高效的安全几何交集计算协议

朱国斌，谭元巍，赵 洋，熊 虎，秦志光

(电子科技大学计算机科学与工程学院 成都 611731)

【摘要】在不泄露各自私有信息的前提下，一组互不信任的参与者进行的多方合作计算叫做安全多方计算(secure multi-party computation, SMC)。而安全交集计算是安全多方计算一个重要的子问题，它主要解决如何通过协同计算求得交集并保证隐私安全的问题，该问题在社交网络、军事、商业领域有重要的应用前景。针对目前交集计算方法效率低下和计算复杂的特点，该文设计了一种高效安全的交集计算协议，该协议通过把集合中的每一个元素转换成平面空间中的点，再利用点与点的距离关系求得交集。最后通过仿真实验验证了协议的正确性、安全性和复杂性。

关 键 词 计算几何; 交集; 隐私保护; 安全多方计算

中图分类号 TN918 文献标志码 A doi:10.3969/j.issn.1001-0548.2014.05.026

An Efficient and Secure Geometric Intersection Computation Protocol

ZHU Guo-bin, TAN Yuan-wei, ZHAO Yang, XIONG Hu, and QIN Zhi-guang

(School of computer science and engineering, University of Electronic and Science Technology of China Chengdu 611731)

Abstract Secure multi-party computation (SMC) is a multi-party cooperative computation conducted by a group of participants on the premise that they do not trust each other and will not disclose any of their private information. Secure intersection computation (SIC), an important sub-concern of SMC, is focused on how to acquire intersection through cooperative computing and ensure the security of privacy, which has a significant application prospect in regard to social networks, military and commercial fields. In view of the low efficiency and complexity of current intersection computation methods, a high-efficient and secure intersection computation protocol is proposed. Under such a protocol, each element within a set will be converted into a dot in planar space and the intersection can be computed and acquired through the distance relations between these dots. At last, the correctness, security and complexity of the protocol are analyzed and verified through simulation experiment.

Key words computational geometry; intersection; privacy preserving; secure multi-parties computation

随着计算机网络的飞速发展，隐私交集计算(private set intersection, PSI)作为近年来一个新兴的研究领域，越来越引起人们的关注。它主要解决多个用户如何利用自己的私有信息合作计算，但并不泄露自己的私有信息给其他的计算参与者。

SMC最早由文献[1]提出，主要是为了在一组互不信任的参与者之间实现合作计算的目的。文献[2]进一步发展了SMC理论，奠定了其理论基础。目前该计算的应用领域已扩展到科学计算、统计分析、计算集合、数据挖掘等方向。

考虑场景：用户Alice和Bob都是社交网络上陌生用户，双方并不确定对方是否有共同好友，现在Alice和Bob想要找出他们的共同好友，但同时又不想让对方在此过程中知道自己有哪些好友。如果把两个用户各自的好友当做一个私有信息来处理，那么Alice和Bob各自拥有一个好友的集合，问题被抽象为两个集合的交集问题。交集保密计算是多方计算领域[3-4]中一个较特殊的问题，目前解决方案并不多，安全程度也有所不同，主要可分为以下几种方案。

1) 基于交换加密的协议

文献[5]提出了一种基于交换加密的PSI协议。该协议的安全性建立在decisional diffie-hellman (DDH)基础上，拥有线性的复杂度。但该协议是单向单方，只允许一方知道交集信息，另外该协议没有考虑恶意攻击。

文献[6]在文献[5]提出的协议基础上，利用幂函数作为交换加密协议，提出了一个应用于移动社交

收稿日期：2013－06－20；修回日期：2014–04–11

基金项目：广东省产学研重点项目(2012B091000054)，中央高校基本科研业务费(ZYGX2011J063)

作者简介：朱国斌(1981－)，男，博士，主要从事信息安全与密码学方面的研究.

