逆向罗伦级数实现了保护隐私的并集运算，其恶意模型下协议的计算复杂度为 k^2n^4 次模乘操作，通信复杂度为 $O(k^2n^3)$。文献[4]将布隆过滤器引入到保护隐私的交集运算(Private Set Intersection, PSI)中，使用安全多方乘法协议得到参与者布隆过滤器向量的交集，进而得到输入集合的交集；但是该算法是不安全的，因为交集布隆过滤器向量泄漏了输入信息。此后，文献[5]使用布隆过滤器和 GM 同态加密方案设计了保护隐私的交集运算协议，其半诚实模型下的协议

收稿日期：2015-08-05；改回日期：2016-02-29；网络出版：2016-05-05
*通信作者：孙茂华　starjingxiang@sina.com
基金项目：首都经济贸易大学 2014 年青年科研启动基金，首都经济贸易大学青年科学基金(2014XJQ016)，国家自然科学基金(61302087)，2016 年北京市教委科研水平提高基金
Foundation Items: Young Scientific Research Starting Foundation of CUEB 2014, Young Scientists Program of CUEB (2014XJQ016), The National Natural Science Foundation of China (61302087), Improve Scientific Rescarch Foundation of Beijing Education 2016

需要 kn 次 Hash 操作和 $(k\log_2 e+kl+k+2l)n$ 次模乘操作。文献[6]使用 XOR-秘密共享和茫然传输设计了基于布隆过滤器的保护隐私交集运算协议，该协议需要 $2(k+k\log_2 e)n$ 次 Hash 操作和数百次公钥操作。文献[7]研究了 Server-aided 模式下数亿级集合上的保护隐私集合并集运算。文献[8]基于 El-Gamal 同态加密方案和茫然多项式估值技术实现了保护隐私的集合并集运算。文献[9]使用茫然传输扩展协议设计了随机混淆布隆过滤器，进而优化了文献[6]协议的效率。文献[10]基于代数伪随机函数的陷门高效性设计了恶意模型下保护隐私集合交集协议。上述保护隐私的集合运算协议假设输入集合的大小信息是公开的，文献[11]研究了如何同时保护输入集合的内容和大小信息。此外，文献[12–14]探讨了算术电路上保护隐私交集外包运算，将协议的计算模型由传统模型转换到外包模型。文献[15]基于 LWE 困难性假设提出了格上的保护隐私集合交集运算协议。布尔电路上保护隐私的集合运算协议主要借助通用混淆电路估值技术实现隐私保护。例如，2005 年 ASIACRYPT 大会上，文献[16]使用比特向量表示参与者的秘密集合，利用 YAO 氏通用混淆电路估值协议在 OR 门组成的电路上实现了保护隐私集合并集运算协议。2012 年 NDSS 大会上，文献[17]在布尔电路上设计并实现了保护隐私的集合交集运算协议；其研究成果表明，通过合理地设计专用电路，布尔电路上安全多方计算协议的效率在某些情况下高于基于算术电路的协议效率。文献[9]基于 GMW 电路估值协议提出了更加高效的保护隐私的集合交集协议。文献[18]提出了基于置换 Hash 的保护隐私集合交集协议，该协议比文献[17]的方案提速约 5 倍。

虽然保护隐私并集运算在近几年取得了一定的研究成果，但是现有研究仍然存在如下问题：(1)布尔电路上保护隐私的并集协议[16]在处理稀疏集合时效率较低；(2)目前的多数文献主要集中在理论研究，即通过对渐进计算复杂度表达式和渐进通信复杂度表达式的对比得到效率分析结果，并没有实现实验模型的开发。鉴于实验模型设计和开发是近几年来安全多方计算研究的一种重要方法，设计并开发保护隐私并集运算的实验模型显得十分必要。由此，本文主要致力于提高布尔电路上针对稀疏集合并集运算协议的效率和开发对应的实验模型。具体地，本文基于 YAO 氏混淆电路估值技术，通过设计并集运算的专用电路来实现布尔电路上保护隐私的并集运算协议。输入信息的数据结构采用双调排序序列。专用电路包括预处理、合并电路、去重电路、混淆电路 4 个部分。为了将理论研究成果转换为实际应用，本文借助于 MightBeEvil 应用程序框架实现了应用模型的开发。实验结果表明，当计算稀疏集合的并集时，本文提出方案的效率较高。

