半诚实模型下安全多方排序问题的研究

肖 倩[1,3],罗守山[1,3],陈 萍[2],吴 波[4]
(1. 北京邮电大学软件学院,北京 100876;2. 北京邮电大学电信工程学院,北京 100876;
3. 西安电子科技大学综合业务网理论及关键技术国家重点实验室,陕西西安 710071;4. 北京邮电大学理学院,北京 100876)

摘 要: 安全多方排序问题是百万富翁问题的推广问题,用于 n 个参与方在不泄漏各方秘密输入的前提下比较出其输入在全体输入中按照一定顺序所处的位置.本文首先提出了半诚实模型下基于同态加密的安全两方排序协议.然后将该协议推广到多方排序的情况,并提出两种提高效率的改进算法.最后本文还提出了基于模糊贴近度的安全多方排序协议,并对这几个协议的安全性和效率做了分析、比较.

关键词: 百万富翁问题; 安全多方排序; 半诚实模型; 同态加密; 模糊贴近度
中图分类号: TN918.1 **文献标识码**: A **文章编号**: 0372-2112 (2008) 04-0709-06

Research on the Problem of Secure Multi-party Ranking Under Semi-honest Model

XIAO Qian[1,3], LUO Shou-shan[1,3], CHEN Ping[2], WU Bo[4]
(1. *School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing* 100876, *China*;
2. *School of Telecommunication Engineering, Beijing University of Posts and Telecommunications, Beijing* 100876, *China*;
3. *National Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, Shaanxi* 710071, *China*;
4. *School of Science, Beijing University of Posts and Telecommunications, Beijing* 100876, *China*)

Abstract: Secure multi-party ranking is a problem generalized from the millionaires' problem, which can be used by n people to know about their secrets' order among all their inputs without leaking further information. Through the study on millionaires' protocol, we presented a secure two-party ranking protocol under semi-honest model based on homomorphic encryption. Then we generalized it to secure multi-party ranking, and we presented two algorithms whose efficiency are both improved. Finally, we gave a secure multi-party ranking protocol based on fuzzy nearness degree, and we analyzed the efficiency and security of these protocols.

Key words: millionaires' problem; secure multi-party ranking; semi-honesty model; homomorphic encryption; fuzzy nearness degree

1 引言

百万富翁问题是安全多方计算中一个非常重要的分支,由该问题又引申出安全多方排序问题.最早由 A. Yao[1] 提出了一个重要的安全多方计算协议——百万富翁协议:两个百万富翁在不泄漏各自财富信息的前提下比较出他们谁更富有.将问题推广到多方,假设有 n 个参与方,每一方都拥有一个秘密输入 m_i,他们希望在不泄漏自己秘密输入的前提下得到其输入按照从大到小的顺序在这 n 个秘密输入中所处的位置,这就是安全多方排序问题.A. Yao 在文献[1]中提出的百万富翁协议的复杂度是指数级的,因此不实用.近年来人们针对现有方案效率低、不实用等问题,设计出各种能够提高效率的百万富翁协议[2-8],其中文献[7]中设计了一个常数复杂性的百万富翁协议,能够比较出两方秘密输入之间的关系究竟是"≥",还是"≤".

现在虽然已经有很多有效的百万富翁协议,但是能够将"相等"关系单独区分出来的协议却很贫乏.这就使百万富翁协议在安全多方排序问题上的运用变得十分困难.本文首先在半诚实模型下将文献[7]中的百万富翁协议进行推广,从"≥"和"≤"关系中将"等于"关系区分出来.然后将修改后的协议推广到安全多方排序问题上,并设计出改进算法来降低协议的复杂度.最后,本文基于模糊数学中模糊贴近度的概念和思想设计了另一

收稿日期:2007-04-09;修回日期:2007-10-19
基金项目:国家自然科学基金(No. 60642008);西安电子科技大学综合业务网理论及关键技术国家重点实验室开放课题(No. ISN7-01)

个安全多方排序协议.

2 准备知识

定义 1 (安全多方排序问题) 有 n 个参与方,每一方都拥有一个秘密输入,他们希望在不泄漏自己秘密输入信息的前提下,得到其输入按一定的顺序在这 n 个秘密输入中所处的位置的问题.

本文中,我们将对秘密输入按照从大到小的顺序排序.

定义 2 (半诚实模型)[9] 参与方将准确完成协议,但同时记录下所有中间结果,用以推导额外信息.

定义 3 (基于语义安全的加同态加密体制)[7] 设加密算法为 $E(\cdot)$,相应的解密算法为 $D(\cdot)$,其中加密密钥公开,解密密钥保密. 明文空间 $M \subseteq Z$, $E(\cdot)$ 满足下述两个性质:

①语义安全性:对任意两个消息 $m_1, m_2 \in M$,不存在任何多项式时间算法区分 $E(m_1), E(m_2)$;

②加法同态性:对任意消息 $m_1, m_2 \in M$,任意 $k \in Z$,若 $m_1 + m_2 \in M$,且 $km_1 \in M$,则 $D(E(m_1)E(m_2)) = m_1 + m_2$ 且 $D(E(m_1)^k) = km_1$.