网络(MSN)的匹配协议，通过扩展协议达到抵御恶意攻击的目的：1) 对集合元素授权，有效的阻止了恶意用户通过随意选择输入猜测集合元素；2) 协议的双方能够对等的获得交集信息；3) 考虑了几种由协议双方发起的恶意攻击。

2) 基于不经意多项式估值的协议

文献[7]提出了一种基于多项式估值和加法同态加密的协议。在该协议中，将集中的数据当做多项式的根构造出一个多项式，并对多项式系数同态加密。但在Freedman的协议中只有客户端能知道交集，而服务器不能获取任何信息。该协议同样也拥有线性的复杂度，适用于半诚实模型，不能抵御恶意攻击。为了克服该问题，提出了两个变种：一种适用于一方恶意、一方半诚实的场景；一种适用于双方恶意的场景。

还有一些利用多项式的数学性质来解决匹配问题的隐私保护方案。例如，文献[8]利用多项式代表多个集合。把该原理和加法的同态加密策略相结合，提出了一个对诚实但好奇(honest but curious, HBC)用户和对恶意用户安全的PSI和PCSI协议。

另外一种思路是在分布式环境扩展集合情况下操作。文献[9]扩展了Freedman协议到分布式环境中，在该方案中，一方的集合用一个多项式表示，再把多项式系数分发到多个服务器，通过密钥分享方案，实现了一种分布式解决方案。

3) 基于不经意伪随机函数的协议

文献[10]在Freedman协议基础上提出了两个协议，分别取得了恶意攻击和隐蔽攻击下的高效性和安全性。文献[11]提出了一种非常有效的协议，该协议用一个承诺密钥(在交互中使用相同密钥)计算伪随机函数，得到一个高效的协议，该结构的主要约束是伪随机函数的输入域大小应该是多项式。

目前的解决方案[2-11]主要是利用多项式以及同态加密等数学工具实现，计算开销大，步骤复杂。另外交集计算目前还未引入几何学，因此，利用几何学解决安全交集计算是一个非常有意义的研究。

1 预备知识

1.1 安全点积计算协议

问题描述：A和B分别拥有长度为d的向量$\boldsymbol{v}$和$\boldsymbol{u}$，s表示安全参数。求$\boldsymbol{v}$和$\boldsymbol{u}$的点积，但不能让对方知道各自拥有的向量。文献[12]给出了一种安全的点积计算协议。具体步骤如下。

1) A向B发起请求，B构造一个 $s\times(d+1)$ 维的矩阵$\boldsymbol{X}$。其中，第i行 $\boldsymbol{x}_i$ 定义为：

$$\boldsymbol{x}_i=\begin{cases}\langle u_1,u_2,\cdots,u_d,1\rangle & i=r\\ \boldsymbol{k}_i & \forall i\neq r\end{cases} \tag{1}$$

式中，r是随机选择的行；$\boldsymbol{k}_i$ 是随机生成的长度为d+1的向量。B再生成一个 $s\times s$ 维的的随机矩阵$\boldsymbol{Q}$、长度为d+1的随机向量$\boldsymbol{f}$ 和3个随机数 r_1,r_2,r_3。此外，

$$b=\sum_{i=1}^{s}Q_{ij} \tag{2}$$

$$\boldsymbol{c}=\sum_{i=1,i\neq r}^{s}\boldsymbol{x}_i^{\mathrm{T}}\sum_{i=1}^{s}Q_{ji} \tag{3}$$

$$\boldsymbol{D}=\boldsymbol{Q}\boldsymbol{X} \tag{4}$$

$$\boldsymbol{c}'=\boldsymbol{c}+r_1r_2\boldsymbol{f}^{\mathrm{T}} \tag{5}$$

$$\boldsymbol{g}=r_1r_3\boldsymbol{f} \tag{6}$$

式中，Q_{ij}表示矩阵$\boldsymbol{Q}$的第i行、第j列元素。B将计算的$\boldsymbol{D}$，$\boldsymbol{c}'$，$\boldsymbol{g}$发送给A。