2　保护隐私集合并集运算协议

本节按照预处理-集合合并-去重-混淆 4 个阶段设计保护隐私的集合并集运算电路。

2.1 预处理

预处理阶段，预处理协议如表 1 所示。参与者首先通过协商得到集合大小 N，然后，参与者在本地对各自的秘密集合进行排序。接下来，参与者在自己的排序集合中插入若干个扰乱因子，从而形成大小为 N 的集合。通过加入扰乱因子，参与者将无法通过攻击获得双方集合交集的大小。

表 1 预处理协议

参与者：	P_1 和 P_2
输入数据：	P_1 输入集合 S_1，P_2 输入集合 S_2
步骤 1	P_i 随机产生数据 N_i，要求 $n_i>\lvert S_i\rvert$。P_i 将 n_i 发送给 P_{1-i}，其中 $i=1,2$。
步骤 2	P_1 和 P_2 在本地计算 $N=\max(n_1,n_2)$。
步骤 3	P_1 在本地对 S_1 按照单调递增的顺序排序，得到集合 $S_1^{'}$。P_2 在本地对 S_2 按照单调递减的顺序排序，得到集合 $S_2^{'}$。
步骤 4	P_i 随机挑选 $S_i^{'}$ 中的 $N-\lvert S_i\rvert$ 个数据，并在集合 $S_i^{'}$ 相应数据的后面插入一个备份，得到集合 G_i。

2.2 保护隐私的集合合并协议

本文使用双调排序网络实现集合 G_1 和 G_2 的合并。由于 G_1 中的元素按照单调递增的顺序排列，G_2 中的元素按照单调递减的顺序排列，因此 G_1 和 G_2 构成双调序列，可以使用双调排序完成集合的合并。双调排序网络算法控制数据比较的顺序，而基本的运算单元为数据比较器 Sorter。Sorter 输入为数据 x,y，输出为 $\min(x,y),\max(x,y)$。当 Sorter 的一个输出数据为 x 时，另一个数据必为 y。因此，只需要调用一次 Kolesnikov 数据比较器[19]，然后通过异或门组成的电路可以得到另一个输出数据，如图 1 所示。通过保护隐私的集合合并协议，两个参与者的秘密集合被合并为一个多重集合 S。

性能比较：使用双调排序网络对参与者的私有集合进行合并时，共需要调用 Sorter 电路 $N\log_2(2N)$ 次。表 2 对保护隐私的集合合并协议所需的门数进行了对比。可以看出，本文方案在门电路消耗上优于文献[17]方案。

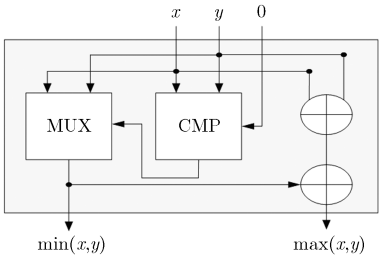


图 1 数据比较器方案

表 2 保护隐私的集合合并协议中门数量对比

方案	总门数	异或门数	其他门数
文献[17]	$11\sigma N\log_2(2N)$	$8\sigma N\log_2(2N)$	$3\sigma N\log_2(2N)$
本文方案	$9\sigma N\log_2(2N)$	$7\sigma N\log_2(2N)$	$2\sigma N\log_2(2N)$

2.3 去重

由于多重集合 S 中重复元素都是相邻的，因此本文对比 S 中两两相邻的元素实现重复元素的过滤。当两个元素相等，输出无效元素(暂且认为 0 元素为无效元素)，否则输出第 1 个元素。去重电路如图 2 所示。去重电路中包含两类过滤器 F_1 和 F_2，如图 3 所示。对于过滤器 F_1，当输入元素 $m_i=m_{i+1}$ 时，$z=m_i\oplus m_{i+1}=\{0\}^n$，即 m_i 和 m_{i+1} 的异或值为 0；当输入元素 $m_i\neq m_{i+1}$ 时，$z=m_i\oplus m_{i+1}\neq\{0\}^n$，即 m_i 和 m_{i+1} 的异或值中至少有一位不为 0。通过对 z 中各比特执行或操作，可以判断 z 的各比特是否全为 0，进而判断出 m_i 和 m_{i+1} 是否相等。如果 $m_i=m_{i+1}$ 则输出 0，否则输出 m_i。接下来，再对 m_{i+1} 和 m_{i+2} 使用过滤器 F_1 进行过滤。对于多重集合 S 中最后的两个元素 m_{2n-1} 和 m_{2n}，如果 $m_{2n-1}=m_{2n}$，则输出 m_{2n} 和 0；否则需要将 m_{2n-1} 和 m_{2n} 都输出，因此对于 S 中最后的两个元素 m_{2n-1} 和 m_{2n} 使用过滤器 F_2 进行过滤。过滤器 F_1 和 F_2 中的多路复合选择器 MUX 仍然选用文献[20]的设计方案。