3 半诚实模型下的两方排序协议

3.1 协议描述

假设 Alice,Bob 分别拥有秘密输入 a, b(以下都只考虑秘密输入为整数的情况). 他们希望不泄漏各自的秘密输入而比较出 a, b 的大小. 设加、解密算法 $E(\cdot)$, $D(\cdot)$ 满足基于语义安全的加同态性,本文首先将文献[7]中的一个常数复杂性的百万富翁协议作如下推广,从而完成比较:

协议 1 推广的两方排序协议

Step1 Alice 用自己的公钥计算 $c = E(a)$ 发送给 Bob.

Step2 Bob 随机选择整数 u_1, v_1, w_1,使 $|v_1 - w_1| < u_1$ 且 $u_1 > 0$,并用 Alice 的公钥计算 $X_1 = c^{u_1}E(v_1)$, $Y_1 = E(u_1 b + w_1)$ 发送给 Alice.

Step3 Alice 解密 $D(X_1) = u_1 a + v_1$, $D(Y_1) = u_1 b + w_1$,根据下述定理 1:若 $D(X_1) > D(Y_1)$,则 $a \geq b$,否则若 $D(X_1) < D(Y_1)$,则 $a \leq b$.

Step4 Alice 通知 Bob 再随机选择整数 u_2, v_2, w_2,使 $|v_2 - w_2| < u_2$, $u_2 > 0$ 且 $(v_2 - w_2)(v_1 - w_1) < 0$,并用 Alice 的公钥计算: $X_2 = c^{u_2}E(v_2)$, $Y_2 = E(u_2 b + w_2)$ 发送给 Alice.

Step5 Alice 解密 $D(X_2) = u_2 a + v_2$, $D(Y_2) = u_2 b + w_2$,根据下述定理 1:若 $D(X_2) > D(Y_2)$,则 $a \geq b$,否则若 $D(X_2) < D(Y_2)$,则 $a \leq b$.

Step6 Alice 分析两次比较的结果,根据下述判定法则一判断秘密输入 a, b 的大小:

判定法则一:(证明见下述定理 2)

①若 Step3 $D(X_1) > D(Y_1)$ Step5 $D(X_2) > D(Y_2)$ 则: $a > b$;

②若 Step3 $D(X_1) < D(Y_1)$ Step5 $D(X_2) < D(Y_2)$ 则: $a < b$;

③若 Step3 $D(X_1) > D(Y_1)$ Step5 $D(X_2) < D(Y_2)$ 则: $a = b$;

④若 Step3 $D(X_1) < D(Y_1)$ Step5 $D(X_2) > D(Y_2)$ 则: $a = b$.

Step7 Alice 把最终的比较结果告知 Bob.

3.2 协议分析

3.2.1 正确性

定理 1 设 $D(X_1) = u_1 a + v_1$, $D(Y_1) = u_1 b + w_1$,且满足 $|v_1 - w_1| < u_1$, $u_1 > 0$; a, b 为整数. 那么若 $D(X_1) > D(Y_1)$,则 $a \geq b$;若 $D(X_1) < D(Y_1)$,则 $a \leq b$.

证明 因 $(D(X_1) - D(Y_1))/u_1 = (a - b) + (v_1 - w_1)/u_1$,且 $|(v_1 - w_1)/u_1| < 1$,又 a, b 为整数,所以若 $a \neq b$,$|a - b|$ 为大于或等于 1 的整数,故 $|a - b| > |(v_1 - w_1)/u_1|$,则 $D(X_1) - D(Y_1)$ 的符号与 $a - b$ 的一致;当 $|a - b| = 0$ 时,$|a - b| = 0 < |(v_1 - w_1)/u_1|$,则 $D(X_1) - D(Y_1)$ 的符号与 $v_1 - w_1$ 的一致. 所以当 $D(X_1) > D(Y_1)$ 时,假设 $a < b$,根据上述结论知 $D(X_1) < D(Y_1)$,与 $D(X_1) > D(Y_1)$ 矛盾,故 $a \geq b$. 同理可证明若 $D(X_1) < D(Y_1)$,则 $a \leq b$.

定理 2 设 $D(X_1) = u_1 a + v_1$, $D(Y_1) = u_1 b + w_1$, $D(X_2) = u_2 a + v_2$, $D(Y_2) = u_2 b + w_2$ 且 u_1, v_1, w_1; u_2, v_2, w_2 满足协议 1 中的条件; a, b 为整数. 那么判定法则一成立.