2) A选择随机数 α，建立向量 $\boldsymbol{v}'=\langle v_1,v_2,\cdots,v_d,\alpha\rangle$，$v_i$ 表示向量$\boldsymbol{v}$的第i个元素。

$$\boldsymbol{y}=\boldsymbol{D}\boldsymbol{v}'^{\mathrm{T}} \tag{7}$$

$$z=\sum_{i=1}^{s}y_i \tag{8}$$

$$\alpha=z-\boldsymbol{c}'\boldsymbol{v}' \tag{9}$$

$$h=\boldsymbol{g}^{\mathrm{T}}\boldsymbol{v}' \tag{10}$$

A将计算后的 α 和 h 值发送给B。

3) B 计算 $\beta=\dfrac{\alpha+h(r_2/r_3)}{b}$，并发送给 A，A 将 α 发送给 B。

4) 计算 $\beta-\alpha$。

1.2 安全两数和平方计算协议

问题描述：A有一个实数 n_1，B有一个实数 n_2，A 和 B 希望保密计算 r_1 和 r_2，使满足 $r_1+r_2=(n_1+n_2)^2$。文献[14]给出了一种安全的两数平方和协议。具体步骤如下。

1) A计算 n_1^2，B计算 n_2^2。

2) A和B利用安全两方点积协议(向量长度为1)计算 n_1n_2，A得到 R_1，B得到 R_2，满足 $R_1+R_2=n_1n_2$。

3) A计算 $r_1=n_1^2+2R_1$，B计算 $r_2=n_2^2+2R_2$。

由 $r_1+r_2=n_1^2+2(R_1+R_2)+n_2^2=(n_1+n_2)^2$ 可知该协议是正确的；由安全两方点积协议的安全性可保证该协议的安全性；该协议只需使用一次安全两方点积协议，其计算复杂性和通信复杂性都依赖于使用的点积协议的计算复杂性和通信复杂性。

1.3 安全两点距离平方计算协议

问题描述：设A有保密点 $p_1(x_1,y_1)$，B有保密点

$p_2(x_2, y_2)$，*A*和*B*希望在不泄露各自保密点信息的情况下，计算两点之间距离的平方 $d_{p_1p_2}^2=(x_1-x_2)^2+(y_1-y_2)^2$，具体步骤如下。

1) *A*和*B*分别利用安全两数和平方协议，保密计算 $(x_1+(-x_2))^2$；计算完成后，*A*得到 r_1^A，*B*得到 r_1^B，满足 $r_1^A+r_1^B=(x_1+(-x_2))^2$。

2) *A*和*B*分别利用安全两数和平方协议，保密计算$(y_1+(-y_2))^2$；计算完成后，*A*得到 r_2^A，*B*得到 r_2^B，满足 $r_2^A+r_2^B=(y_1+(-y_2))^2$。

3) *A*计算 $r_1=r_1^A+r_2^A$，*B*计算 $r_2=r_1^B+r_2^B$。

此时，$r_1+r_2=(x_1-x_2)^2+(y_1-y_2)^2$，两点距离为 $d_{p_1p_2}=\sqrt{r_1+r_2}$。

该协议的正确性和安全性依赖于安全两数和平方协议的安全性和正确性，只需使用两次安全两数和平方协议，其计算复杂性和通信复杂性都依赖于安全两数和平方协议的计算复杂性和通信复杂性。

2　双向安全交集计算协议

2.1　问题描述

*A*拥有集合 $S_A=\{a_1,a_2,\cdots,a_n\}$，*B*拥有集合 $S_B=\{b_1,b_2,\cdots,b_n\}$，*A*和*B*同时希望找出交集IS，即 $S_A\bigcap S_B$，同时双方又不愿泄露任何非交集元素信息给对方，即*B*不知道任何 $\{x|x\in(S_A-\mathrm{IS})\}$，*A*不知道任何 $\{x|x\in(S_B-\mathrm{IS})\}$。

2.2　协议步骤

1) 将集合中的元素值拆分成两部分，分别表示点的横纵坐标，即一个元素值可以表示成一个点坐标。设集合S_A中的元素表示成点集合 $PS_A=\{p_{a_1},p_{a_2},\cdots,p_{a_n}\}$，集合$S_B$中的元素表示成点集合 $PS_B=\{p_{b_1},p_{b_2},\cdots,p_{b_n}\}$。

