软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2023,34(11):5408−5423 [doi: 10.13328/j.cnki.jos.006737]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

分布式数据集极差与极值和的保密计算*

李顺东, 家珠亮, 赵雪玲

(陕西师范大学 计算机科学学院, 陕西 西安 710119)

通信作者: 李顺东, E-mail: shundong@snnu.edu.cn

摘　要: 随着信息通信技术的不断突破与发展, 信息获取变得非常便利. 与此同时, 隐私信息也更容易泄露. 将智能领域与安全多方计算技术相结合, 有望解决隐私保护问题. 目前, 安全多方计算已经解决了许多不同隐私保护问题, 但还有更多的问题等待人们去解决. 对于极差、极值和的安全多方计算问题目前研究的结果很少, 极差、极值和作为统计学的常用工具在实际中有广泛的应用, 研究极差、极值和的保密计算具有重要意义. 提出新编码方法, 用新编码方法解决了两种不同的安全多方计算问题, 一是极差的保密计算问题, 二是极值和的保密计算问题. 新编码方法结合 Lifted ElGamal 门限密码系统, 设计多方参与、每方拥有一个数据场景下分布式隐私数据集极差的保密计算协议; 将新编码方法稍作改动解决相同场景下保密计算极值和的问题. 以此为基础, 对新编码方法进一步修改, 结合 Paillier 密码系统设计了两方参与、每方拥有多个数据情况下分布式隐私数据集极差、极值和的保密计算协议. 用模拟范例方法证明协议在半诚实模型下的安全性. 最后, 用模拟实验测试协议的复杂性. 效率分析和实验结果表明所提协议简单高效, 可广泛用于实际应用中, 是解决其他很多安全多方计算问题的重要工具.

关键词: 安全多方计算; 两方保密计算; 编码方法; 极差; 极值和

中图法分类号: TP309

中文引用格式: 李顺东, 家珠亮, 赵雪玲. 分布式数据集极差与极值和的保密计算. 软件学报, 2023, 34(11): 5408−5423. http://www.jos.org.cn/1000-9825/6737.htm

英文引用格式: Li SD, Jia ZL, Zhao XL. Secure Computation of Range and Sum of Extremums on Distributed Datasets. Ruan Jian Xue Bao/Journal of Software, 2023, 34(11): 5408−5423 (in Chinese). http://www.jos.org.cn/1000-9825/6737.htm

Secure Computation of Range and Sum of Extremums on Distributed Datasets

LI Shun-Dong, JIA Zhu-Liang, ZHAO Xue-Ling

(School of Computer Science, Shaanxi Normal University, Xi'an 710119, China)

Abstract: Due to the continuous breakthrough and development of information and communication technologies, information access has become convenient on the one hand. On the other hand, private information is now easier to leak than before. The combination of the intelligent field and secure multiparty computation (SMC) technology is expected to solve privacy protection problems. Although SMC has solved many different privacy protection problems so far, problems that remain to be settled are numerous. Research results about the SMC of range and the sum of extremums are currently seldom reported. As a common statistical tool, range and sum of extremums have been widely used in practice. Therefore, the secure computation of range and the sum of extremes are of great research significance. This study proposes a new encoding method and solves two types of SMC problems by the method: One is the secure computation of range, and the other is that of the sum of extremums. The new encoding method is combined with the Lifted ElGamal threshold cryptosystem to design a secure range computation protocol for distributed private datasets in the scenario in which multiple parties participate and each party has one data. Then, the new encoding method is slightly modified for the secure computation of the sum of extremums in the same scenario. On this basis, the study further modifies the new encoding method and combines it with the Paillier cryptosystem to design a protocol for the secure computation of range and the sum of extremums on distributed private datasets in the scenario in which two parties participate and each party has more than one data. Furthermore, this study proves that the proposed protocols are secure in the semi-honest

* 收稿时间: 2021-11-29; 修改时间: 2022-02-21, 2022-04-03; 采用时间: 2022-06-24; jos 在线出版时间: 2023-05-18
CNKI 网络首发时间: 2023-05-19

model with the simulation paradigm. Finally, the complexities of these protocols are tested by simulation experiments. The results of the efficiency analysis and experiments show that the simple and efficient proposed protocols can be widely used in practical applications and are important tools for solving many other SMC problems.