证明 ①若 $D(X_1) > D(Y_1)$, $D(X_2) > D(Y_2)$,由上述定理 1 分别有 $a > b$ 或 $a = b$;若 $a = b$,则 $D(X_1) - D(Y_1)$, $D(X_2) - D(Y_2)$ 的符号分别由 $v_1 - w_1$, $v_2 - w_2$ 的符号决定,又 $(v_2 - w_2)(v_1 - w_1) < 0$,所以 $(D(X_1) - D(Y_1))(D(X_2) - D(Y_2)) < 0$,与题设矛盾,故 $a > b$; ③因 $D(X_1) > D(Y_1)$,由上述定理 1, $a \geq b$; $D(X_2) < D(Y_2)$,由上述定理 1, $a \leq b$,综上知 $a = b$. 同理可证 ②、④.

根据定理 1、2 知协议 1 的正确性显然.

3.2.2 安全性[7]

(1) Bob 接收到 $c = E(a)$ 后,因为 $E(\cdot)$ 是语义安全的,所以 Bob 无法从中获得任何有关 a 的信息.

(2) Alice 接收并解密 $D(X_1) = u_1 a + v_1$, $D(Y_1) = u_1 b + w_1$ 后, $D(X_1) - D(Y_1) = u_1(a - b) + (v_1 - w_1)$,由于 $u_1, (v_1 - w_1)$ 对于 Alice 来说未知,她无法从 $D(X_1) - D(Y_1)$ 中解出 $a - b$,进而 Alice 不知道 b 的取值;即使 v_1

$-w_1=0$,由于 $D(X_1)-D(Y_1)=u_1(a-b)$. 要求出 $a-b$ 相当于对大整数 $D(X_1)-D(Y_1)$ 进行分解,因此它是计算上安全的. Alice 接收并解密 $D(X_2)=u_2a+v_2$, $D(Y_2)=u_2b+w_2$ 后同理也不知道 b 的取值为多少.

(3) 如果 Alice 将两次得到的信息联立起来,有 6 个未知数,只有 4 个已知条件,所以她无法从中解出 b 的值,至多只能了解 b 的取值与 a 的大小关系.

3.2.3 效率分析

由于协议 1 是对常数复杂性的百万富翁协议的推广,由其内容知,协议 1 也具有常数复杂性 $O(1)$.

综上,我们只需要选择一种满足语义安全的加同态密码算法,就可以通过上述协议 1 完成两个保密数据的比较.

4 推广的两方排序协议应用于安全多方排序

4.1 协议 1 在安全多方排序中的直接应用

设加密和解密算法 $E(\cdot)$, $D(\cdot)$ 满足基于语义的加同态性,将协议 1 推广到 n 方分别拥有秘密输入 m_1, m_2, …, m_n 的比较上,并将参与方按照秘密输入从大到小排列,秘密输入相等时参与方按照角标从小到大排列. 安全多方排序的基本步骤如下:

协议 2: 安全多方排序协议

p_i, $i=1,2,\cdots,n-1$ 依次执行以下的步骤:

Step1 p_i 计算 $c_i=E_{k_i}(m_i)$ 发送给 p_j, $j=2,\cdots,n$, $j>i$. (以下,为书写简便,均用 $E(\cdot)$ 代替 $E_{k_i}(\cdot)$, $D(\cdot)$ 代替 $D_{k_i}(\cdot)$.)

Step2 p_j $(j=2,\cdots,n, j>i)$ 随机选择整数 $u_{ij}^{(1)}$, $v_{ij}^{(1)}$, $w_{ij}^{(1)}$, 使 $|v_{ij}^{(1)}-w_{ij}^{(1)}|<u_{ij}^{(1)}$, $u_{ij}^{(1)}>0$, p_j 计算: $X_{ij}^{(1)}=c_i^{u_{ij}^{(1)}}E(v_{ij}^{(1)})$, $Y_{ij}^{(1)}=E(u_{ij}^{(1)}m_j+w_{ij}^{(1)})$ 发送给 p_i.

Step3 p_i 解密: $D(X_{ij}^{(1)})=u_{ij}^{(1)}m_i+v_{ij}^{(1)}$, $D(Y_{ij}^{(1)})=u_{ij}^{(1)}m_j+w_{ij}^{(1)}$, $j=2,\cdots,n, j>i$. 若 $D(X_{ij}^{(1)})>D(Y_{ij}^{(1)})$, 则 $m_i\geq m_j$; 若 $D(X_{ij}^{(1)})<D(Y_{ij}^{(1)})$, 则 $m_i\leq m_j$.