2) *A*发起协议，选取一个随机点 p_a，向*B*发出协议请求。

3) 对每一个点 $p_b\in S_B$，通过安全两点距离平方计算协议计算 p_a 与 p_b 的距离，*A*计算得到*n*个 r_1，即 $R_1=\{r_1^1,r_1^2,\cdots,r_1^n\}$，*B*计算得到*n*个 r_2，即 $R_2=\{r_2^1,r_2^2,\cdots,r_2^n\}$。

4) *A*将 R_1 发送给*B*。

5) *B*通过 R_2 和 R_1 计算 $d_{p_ap_b}$ 的集合 $D=\{d_1,d_2,\cdots,d_n\}$，即代表*B*中各点到 p_a 的距离；*A*本地计算自己集合 S_A 中元素与点 p_a 的距离的集合 $D'=\{d_1',d_2',\cdots d_n'\}$，即代表*A*中各点到 p_a 的距离。

6) *A*将集合 D' 中的元素做哈希运算，得到 $\mathrm{HD}'=\{\mathrm{H}(d_1'),\mathrm{H}(d_2'),\cdots,\mathrm{H}(d_n')\}$ (用 H(*m*) 表示对元素*m*进行哈希计算的结果)，然后把 HD′ 发送给*B*。*B*将集合*D*中的元素进行哈希运算，得到 $\mathrm{HD}=\{\mathrm{H}(d_1),\mathrm{H}(d_2),\cdots,\mathrm{H}(d_n)\}$，然后把HD发送给*A*。

7) *A*和*B*分别比较集合HD与 HD′ 中的元素，如果没有相同元素，则退出协议；如果有相同元素 $\mathrm{H}(d_i)=\mathrm{H}(d_j')$，则把二元组(*i*, *j*)放入集合*I*，继续运行协议。

8) *B*随机选取一个点 p_c，向*A*发出请求。

9) 对每一个点 $p_d\in S_B$，通过安全两点距离平方计算协议计算 p_c 与 p_d，*A*计算得到*n*个 r_1'，即 $R_1'=\{r_1^{1'},r_1^{2'},\cdots,r_1^{n'}\}$，*B*计算得到*n*个 r_2'，即 $R_2'=\{r_2^{1'},r_2^{2'},\cdots,r_2^{n'}\}$。

10) *B*将 R_2' 发送给*A*。

11) *A*通过 R_2' 和 R_1' 计算 $d_{p_cp_d}$ 的集合 $D''=\{d_1'',d_2'',\cdots,d_n''\}$，即代表*A*中各点到 p_c 的距离；*B*本地计算自己集合 S_B 中元素与点 p_c 的距离的集合 $D'''=\{d_1''',d_2''',\cdots d_n'''\}$，即代表*B*中各点到 p_c 的距离。

12) *B*将集合 D''' 中的元素进行哈希运算，得到 $\mathrm{HD}'''=\{\mathrm{H}(d_1'''),\mathrm{H}(d_2'''),\cdots,\mathrm{H}(d_n''')\}$，然后把 HD‴ 发送给*A*。*A* 将集合 $D^{"}$ 中的元素进行哈希运算，得到 $\mathrm{HD}''=\{\mathrm{H}(d_1''),\mathrm{H}(d_2''),\cdots,\mathrm{H}(d_n'')\}$，然后把 HD″ 发送给 *B*。

13) *A* 和 *B* 分别比较集合 HD″ 与 HD‴ 中的元素，如果没有相同元素，则退出协议；如果有相同元素 $\mathrm{H}(d_i'')=\mathrm{H}(d_j''')$，则把二元组(*i*, *j*)放入集合 I'。如果 *I* 和 I' 中有相同二元组(*i*, *j*)，则表示 PS_A 中第 *j* 个点与 PS_B 中第 *i* 个点是相同点，则点 $a_i\in\mathrm{IS},b_j\in\mathrm{IS}$。

2.3　正确性

在步骤5)和步骤11)中计算两点的距离，该步骤的正确性依赖于安全两点距离平方计算协议的正确性。

引理 1　如果点*x*是交集IS中的元素，$a_i=b_j=x$，则通过该协议能够得到交集元素*x*。

证明：因为*x*为交集中的元素，所以*A*和*B*双方计算得出的到定点的距离的哈希是相同的，所以步骤7得到的集合*I*非空，包含元素(*i*, *j*)。同样，在步骤13中得到的集合 I' 非空，一定包含元素(*i*, *j*)，所以能够得到交集元素*x*。