2.4 混淆

合并后的集合 S 经过去重电路后得到集合 $F=\{f_1,f_2,\cdots,f_{2N}\}$。事实上，$F$ 中的元素是由并集元素和若干个零元素组成。零元素是由去重电路引入的，当某个位置 $f_i=0$ 且 $f_{i+1}\neq 0$，则说明集合 S 中 $m_i=m_{i+1}$。导致 $m_i=m_{i+1}$ 的原因可能是由于参与者在预处理阶段插入的干扰因子，也可能是由于 m_i 是交集元素。如果直接将 F 作为协议的最终输出公布，在极端情况下参与者可能会得知集合交集中某些元素的信息。考虑如下情况：假设参与者 P_i 通过某些社会学方法提前得知了对方集合的大小 $|S_{1-i}|$。对于 P_i 私有集合中的元素 m_i，P_i 在预处理阶段插入了 $t-1$ 个 m_i 的副本。如果集合 F 中元素 $f_x=m_i$，且集合 F 中 f_x 前面的 $N-|S_{1-i}|+t$ 个元素均为 0，则 P_i 可以判断出 m_i 必为集合交集中的元素。

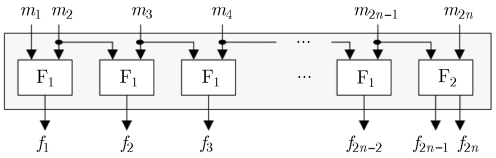


图 2 去重电路顶图

为了抵抗上述攻击方法，本文将对去重电路的输出集合进行混淆，打乱集合中元素的位置，从而保护参与者的私有信息。使用通用电路估值方案对上述预处理、合并和去重电路进行安全计算后，参与者 P_i 将得到集合 F 的一个秘密份额 $F_i=\{f_1^i,f_2^i,\cdots,f_{2N}^i\}$，且满足 $f_x^i\oplus f_x^{1-i}=f_x$。表 3 是本文设计的混淆协议。

表 3 保护隐私的集合混淆协议

参与者：P_1 和 P_2

输入数据：P_1 输入秘密份额 F_1，P_2 输入秘密份额 F_2

输出数据：P_1 得到集合并集，P_2 无输出数据

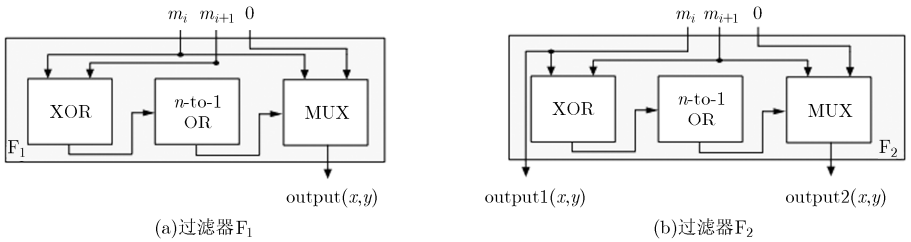
步骤 1　P_1 产生长度是 n bit 的随机数 $r_1=\{0,1\}^n$，并在本地计算 $A=\{f_1^1\oplus r_1,f_2^1\oplus r_1,\cdots,f_{2N}^1\oplus r_1\}$。$P_1$ 将集合 A 发送给 P_2。

步骤 2　P_2 在本地计算

$$B=\{b_1,b_2,\cdots,b_{2N}\}=A\oplus F_2$$
$$=\{f_1^1\oplus r_1\oplus f_x^2,f_2^1\oplus r_1\oplus f_x^2,\cdots,f_{2N}^1\oplus r_1\oplus f_x^2\}$$

步骤 3　P_2 产生 N 个随机对 $K=\{(k_1,k_2),(k_3,k_4),\cdots,(k_{2N-1},k_{2N})\}$，其中 $1\le k_i\le 2N$。P_2 根据 K 中的随机对依次对换集合 B 中的元素 b_{k_i} 和 $b_{k_{i+1}}$。经过 N 轮对换，P_2 得到集合 B' 并发送给 P_1。