Step4 p_i 通知每一方 p_j $(j=2,\cdots,n, j>i)$ 重新随机选择整数 $u_{ij}^{(2)}$, $v_{ij}^{(2)}$, $w_{ij}^{(2)}$, 使 $|v_{ij}^{(2)}-w_{ij}^{(2)}|<u_{ij}^{(2)}$, $u_{ij}^{(2)}>0$ 且 $(v_{ij}^{(2)}-w_{ij}^{(2)})(v_{ij}^{(1)}-w_{ij}^{(1)})<0$, 计算: $X_{ij}^{(2)}=c_i^{u_{ij}^{(2)}}E(v_{ij}^{(2)})$, $Y_{ij}^{(2)}=E(u_{ij}^{(2)}m_j+w_{ij}^{(2)})$ 发送给 p_i.

Step5 p_i 解密 $D(X_{ij}^{(2)})=u_{ij}^{(2)}m_i+v_{ij}^{(2)}$, $D(Y_{ij}^{(2)})=u_{ij}^{(2)}m_j+w_{ij}^{(2)}$, $j=2,\cdots,n, j>i$. 若 $D(X_{ij}^{(2)})>D(Y_{ij}^{(2)})$, 则 $m_i\geq m_j$; 若 $D(X_{ij}^{(2)})<D(Y_{ij}^{(2)})$, 则 $m_i\leq m_j$. p_i 综合第 3 步的结果可得到自己的秘密输入 m_i 与其他 $n-1$ 方的秘密输入的大小关系.

p_i 统计比 m_i 大的秘密输入 m_j 的个数记为 $|k_i|$, 并将与 m_i 相等的 m_j 所属参与方 p_j 加入集合 k_i', $|k_i'|$ 表示 k_i 集合中成员的个数 $(j=1,2,\cdots,n, j\neq i)$. 将 k_i 中的成员连同 p_i 按角标从小到大排列,设 p_i 处于这个排列的第 l 个位置,取出排头的成员 p_{j_1}, $K_i=K_{j_1}+l-1=|k_{j_1}|+l$ 即为 p_i 的秘密输入按从大到小的顺序在秘密输入队列中所处的位置.

根据协议 2 易知, p_k 与 p_{k+1}, …, p_n 共执行 $n-k$ 次协议 1 得到他在排列中的位置 $(k=1,2,\cdots,n)$, 故协议 2 总共需要执行 $n(n-1)/2$ 次协议 1,由于协议 1 的算法复杂度是常数级的,所以协议 2 的算法复杂度为 $O(n^2)$, 是平方阶的,比较高. 下面给出算法来降低协议 2 的算法复杂度.

4.2 算法的改进

符号约定:

$cmp(p_1,p_2)>0/<0/=0$: p_1 的秘密输入比 p_2 的大/小/相等.

$compare(p_1,p_2)$: p_1 与 p_2 执行一次协议 1.

4.2.1 改进算法一

(1) 算法描述

算法的过程可以参见图 1.

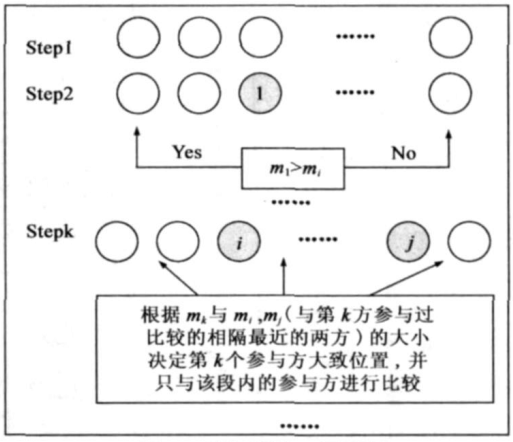

图 1 改进算法一示意图

Step1 p_1 与 p_2, …, p_n 各执行一次协议 1 比较出秘密输入的大小关系并告知对方,同时 p_1 得到在排列中所处的位置. 把已排过序的参与方标为 L_1.

Step2 p_k $(k=2,3,\cdots,n)$ 依次按照以下方式完成 m_k 的位置排序:

p_k 找出 L_1, …, L_{k-1} 中与其执行过协议 1 的参与方,若只有一个则记为 L_i, $i\in\{1,2,\cdots,k-1\}$, p_k 执行: ①若 $cmp(p_{L_i},p_k)>0$,则 $compare(p_k,p_j)$,其中 $cmp(p_{L_i},p_j)>0$, $j>k$, $j\in\{1,\cdots,n\}$; ②若 $cmp(p_{L_i},p_k)<0$,则 $compare(p_k,p_j)$,其中 $cmp(p_{L_i},p_j)<0$, $j>k$, $j\in\{1,\cdots,n\}$; ③若 $cmp(p_{L_i},p_k)=0$,设与 p_{L_i} 的秘密输入相等且角标最大(但其角标比 k 小)的成员为 p_{L_m},则 p_k 在排列中所处的位置为 p_{L_m} 在排列中所处位置 K_{L_m} 加 1,即 $K_k=K_{L_m}+1$.