引理 2　如果点*x*不是交集IS中的元素，则不会通过该协议得到。

证明：*x*是S_A中的一个非交集元素，设点*x*到 p_c 的距离为*r*。如果存在点*y*(S_B中的元素)到点 p_c 的距离不等于*r*，则协议在步骤7)退出；如果*y*到 p_c 的距离等

于r，则该协议会再次选定一个点 p_d，并求点x，y到点 p_d 的距离。如果x到 p_d 的距离与y到 p_d 的距离不等，则不会出现在 I' 中；如果相等，则说明点x，y在以 p_d 为圆心的一个圆 O_{p_c} 上，也在以 p_c 为圆心的圆 O_{p_c} 上。两圆相交关系如图1所示。

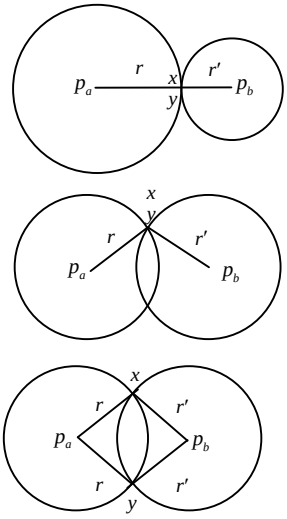


图1 两圆相交关系

当两圆 O_{p_c}，O_{p_d} 只有一个交点的时候，则$x=y$，与假设矛盾，不成立；当两圆 O_{p_c}，O_{p_d} 有两个交点，因为x不为交集元素，所以 $x \neq y$，x，y分别为两个不同的交点。由于到两点x，y距离相同的圆心只能在线段xy的中垂线上，而点 p_c，p_d 是随机选择的平面空间上的两个点，两点同时处于线段xy的中垂线的概率与平面半径的平方成反比，当平面半径足够大时，此种情况概率小到忽略不计。综上所述，点x不会出现在本协议的结果中。

2.4 复杂性

设$|S_A|=|S_B|=n$，本协议一共运行两轮求集合中点与选定点之间的距离，每轮计算运行一次点与点距离协议，最后本地计算哈希值集合的交集，由于主要计算开销在于计算点与点距离，计算复杂度依赖于两点距离计算复杂度，共计算$2n$次距离，计算复杂度为$O(n)$。通信复杂度主要开销在双方计算距离时传递 R_1，R_2 以及距离的哈希值，所以通信复杂度为$O(n)$。

3 安全性证明与实验

3.1 安全性证明

定义 1 在交集计算过程中，协议双方除了交集元素以外不能获得或推导出对方集合中的其他元素，则称协议是安全的。

证明：在协议中，计算双方获得对方信息，包括计算距离的中间值R_1或R_2以及对方集合中的点到随机选定点距离平方的哈希值，对于非交集中的点，如果存在情况：协议方A的某点 p_a 与另一方B的某点 p_b 与一个随机选定点O_1的距离平方哈希相等，而与另一个随机选定点O_2的距离平方哈希不等。则$A(B)$可以得知$B(A)$的某点与点 $p_a(p_b)$ 到O_1的距离相等，即在以O_1为圆心的同一个圆上。但在同一圆上的点不唯一，确定该点的位置是困难的，无法推导出点的具体位置。如果对于非交集元素则不存在上述情况，且假设安全两点距离平方和协议以及所用的哈希函数是安全的，那么计算双方不能从距离的平方的哈希值推导出距离的平方，从而获得任何交集元素以外的元素信息。综上，协议是安全的。

3.2 实验

实验环境如表1所示。

表1 实验环境

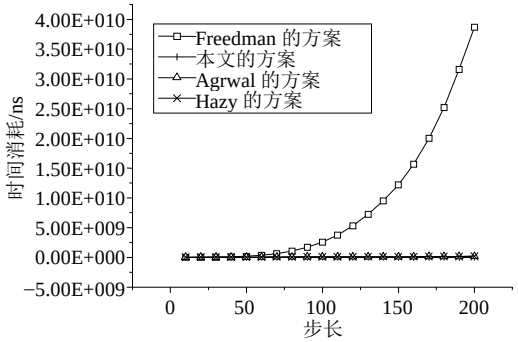
项目	数据
计算机数/台	1
操作系统	WIN7
内存/GB	2
主频/GHz	2.5
编程语言	C++
编程工具	Eclipse