步骤 4　P_1 在本地计算 $U'=B'\oplus r_1$，并计算集合的并集 $U=U'-\{0\}$。



(a)过滤器 F_1　　(b)过滤器 F_2

图 3 去重电路详细电路图

在混淆阶段，参与者主要是在本地使用异或门进行计算。参与者 P_1 和 P_2 分别调用异或门 $4nN$ 次。该阶段不需要参与者交互完成门电路的计算。两个参与者共传输 $4nN$ bit 数据。

3　理论分析

3.1 安全性分析

定理 1　假设底层 YAO 氏混淆电路估值方案在半诚实模型下是安全的，本文提出的保护隐私的集合并集协议在半诚实模型下安全地计算了参与者私有集合的交集。

证明　本文提出的保护隐私的集合并集协议是非对称的，也就是说只有参与者获知结果，因此

$$f(x,y)\overset{\text{def}}{=}(f_c(x,y),\Lambda) \tag{1}$$

式中，Λ 代表空字符串。π 代表本文提出的保护隐私的集合并集协议。

服务器视图：下面分析服务器被攻击的情况。在协议 π 执行过程中，服务器的视图为

$$\text{view}_S^{\pi}(S_S,S_C)=\{S_S,\Lambda,I_1,I_2,I_3,K_1,K_2,K_3\} \tag{2}$$

其中，K_1 代表预处理阶段服务器端接收到的信息，可知 $K_1=N_C$。K_2 代表集合合并和去重阶段服务器端接收到的信息，该阶段服务器执行 YAO 氏混淆电路估值方案中的电路构造者角色。K_3 代表混淆阶段服务器端接收到的信息，可知 $K_3=A$。I_1,I_2,I_3 分别代表预处理阶段、集合合并和去重阶段、混淆阶段服务器产生的中间数据。

下面创建模拟器 Sim_S，模拟器接收服务器的私有输入数据 S_S 和服务器输出 U，并模拟服务器在协议执行过程中的视图。首先，模拟器模拟预处理阶段服务器的行为。在预处理阶段，服务器产生的中间数据 $I_1=\{N_S,S_S',G_S\}$。在 $[0,|\,G_S\,|]$ 的范围内，Sim_S 通过多次掷币得到随机数 N_C'，使得 $N_C'=N_C$。然后，模拟器模拟集合合并和去重阶段服务器的行为。单独观察集合合并和去重阶段时，服务器的输入数据为上一阶段的输出 G_S，Sim_S 可以从 I_1 中获取 G_S；服务器的输出数据为下一阶段的输入数据 F_2，Sim_S 可以从 I_2 中获取 F_2。Sim_S 根据 G_S，F_2 以及该阶段服务器独立产生的中间数据 I_2 模拟得到集合合并阶段和去重阶段在电路估值时服务器接收到的信息 K_2'，由于假设底层 YAO 氏混淆电路估值方案在半诚实模型下是安全的，因此 $K_2'\overset{\text{c}}{\equiv}K_2$。最后，$\text{Sim}_S$ 根据服务器在混淆阶段的输入 F_2，输出 Λ，和服务器生成的中间数据 $I_3=\{B,K,B'\}$ 模拟服务器的行为。Sim_S 计算 $A'=B\oplus F_2$。由服务器在协议运行中的行为可知，$A'=A$。协议模拟完成后，Sim_S 输出模拟视图 $\text{view}_{\text{Sim}_S}^{\pi}(S_S,S_C)=\{S_S,\Lambda,I_1,I_2,I_3,N_C',K_2',A'\}$。

由上述模拟行为可知：

$$\text{view}_{\text{Sim}_S}^{\pi}(S_S,S_C)\overset{c}{\equiv}\text{view}_S^{\pi}(S_S,S_C) \tag{3}$$

同理，可以为客户端创建模拟器 Sim_C，且该模拟器的模拟视图和客户端的视图具有如下关系：

$$\text{view}_{\text{Sim}_C}^{\pi}(S_S,S_C)\overset{c}{\equiv}\text{view}_C^{\pi}(S_S,S_C) \tag{4}$$