若 L_1, …, L_{k-1} 中与 p_k 一起执行过协议 1 的参与方

至少有两个,找出其中相隔最近的两个参与方 L_i, L_j, $i<j$ 如果有多对,任取一对,完成以下步骤:p_k 根据 m_k 与 m_{L_i}, m_{L_j} 的相对大小大致确定自己的位置:①在 L_i 之前,则 $compare(p_k, p_m)$,其中 $cmp(p_{L_i}, p_m)<0$;②在 L_i, L_j 之间,则 $compare(p_k, p_m)$,其中 $cmp(p_{L_j}, p_m)<0$, $cmp(p_{L_i}, p_m)>0$;③在 L_j 之后,则 $compare(p_k, p_m)$,其中 $cmp(p_{L_j}, p_m)>0$;以上 $m\in\{1,2,\cdots,n\}$.

p_k 得到 m_k 在排列中所处的位置,同时 p_k 将比较结果告知对方;将已确定位置的参与方重新从前向后标号为 $L_1, L_2, \cdots, L_k$.

Step3 当 n 方都执行完 Step2 后,新队列 $L_1, L_2, \cdots, L_n$ 就是 n 个参与方按秘密输入从大到小排成的队列.

(2) 算法 1 的复杂度分析

①最差的情况:

$m_1, m_2, \cdots, m_n$ 恰好从大到小或从小到大排列,这时的复杂度为 $O(n^2)$,没有改进.

②几种特殊情况:

二分的情况:假设在给 n 个参与方依次排序时,p_1 恰好位于已排序列的中间位置,将序列分成等长的两部分 l_1, l_2;p_2, p_3 分别恰好位于 l_1, l_2 的中间位置,分别将 l_1, l_2 分成等长的两部分:$l_3 \sim l_6$; ……

为计算方便,设 $1+2+2^2+\cdots+2^{s-1}=n$,那么,n 个参与方在总协议完成后分别执行协议 1 的次数是:

$1: p_1: n-1$ 次;

$2: p_2, p_3: (n-1-2)/2$ 次;

……

$s: (n-1-2-2^2-2^3-\cdots-2^{s-1})/2^{s-1}$ 次.

所以改进算法一中总共执行协议 1 的次数为:

$$\frac{n-1}{2^0}+\cdots+\frac{n-1-2-2^2-\cdots-2^{s-1}}{2^{s-1}}=(n+1)s-2^{s+1}+2 \tag{1}$$

又 $1+2+2^2+\cdots+2^{s-1}=n$,即 $s=\log_2(n+1)$. 代入式(1),得到 $(n+1)\log_2(n+1)-2n$ 为执行协议 1 的总次数,算法复杂度为 $O((n+1)\log_2(n+1))$.

同理,针对三分的情况,可估算出其执行协议 1 的次数为 $\frac{5}{3}(n+1)\log_3(2n+1)-4n$,算法复杂度为 $O((n+1)\log_3(2n+1))$.

依此类推,p 分的情况,可以求得算法复杂度为 $O((n+1)\log_p((p-1)n+1))$,是线性对数阶的.

借助 Matlab 软件分析发现当 n 取定时,算法复杂度都随 p 的增大而减小. 即 p 分的情况下,p 越大,算法复杂度越小.

4.2.2 改进算法二

(1) 算法描述

Step1 p_2 与 p_1 执行一次协议 1,确定他与 p_1 所处的相对位置:若 $m_2 \le m_1$, p_2 站 p_1 后面;否则,p_2 站 p_1 前面;将已排队列从前到后重新标号为 L_1, L_2.

Step2 $p_k(k=3, \cdots, n)$ 依次执行以下步骤:

p_k 与 L_{k-1} 执行一次协议 1,确定他与 L_{k-1} 的相对位置:若 $m_k \le m_{L_{k-1}}$, p_k 站 L_{k-1} 后面;若 $m_k > m_{L_{k-1}}$, p_k 与 L_1 执行一次协议 1,根据结果选择 p_k 的位置:如果 $m_k > m_{L_1}$, p_k 站 L_1 前面;如果 $m_k \le m_{L_1}$,则:①若 L_1, L_{k-1} 之间还有两个或以上参与方,将 $L_2, \cdots, L_{k-2}$ 作为一个新的已排序列转 Step2;②若 L_1, L_{k-1} 之间只有一个参与方 L_2,若 $m_k \le m_{L_2}$,将 p_k 插入 L_2, L_{k-1} 之间,若 $m_k > m_{L_2}$,将 p_k 插入 L_1, L_2 之间;③若 L_1, L_{k-1} 之间没有其他参与方,将 p_k 插入 L_1, L_{k-1} 之间;将已排队列从前到后重新编号 $L_1, \cdots, L_k$.

Step3 当 n 个参与方都执行完 Step2 后形成的新队列 $L_1, \cdots, L_n$ 就是 $p_1, \cdots, p_n$ 按秘密输入从大到小顺序排列以后的队列.