为了验证协议的有效性和计算复杂性，实验选择了单向安全交集计算协议，并主要从两个部分进行测试：实验3.2.1验证了协议的线性计算复杂性；实验3.2.2验证了协议的有效性。

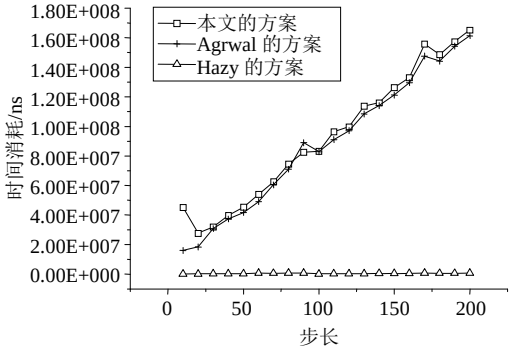
3.2.1 协议计算复杂度

协议运行时间主要由两个方面组成：协议运行双方本地计算时间和计算结果在网络中的传输时延。为了体现协议的真实计算复杂性，本实验在同一台主机上进行，忽略了传输时延。同时为了说明协议的效率，与引言中列出的3类协议进行了对比，协议时间消耗对比如图2所示。4种协议运行时间随集合元素个数的变化而变化的趋势图如图2a所示。从图中可以看到，Freedman协议的多项式方案时间消耗随集合大小的增加非线性增加，时间开销最大。后3种方法详细图如图2b所示，从图中看出，后3种方案运行时间基本上随集合大小的增加而线性增长。Agrwal方案和Hazy方案的时间消耗是相当的，但比起本文的方案的时间消耗要高得多，本文的方

案在时间消耗上是最优的。



a. 4种协议运行时间随集合元素个数的变化而变化的趋势图



b. 后三种方法详细图

图2　协议时间消耗对比

3.2.2　协议的正确性

设另选择两个不同点 p_a、p_b，最后通过协议错判为相同点，由协议具体内容可知两点到随机选择的两点 p_c、p_d 的距离相同。因此，点 p_c、p_d 在点 p_a、p_b 的中垂线上，而随机选择两点在给定两点的中垂线上的概率与平面空间面积成反比，与空间半径的平方成反比，即随着空间半径的增大，错误率按平方减小。为了验证协议的准确性，实验主要观察错误率随着空间半径增长的变化趋势。错误率随半径变化图如图3所示。

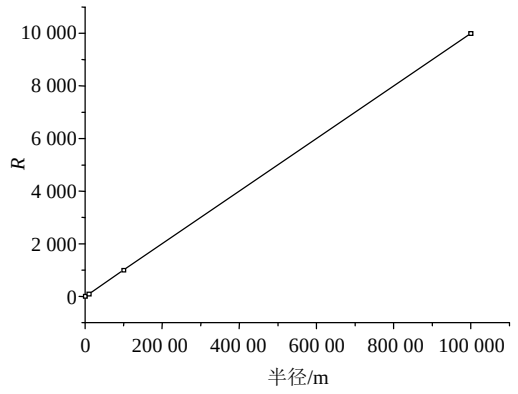


图3　错误率随半径变化图

图中，为了方便看出半径和错误率的关系，纵坐标R取值为错误率的平方根的倒数，可以看出半径和错误率平方根的倒数成线性关系，即错误率与半径的平方成反比。当平面半径取100 m时，错误率为0.012；当平面半径取1 000 m时，错误率为0.000 11；当平面半径取10 000 m时，错误率为0.000 001。可以推测出随着分布空间范围的变大，错误率无限趋近于0。当平面空间半径取一个较大值(如1×10^{10})时，错误率(1×10^{-20})小到忽略不计。