综上所述，半诚实模型下本文提出的保护隐私的集合并集协议是安全的，定理得证。

3.2 性能分析

本文设计的保护隐私集合并集运算协议所需要的门电路数量如表 4 所示。需要指出的是，在混淆阶段参与者 P_1 和 P_2 分别在本地调用异或门 $4\sigma N$ 次，表 4 中给出的数据是需要参与者交互计算的门电路数量，其中混淆阶段参与者需要交互计算的门电路数量为零。当使用 GMW 评估方案时，对异或门的安全估值不消耗密码学操作，因此本文方案中所需耗费型门数量为 $2\sigma N\log_2(2N)+(2\sigma-1)(2N-1)$。当使用 YAO 氏混淆电路估值方案时，本文方案所需门的数量为 $9\sigma N\log_2(2N)+(5\sigma-1)(2N-1)$。

表 4 本文方案所需门数量

性能分析	合并阶段	去重阶段	混淆阶段
总门数	$9\sigma N\log_2(2N)$	$(5\sigma-1)(2N-1)$	0
异或门数	$7\sigma N\log_2(2N)$	$3\sigma(2N-1)$	0
其他门数	$2\sigma N\log_2(2N)$	$(2\sigma-1)(2N-1)$	0

表 5 对近几年的 PSI 协议和 PSU 协议进行了理论对比。通过对表 5 的分析，我们得出如下结论：

(1)布尔电路上的 PSU 协议的对比：　文献[16]的电路类型只有 OR 门，该方案所需消耗型门数量为 $n^2 2^{\sigma}$。当参数 N 和 σ 满足 $N\prec\sqrt{(2^{\sigma}-1)/\sigma}$ 时，本文协议所消耗的门数量更少，其中 $\prec$ 表示约小于。当使用 YAO 氏电路估值方案，评估不同类型的门电路所消耗的通信复杂度和计算复杂度相同，即此时通信复杂度和计算复杂度仅取决于当前电路中的门数量。因此，针对于稀疏集合，本文方案的通信复杂度和计算复杂度都优于文献[16]方案。

(2)不同类型电路上的 PSU 协议的对比：　布尔电路上的 PSU 协议的复杂度和集合中元素的位数、安全参数、集合大小有关；而表 5 中算术电路上的 PSU 协议的复杂度和集合中元素的位数无关，和安全参数以及集合大小有关。对于给定的系统，集合中元素的位数是确定的，此时本文方案的通信复杂度低于文献[3]方案，高于文献[1]方案和文献[2]

表 5 协议性能的理论对比

协议类型	使用方法	电路类型	通信复杂度	计算复杂度
PSU	文献[1]	算术电路	$O(4kn)$	$O(8n^2asym)$
	文献[2]	算术电路	$O(4kn)$	$O(2n^2asym)$
	文献[3]	算术电路	$O(8kn^2)$	$O(20n^2asym)$
	文献[16]	布尔电路	$O(2^\sigma kn^2)$	$O(n^2 2^\sigma asym)$
	本文方法	布尔电路	$O(6.75\sigma kN\log_2(2N))$	$O(9\sigma N\log_2(2N)sym)$
PSI	文献[21]	算术电路	$O(2kn)$	$O(2n\ asym)$
	文献[6]	算术电路	$O(2.88k^2n)$	$O(4.32kn\ sym)$
	文献[9]	算术电路	$O(0.5\sigma kn)$	$O(0.75\sigma n\ sym)$
	文献[17]	布尔电路	$O(9\sigma kn\log_2 n)$	$O(12\sigma n\log_2 n\ sym)$

注：σ 代表集合中元素的二进制位数，k 代表系统安全参数，n 代表参与者集合的最大长度，N 代表一个大于 n 的随机数。

方案；本文方案的计算复杂度低于文献[1]，文献[2]和文献[3]方案。

(3)本文方案和 PSI 协议的对比： 相比于当前先进的 PSI 协议如文献[6,9,17,21]，本文 PSU 协议的复杂度更高。这是因为 PSU 协议和 PSI 协议虽然都是保护隐私的集合运算协议，但是由于所要实现的运算类型不同，且并集协议中不仅要保护参与者的输入信息还要保护参与者的集合交集信息，使得多数 PSU 协议的计算复杂度和通信复杂度高于 PSI 协议。