(2) 算法 2 的复杂度分析

分析以上算法,易知:p_k 在这个算法中最多执行 $k-1$ 次协议 1,$k=1,2,\cdots,n$,那么,在最坏的情况下总共执行 $n(n-1)/2$ 次协议 1,算法复杂度是 $O(n^2)$,与未改进前相同.

最好的情况:如果参与方的输入就是按照从大到小的次序排列的. 那么在改进算法二中除 p_1 外每一方只需主动执行一次协议 1,而 p_1 主动执行零次协议 1. 那么该协议总共执行了 $n-1$ 次协议 1,算法的复杂度降低到一阶 $O(n)$.

一般情况下算法复杂度介于 $O(n)$ 和 $O(n^2)$ 之间,一般比未改进之前均有改善.

4.3 结合模糊贴近度的安全多方排序

由于协议 2 中两两参与方之间需要执行一次协议 1 造成很高的算法复杂度,下面考虑结合模糊数学中模糊贴近度的定义来实现安全多方排序.

4.3.1 相关概念的介绍

定义 4 (模糊贴近度的公理化定义)[10] 映射 $\sigma: F(U)\times F(U)\to[0,1]$, $(A,B)\mapsto\sigma(A,B)$,若 σ 满足:①$\sigma(A,A)=1$, $\sigma(\Phi,U)=0$;②$\sigma(A,B)=\sigma(B,A)$;③若 $A\subseteq B\subseteq C$,则 $\sigma(A,C)\le\sigma(A,B)\wedge\sigma(B,C)$,则称 σ 是 $F(U)$ 上的贴近度函数,$\sigma(A,B)$ 称为 A, B 的贴近度. 其中,$F(U)$ 表示论域 U 上的全体模糊集.

在本文中,我们定义这样的贴近度函数:

$$\sigma(A,B)=\begin{cases}0; & A=\Phi, B=U/A=U, B=\Phi\\ 1-c\cdot d(A,B); & \text{其他情况}\end{cases}$$

$A=(a_1, a_2, \cdots, a_l)$, $B=(b_1, b_2, \cdots, b_l)$

其中, $d(A,B)=\sum_{i=1}^{l}|a_i-b_i|\times 2^{l-i}$, c 值由其中一方随机选定.

贴近度表征两个事物的相似程度,贴近度越大表示两个事物越相似,否则表示它们差别越大.

4.3.2 结合模糊贴近度的安全多方排序

(1) 协议描述

设加密和解密算法 $E(\cdot)$, $D(\cdot)$ 满足基于语义安全的加同态性,结合模糊贴近度的安全多方排序协议的基本步骤如下:

协议 3　结合模糊贴近度的安全多方排序协议

Step1　p_1 与 $p_2,\cdots,p_n$ 各执行一次协议 1,得到自己在有序排列中的位置,并将 p_1 与 p_i 的比较结果告知 $p_i(i=2,\cdots,n)$.

Step2　$p_i(i=2,\cdots,n)$ 首先根据自身秘密输入与 p_1 秘密输入的大小初步确定自己的位置: $m_i>m_1$, p_i 站在 p_1 前面; $m_i\leq m_1$, p_i 站在 p_1 后面.

Step3　$p_i(i=2,\cdots,n)$ 与 p_1 分别完成下列操作:

① p_1, p_i 分别将自己拥有的秘密输入转化成二进制的形式: $m_1=a_{11}a_{12}\cdots a_{1l}$, $m_i=a_{i1}a_{i2}\cdots a_{il}$. ($l$ 是 $p_1,\cdots,p_n$ 事先商定的足够大的数,若转化成的二进制形式长度小于 l,则在其左边补充足够的 0.)

② p_1, p_i 按照如下步骤保密计算 $d(m_1,m_i)$,使两方得到的输出之和为 $d(m_1,m_i)$:

p_i 将其秘密输入 m_i 的全部奇数位发送给 p_1, p_1 计算:

$$s_{1i}=\sum_{j}|a_{1j}-a_{ij}|\times 2^{l-j},\ j=2k+1,\ 0\leq k\leq\lfloor(l-1)/2\rfloor$$

p_1 将其秘密输入 m_1 的全部偶数位发送给 p_i, p_i 计算:

$$s_{2i}=\sum_{j}|a_{1j}-a_{ij}|\times 2^{l-j},\ j=2k,\ 0\leq k\leq\lfloor l/2\rfloor$$

Step4　完成 Step3 后, p_1, p_i 分别拥有保密数据 s_{1i}, s_{2i},且 $s_{1i}+s_{2i}=d(m_1,m_i)$. p_1 随机选择一个足够大的数 R, R 作为贴近度定义中的 c 将被应用在 p_1 与所有 p_i 贴近度的计算中,且它能使贴近度保持在 $[0,1]$ 范围内. 然后 $p_i(i=2,\cdots,n)$ 与 p_1 完成下列操作:

① p_i 选择一个公钥/私钥对,公钥公开、私钥保密. 然后 p_i 利用满足语义安全的加同态性质的加密算法加密 s_{2i},得到 $E_{p_i}(s_{2i})$,将 $E_{p_i}(s_{2i})$ 发送给 p_1;

② p_1 用 p_i 的公钥加密 s_{1i},得到 $E_{p_i}(s_{1i})$,然后 p_1 计算 $E_{p_i}(s_{2i})^R$ 以及 $E_{p_i}(s_{1i})^R$,再把两个结果相乘得到 $E_{p_i}(R\times s_{2i}+R\times s_{1i})$ 发送给 p_i;

③ p_i 用自己的私钥解密 $E_{p_i}(R\times s_{2i}+R\times s_{1i})$ 得到 $R\times s_{2i}+R\times s_{1i}$,并计算 $\sigma(m_1,m_i)=1-(R\times s_{1i}+R\times s_{2i})$.

Step5　p_1 之前的参与方按照他们各自与 p_1 贴近度的大小从小到大排列,形成队列 L_1; p_1 之后的参与方按照他们各自与 p_1 贴近度的大小从大到小排列,形成队列 L_2(注意:在 $p_2,\cdots,p_n$ 按照贴近度比较大小时,它们各自的贴近度只在 $p_2,\cdots,p_n$ 之间公开,而不泄漏给 p_1,否则 p_1 可以从中计算出其他参与方的秘密输入). $L=\{L_1,p_1,L_2\}$ 就是 $p_1,p_2,\cdots,p_n$ 按照秘密输入从大到小排成的队列.

(2) 协议分析

ⅰ. 正确性

可以验证本文中定义的贴近度满足贴近度的公理化定义:

验证　第一二条性质容易验证,本文略;下面验证第三条性质:

$\forall A=(a_1,\cdots,a_l)$, $B=(b_1,\cdots,b_l)$, $C=(c_1,\cdots,c_l)$,若 $A\subseteq B\subseteq C$,则 $\forall i\in\{1,\cdots,l\}$, $a_i\leq b_i\leq c_i$,所以 $|a_i-c_i|\geq|a_i-b_i|\vee|b_i-c_i|$, $\vee$为两者中取大,故

$$\sigma(A,C)=1-c\sum_{i=1}^{l}|a_i-c_i|\times 2^{l-i}\leq(1-c\sum_{i=1}^{l}|a_i-b_i|\times 2^{l-i})\wedge(1-c\sum_{i=1}^{l}|b_i-c_i|\times 2^{l-i})$$

$$=\sigma(A,B)\wedge\sigma(B,C)$$

$\wedge$为两者中取小. 由于模糊贴近度表征了 A, B 之间的相似程度,因此任两个站在 p_1 同一边的参与方 p_i, p_j(不妨设他们站在 p_1 前面),那么如果 $\sigma(p_1,p_i)<\sigma(p_1,p_j)$,说明 p_1 与 p_j 更接近,所以 $p_i>p_j$;同理可以解释其他情况. 因此,协议 3 是正确的.

ⅱ. 安全性

p_1, p_i 分别拥有 s_{1i}, s_{2i},求模糊贴近度的问题转化为 p_1, p_i 共同安全计算 $R\cdot(s_{1i}+s_{2i})$ 的问题: p_i 将 $E_{p_i}(s_{2i})$ 发送给 p_1 时,由于 p_1 不知道 p_i 的私钥, p_1 无法解密 $E_{p_i}(s_{2i})$ 从而得到 s_{2i}的信息; p_1 利用语义安全的加密算法的加同态性计算 $E_{p_i}(R\times s_{2i}+R\times s_{1i})$, p_i 解密后得到 $R\times s_{2i}+R\times s_{1i}$,由于 p_i 不知道 R,所以他无法从中得知 s_{1i}的信息,因此他们最终能够安全地完成模糊贴近度的计算. 因此协议是安全的.

ⅲ 效率分析

首先该协议执行了 $n-1$ 次协议 1,复杂度为 $O(n)$. 然后, p_1 执行了 $n-1$ 次复杂度为 $O(1)$ 的连加运算, $p_i(i=2,\cdots,n)$ 执行一次复杂度为 $O(1)$ 的连加运算,故复杂度为 $O(n)$. 最后, p_1 执行 $n-1$ 次乘法运算. 综上所述,协议 3 的算法复杂度为 $O(n)$. 其算法复杂度始终是一阶的,故相对比较低.