通过理论推导和实验数据可以得出：空间半径的平方和错误率大致成反比。

4　结 束 语

安全两点和平方计算协议可以在不泄露本方点信息的情况下得到两点的距离，通过两轮的计算集合中点到随意选定点的距离，从而确定两个集合的交集元素。本文提出了一个几何安全交集计算协议，同时，在求得交集元素的前提下，考虑了信息的安全性，保证了双方只知道交集元素而不知道任何交集外的信息。最后通过实验验证了协议的准确性和复杂性。

本文设计的协议复杂度低，开销小，下一步工作主要是研究如何进一步把协议应用到隐私保护的其他场景中去，并考虑如何发现抵御恶意攻击。

参　考　文　献

[1] YAO A C. Protocols for secure computations [C]// Proceedings of 23rd Annual IEEE Symposium on Foundations of Computer Science. Los Alamitos: IEEE Computer Society Press, 1982.

[2] EMILIANO D C, GENE T. Experimenting with fast private set intersection[J]. Trust and Trustworthy Computing, 2012, 7344: 55-73.

[3] EMILIANO D C, PAOLO G. Fast and private computation of cardinality of set intersection and union[J]. Cryptology and Network Security: Springer Berlin Heidelberg, 2012.

[4] DONG C, CHEN L, JAN C, et al. Fair private set intersection with a semi-trusted arbiter[J]. Data and Applications Security and Privacy XXVII, 2013, 7964: 128-144.

[5] AGRAWAL R, EVFIMIEVSKI A, SRIKANT R. Information sharing across private databases[C]// Proceedings of the 2003 ACM SIGMOD international conference on Management of data, [S.l.]: ACM, 2003.

[6] XIE Q, HENGARTNER U. Privacy-preserving matchmaking for mobile social networking secure against malicious users[C]//2011 Ninth Annual International Conference on Privacy, Security and Trust(PST), [S.l]: IEEE,

2011.

[7] STANISLAW J, LIU Xiao-min. Fast secure computation of set intersection[J]. Security and Cryptography for Networks, 2010, 6280:418-435.

[8] EMILIANO D C, KIM J. Linear-complexity private set intersection protocols secure in malicious model[J]. Advances in Cryptology-ASIACRYPT 2010, 2010, 6477: 213-231.

[9] YE Q, WANG H, PIEPRZYK J. Distributed private matching and set operations[J]. Information Security Practice and Experience, 2008, 4991: 347-360.

[10] HAZAY C, LINDELL Y. Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries[J]. Theory of Cryptography, 2008, 4948: 155-175.

[11] JARECKI S, LIU X. Efficient oblivious pseudorandom function with applications to adaptive ot and secure computation of set intersection[J]. Theory of Cryptography, 2009, 5444: 577-594.

[12] IOANNIDIS I, GRAMA A, ATALLAH M. A secure protocol for computing dot-products in clustered and distributed environments[C]//International Conference on Parallel Processing Proceedings, Washington DC: IEEE, 2002.

[13] WANG Ting, LUO Wen-jun. Design and analysis of private-preserving dot product protocol[C]//International Conference on Electronic Computer Technology, Macau, China: IEEE, 2009.

[14] 刘文, 罗守山, 杨义先, 等. 安全两方圆计算协议[J]. 北京邮电大学学报, 2009, 32(3): 32-35.
LIU Wen, LUO Shou-shan, YANG Yi-xian, et al. Safe two-party circle calculation protocol[J]. Journal of BUPT, 2009, 32(3): 32-35.

[15] 罗文俊, 李祥. 多方安全矩阵乘积协议及应用[J]. 计算机学报, 2005, 28(7): 1230-1235.
LUO Wen-jun, LI Xiang. Multi-party safe matrix product protocol and application[J]. Journal of computer, 2005, 28(7): 1230-1235.

[16] 符祖峰, 罗文俊, 童玲. 一个保护私有信息的线段与椭圆相交判定协议[J]. 计算机工程与应用, 2010, 46(17): 77-80.
FU Zu-feng, LUO Wen-jun, TONG Ling. A privacy-preserving protocol judging the intersecting between oval and line[J]. Engineering and application of computer, 2010, 46(17): 77-80.

编 辑 叶 芳