4 实验和分析

4.1 实验模型的设计

本文基于MightBeEvil中的YAO氏混淆电路估值框架(下文中简称为 MightBeEvil 框架)设计了实验模型。在 MightBeEvil 框架中，应用问题使用电路库中的混合电路或简单门电路完成布尔电路的搭建，然后使用 YAO 氏混淆电路估值协议完成安全计算；该框架使用 JCE(Java Cryptography Extension)中的密码学算法构建 YAO 氏混淆电路估值协议。因此，可以将实验模型的设计工作转换为布尔电路的搭建和 YAO 氏混淆电路估值协议的实例化。下面介绍本文实验模型在使用 JCE 设计 YAO 氏混淆电路估值协议时所选用的底层协议及相关参数配置。YAO 氏混淆电路估值协议使用 Hash 算法 SHA-1 产生混淆真值表，混淆电路中信号密钥长度 $\omega=80$。OT 协议采用经典 Naor-Pinkas 的方案[22]，该方案中的公钥操作基于 Z_p^* 上的 q 阶子群实现，其中 $|q|=128,|p|=1024$。OT 扩展协议的统计安全参数 $k=80$。可以看出，上述参数符合 NIST 准则中的 ultra-short 安全级别。应用模型包含两个可执行程序，服务器端程序和客户端程序。服务器端程序和客户端程序之间通过 TCP 协议、基于管线模型完成通信。

4.2 实验分析

本文所有实验均在局域网内的两台 PC 上模拟执行。其中，一台 PC 模拟服务器，使用 Thinkpad X230i, CPU 为 2.5 GHz Intel Core i3-3120M，内存为 3.6 GB，操作系统为 Ubuntu。另一台 PC 模拟客户端，使用 Thinkpad R400, CPU 为 2.1 GHz Intel Core 2 Duo，内存为 3 GB，操作系统为 Ubuntu。服务器和客户端之间通过 Wifi 连接。

4.2.1 本文实验模型性能分析 本文实验假设 $N=n$，客户端输入集合和服务器输入集合的交集大小为 $n/2$。本文分别验证了元素位数 $\sigma=16$ 和 $\sigma=32$ 并且客户端和服务器输入集合从 2^7 增加至 2^{13} 时协议运行所需要的时间和占用的带宽，如图 4 和图 5 所示。可以看出，协议的运行时间、占用带宽和元素位数 σ 成正比，和输入集合大小 n 满足 $O(cn\log_2(2n))$ 的关系，其中 c 表示常数。实验结果和理论分析结果保持一致。

4.2.2 实验模型性能对比 由于文献[16]方案只给出了理论方案，本文基于 MightBeEvil 中的 YAO 氏混淆电路估值框架实现了该方案应用模型的开发。在运行实验模型的过程中，当集合元素位数 $\sigma=32$ 时，文献[16]方案中服务器和客户端程序都需要开辟 2^{32} 的 Bit-Set 来表示各自的秘密集合，可惜的是很多客户端 PC 的内存无法满足该需求，本文实验环境也遭遇了同样的问题。当 $\sigma=16,n=13$ 时，文献[17]的应用模型由于存在错误导致无法产生正确结果。

表 6 总结了不同集合大小、不同元素位数下各应用模型的实验数据。由于 $\sigma=32$ 时文献[16]的实验模型无法正常运行，因此我们仅在图 6 和图 7 中对 $\sigma=16$ 时各协议实验模型的运行时间和占用带宽进行了对比。从图 6 和图 7 可以得出如下结论：

(1)针对于布尔电路上的 PSU 协议，当参与者集合元素数量较少时，本文方案在运行时间和占用带宽两个参数上都优于文献[16]方案；随着集合元素数量的增加，文献[16]方案的运行时间和占用的带宽优于本文提出的方案。但是，随着集合中元素位数的增加，文献[16]方案存在应用模型内存泄漏的风