4.4 安全多方排序协议的比较

我们可以通过下面的表 1 对上述几个安全多方排

序协议的效率和安全性作一比较:

表 1 各协议和算法的效率、安全性比较

协议和算法	算法效率	安全性
协议 2	$O(n^2)$ 效率低	每一方都不知道其他方秘密输入的具体值,只知道其他方的秘密输入与自己的大小关系;每一方都只知道自己在排列中的具体位置,而不知道其他方在排列中的位置和其他任何两方的秘密输入之间的大小关系;在四种算法中,其安全性最高.
改进算法一	$O((n+1)\log_p((p-1)n+1)) \sim O(n^2)$ 效率较低	每一方都不知道其他方秘密输入的具体值,只知道其他方的秘密输入与自己的大小关系;每一方虽不完全知道其他方在排列中的具体位置,但后参与排序的参与方能得到其中一个已排好序的参与方的位置信息;其安全性较高.
改进算法二	$O(n) \sim O(n^2)$ 在一些特殊情况下能够获得较高的效率	每一方都不知道其他方秘密输入的具体值,只知道其他方的秘密输入与自己的大小关系;但后参与排序的参与方能够得知已排好序的参与方之间的大小关系. 其安全性在四种算法中比较低,但是已经满足协议要求的安全性.
协议 3	$O(n)$ 效率始终较高	每一方都不知道其他方秘密输入的具体值,只知道其他方的秘密输入与自己的大小关系;最终,每一方都能知道其他方在排列中的具体位置;在四种算法中,其安全性最低,但已满足协议要求的安全性.

5 结束语

目前对于安全多方排序问题的研究还比较贫乏,没有很多有效的解决方案. 本文给出了一个安全多方排序协议,并通过两种改进算法提高了效率,然后本文基于模糊贴近度的定义给出了另一个安全多方排序协议,并对协议的正确性和安全性作了说明. 正如文献[7,9]等中的大多数协议一样,协议被设计在半诚实模型下,但协议的复杂度是比较低的.

参考文献:

[1] A Yao. Protocols for secure computation[A]. Proceeding of the 23th IEEE Symposium on Foundations of Computer Science [C]. Los Alamitos, CA: IEEE Computer Society Press, 1982. 160 - 164.

[2] C Cachin. Efficient private bidding and auctions with an oblivious third party[A]. Proceedings of the 6th ACM Conference on Computer and Communications Security[C]. New York: ACM Press, 1999. 120 - 127.

[3] H Y Lin, W G Tzeng. An efficient solution to the millionaires problem based on homomorphic encryption[A]. Proceedings of the 4th International Conference on Applied Cryptography and Networks Security[C]. volume 3531 of LNCS: 2005. 456 - 466.

[4] 秦静,张振峰,冯登国,李宝. 无信息泄漏的比较协议[J]. 软件学报,2004,15(3):421 - 427.
Qing Jing, Zhang Zhen-feng, Feng Deng-guo, Li Bao. A protocol of comparing information without leaking[J]. Journal of Software, 2004, 15(3): 421 - 427. (in Chinese)

[5] Ronald Fagin, Moni Naor, Peter Winkler. Comparing information without leaking it[J]. Communications of the ACM, 1996, 39(5): 77 - 85.

[6] 李顺东,戴一奇,游启友. 姚氏百万富翁问题的高效解决方案[J]. 电子学报,2005,33(5):769 - 773.
Li Shun-dong, Dai Yi-qi, You Qi-you. An efficient solution to yao's millionaires' problem[J]. Acta Electronica Sinica, 2005, 33(5): 769 - 773. (in Chinese)

[7] 秦波,秦慧,周克复,王晓峰,王育民. 常数复杂性的百万富翁协议[J]. 西安理工大学学报,2005,21(2):149 - 152.
Qin Bo, Qin Hui, Zhou Ke-fu, Wang Xiao-feng, Wang Yu-ming. Millionaires' protocol with constant complexity[J]. Journal of Xi'an University of Technology, 2005, 21(2): 149 - 152. (in Chinese)

[8] Ioannidis I, Grama A. An efficient protocol for Yao's millionaires' problem[A]. Proceedings of the 36th Annual Hawaii International Conference on System Sciences[C]. 2003. 6pp.

[9] 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. The secure multi-party protocol of matrix product and its application[J]. Chinese Journal of Computers, 2005, 28(7): 1230 - 1235. (in Chinese)

[10] 李洪兴,汪培庄. 模糊数学[M]. 北京:国防工业出版社,1994.

作者简介:

肖 倩 女,1984 年 9 月出生于湖南长沙,2007 年毕业于北京邮电大学理学院数学与应用数学系,并进入北京邮电大学软件学院计算机科学与技术专业,现为硕士研究生. 从事信息安全、安全多方计算方面的有关研究.
E-mail: xiaoqianbupt@gmail.com

罗守山 男,1962 年 8 月出生于北京市,1985 年、1994 年和 2001 年分别在北京师范大学、北京邮电大学和北京邮电大学获理学学士、理学硕士和工学博士学位,现为北京邮电大学教授. 博士生导师,主要从事信息安全和网格安全等方面的研究工作. E-mail: buptlou@263.net