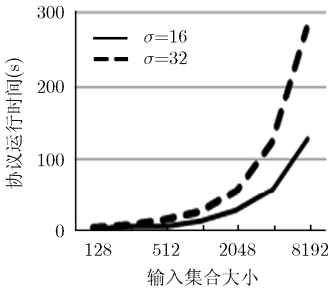


图 4 本文协议运行时间

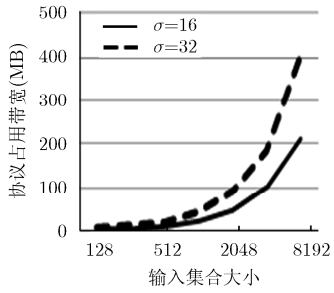


图 5 本文协议占用带宽

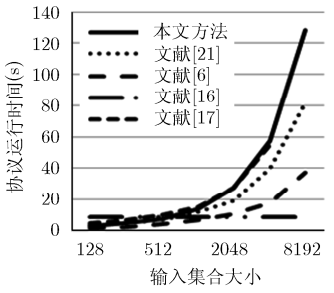


图 6 $\sigma = 16$ 时几种方法协议运行时间对比

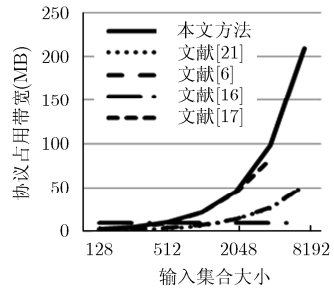


图 7 $\sigma = 16$ 时几种方法协议占用带宽对比

表 6 实验运行时间和带宽

参数	电路类型	使用方法	协议	对比项\集合大小	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}
$\sigma = 32$	布尔电路	本文方法	PSU	时间(s)	4.507	7.556	14.453	26.284	54.794	119.470	279.850
				带宽(MB)	4.25	9.20	19.95	42.95	92.10	180.65	396.05
		文献[16]	PSU	时间(s)	–	–	–	–	–	–	–
				带宽(MB)	–	–	–	–	–	–	–
		文献[17]	PSI	时间(s)	4.257	7.368	13.576	26.531	52.066	113.44	263.26
				带宽(MB)	3.60	8.59	18.87	40.50	90.23	176.19	380.43
	算术电路	文献[21]	PSI	时间(s)	3.447	4.864	7.110	11.083	20.253	40.541	82.268
				带宽(MB)	0.89	1.58	3.30	6.59	13.33	26.41	53.72
		文献[6]	PSI	时间(s)	1.309	2.477	4.613	7.002	10.511	18.044	37.511
				带宽(MB)	1.09	2.22	3.83	6.74	14.19	27.88	52.87
$\sigma = 16$	布尔电路	本文方法	PSU	时间(s)	3.091	4.636	7.254	13.578	27.281	57.424	128.515
				带宽(MB)	2.10	4.70	9.92	21.54	46.05	98.19	208.89
		文献[16]	PSU	时间(s)	8.876	8.876	8.876	8.876	8.876	8.876	8.876
				带宽(MB)	9.30	9.30	9.30	9.30	9.30	9.30	9.30
		文献[17]	PSI	时间(s)	4.971	6.429	9.776	14.950	27.449	54.944	–
				带宽(MB)	1.75	4.23	9.03	20.97	45.05	82.22	–
	算术电路	文献[21]	PSI	时间(s)	3.197	4.713	7.038	11.970	19.013	40.225	81.573
				带宽(MB)	0.78	1.53	3.02	6.32	13.31	25.99	52.65
		文献[6]	PSI	时间(s)	1.374	2.382	4.233	6.630	10.413	17.921	37.608
				带宽(MB)	1.10	2.19	3.81	6.84	14.26	27.52	52.39

险。例如，当 $\sigma = 32$ 时，文献[16]方案的实验模型在本文实验环境下已经不能正常运行，但此时本文提出的方案仍然可以正常使用。因此，可以得出结论：针对于稀疏集合，本文方案的通信复杂度和计算复

杂度都优于布尔电路上文献[16]的方案。

(2)当前先进的PSI实验模型在运行时间和占用带宽上都优于本文方案。

5 结束语

基于布尔电路的隐私保护技术在近几年有了较快的发展。本文设计了保护隐私的集合合并电路、去重电路和混淆电路，通过在这些专用电路上应用YAO氏混淆电路估值技术，实现了保护隐私的集合并集运算协议。实验分析结果表明，本协议适用于稀疏集合并集的秘密计算。在本文研究的基础上，未来可进一步开展的研究包括保护隐私集合运算协议效率的进一步提高和设计抵抗恶意攻击者算法的设计。

参 考 文 献

[1] KISSNER L and SONG D X. Privacy-preserving set operations[C]. Advances in Cryptology- CRYPTO, Santa Barbara, USA, 2005: 241–257.doi: 10.1007/11535218_15.

[2] FRIKKEN K B. Privacy-preserving set union[C]. Applied Cryptography and Network Security, Zhuhai, China, 2007: 237–252. doi: 10.1007/978-3-540-72738-5_16.

[3] SEO J H, CHEON J H, and KAZA J. Constant-round multi-party private set union using reversed laurent series[C]. Proceedings of the 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, 2012: 398–412. doi: 10.1007/978-3-642-30057-8_24.

[4] MANY D, BURKHART M, and DIMITROPOULOS X. Fast private set operations with sepia[OL]. http://sepia.ee.ethz.ch/publications/setops_TIK-Report-345.pdf, 2012.

[5] KERSCHBAUM F. Outsourced private set intersection using homomorphic encryption[C]. Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, Seoul, Republic of Korea, 2012: 85–86. doi: 10.1145/2414456.2414506.

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

[7] KAMARA S, MOHASSEL P, RAYKOVA M, *et al*. Scaling private set intersection to billion-element sets[C]. Financial Cryptography and Data Security, Barbados, West Indies, 2014: 195–215.

[8] FREEDMAN J F, HAZAY C, NISSIM K, *et al*. Efficient set-intersection with simulation-based security[J]. *Journal of Cryptology*, 2016, 29(1): 115–155. doi: 10.1007/s00145-014-9190-0.

[9] PINKAS B, SCHNEIDER T, and ZOHNER M. Faster private set intersection based on OT extension[C]. Proceedings of the 23rd USENIX Security Symposium, San Diego, 2014: 797–812.

[10] HAZAY C. Oblivious polynomial evaluation and secure set-intersection from algebraic PRFs[C]. Theory of Cryptography Conference, Warsaw, Poland, 2015: 90–120. doi: 10.1007/978-3-662-46497-7_4.

[11] D'ARCO P, VASCO M I G, DEL POZO A L P, *et al*. Size-hiding in private set intersection: What can be done and how to do it without random oracles[OL]. https://eprint.iacr.org/2015/321. 2015.

[12] ZHENG Q J and XU S H. Verifiable delegated set intersection operations on outsourced encrypted data[C]. 2015 IEEE International Conference on Cloud Engineering, Tempe, AZ, USA, 2015: 175–184. doi: 10.1109/IC2E.2015.38.

[13] WANG T T, ZHU Y Q, and LUO X Z. Publicly verifiable delegation of set intersection[C]. 2014 International Conference on Cloud Computing and Internet of Things, Changchun, China, 2014: 26–30. doi: 10.1109/CCIOT.2014.7062500.

[14] LIU F, WEE K N, ZHANG W, *et al*. Encrypted set intersection protocol for outsourced datasets[C]. 2014 International Conference on Cloud Engineering, Boston, USA, 2014: 135–140.

[15] 夏峰, 杨波, 张明武, 等. 基于 LWE 的集合相交和相等的两方保密计算[J]. 电子与信息学报, 2012, 34(2): 462–467. doi: 10.3724/SP.J.1146.2011.00541.
XIA F, YANG B, ZHANG M W, *et al*. Secure two-party computation for set intersection and set equality problems based on LWE[J]. *Journal of Electronics & Information Technology*, 2012, 34(2): 462–467. doi: 10.3724/SP.J.1146.2011.00541.

[16] BRICKELL J and SHMATIKOV V. Privacy-preserving graph algorithms in the semi-honest model[C]. Advances in Cryptology-ASIACRYPT, Chennai, India, 2005: 236–252.

[17] HUANG Y, EVANS D, and KATZ J. Private set intersection: Are garbled circuits better than custom protocols?[C]. Proceedings of the 19th Network and Distributed Security Symposium, San Diego, CA, USA, 2012.

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

[19] KOLESNIKOV V, SADEGHI A R, and SCHNEIDER T. Improved garbled circuit building blocks and applications to auctions and computing minima[C]. Proceedings of the 8th International Conference on Cryptology and Network Security, Kanazawa, Japan, 2009: 1–20. doi: 10.1007/978-3-642-10433-6_1.

[20] KOLESNIKOV V and SCHNEIDER T. Improved garbled circuit: free XOR gates and applications[C]. International Colloquium on Automata, Languages and Programming, Reykjavik, Iceland, 2008: 486–498.

[21] DE CRISTOFARO E and TSUDIK G. Practical private set intersection protocols with linear complexity[C]. Financial Cryptography and Data Security, Tenerife, Canary Islands, 2010: 143–159. doi: 10.1007/978-3-642-14577-3_13.

[22] NAOR M and PINKAS B. Efficient oblivious transfer protocols[C]. Proceedings of the 12th Annual Symposium on Discrete Alogrithms, Washington, D.C., USA, 2001: 448–457.

孙茂华： 女，1986 年生，博士，讲师，研究方向为安全多方计算.

胡 磊： 男，1983 年生，博士，副教授，研究方向为安全多方计算.

朱洪亮： 男，1982 年生，博士，讲师，研究方向为信息安全.

李 祺： 女，1982 年生，博士，副教授，研究方向为信息安全.