Key words: secure multiparty computation (SMC); secure two-party computation; encoding method; range of extremums; sum of extremums

1 引 言

安全多方计算是指互不信任的两方或多方参与者合作计算解决一个问题, 计算结束后各个参与者除了得到提前设定的计算结果外, 不获得其他参与者的任何隐私信息. 两方安全计算问题首先被 Yao 在 1982 年提出[1], 随后 Goldreich 等人提出了多方安全计算问题[2]. 经过多年来众多学者对该问题的深入研究与钻研, 安全多方计算现已成为国际密码学界热点问题[3−6], 形成了较为完备的理论体系, 被用于解决信息时代下各类隐私保护实际问题: 保密的科学计算问题[7−10]、保密的集合计算问题[11,12]、保密的数据挖掘问题[13]、保密的几何计算问题[14]、保密的深度学习问题[15,16]及其他安全多方计算实际应用问题[17−20].

极差是指一组数据中最大值与最小值之间的差距, 即最大值减最小值后所得的数值. 在心理学中, 极差也称为全距, 是一个重要的心理学测量指标, 利用全距有助于提早发现心理疾病, 以便采取预防措施; 工业生产中, 系统数据的极差反应系统稳定程度; 统计学中, 极差常用来描述一组数据的离散程度, 能体现数据的波动程度. 极差的保密计算在保密敏感数据, 保密判断金融交易风险, 保密优化生产链等各种实际社会生活中有重要的应用. 例如: (1) 在我国脱贫攻坚战中, 调查某一区域内人口的贫富差距是常有的工作, 但是被调查者通常不想透露自己的具体资产状况 (数值), 通过极差的保密计算就可以在保护被调查者个人财产隐私安全的情况下完成贫富差距的调查. (2) 某一涉及多方行业的项目招标时要求不同类型的公司先组队再投标, 多家有意愿投标的公司想在不泄露自方预期价情况下决定是否组队, 就需要保密计算 (预期价的) 极差来作为是否组队投标的参考.

极值和是指一组数据中最大值与最小值之和. 极值和在各种赛事的评分系统中常被用到: 比如在各种国际赛事中, 裁判给出评分后, 为了公平起见一般会去掉一个最高分一个最低分, 然后再计算参赛选手的最终分数. 隐私保护各个裁判给出的分数, 可以促进良性竞技, 维护公平友好竞技环境. 当然, 实现隐私保护各个裁判给出的分数同时应遵循以往评分规则, 在这个过程中, 保密计算最大值与最小值之和是重要工作.

据我们所知, 目前还没有关于极值和保密计算问题的研究, 关于极差保密计算问题的研究只有文献 [21]. 理论上这两个不同的问题都可以归约到最大 (最小) 值的保密计算问题, 目前有关最值的保密计算方面已有不少成果[21−28], 文献 [21−24] 中的方案都借助了外包云手段和诚实第三方服务器, 平台的构建对技术及计算开销有着较高要求, 且文献 [21] 采用的是全同态加密系统, 目前全同态的实现对各方面限制和要求比较高, 理论上可以实现, 然而实际工程上很难实现. 文献 [25] 首次给出了安全多方计算最小值的解决方案, 方案以 1-r 编码、同态及秘密共享为基础计算最小值, 设计的协议的效率随着数据大小的增大而线性降低. 文献 [26] 与文献 [25] 类似, 用保密替换方法提升计算最大值的效率, 并给出了所有数据都属于某个稀疏集合内的最大值问题解决方案. 文献 [27] 提出了一次性计算出一组数据中的最大值和最小值的方案. 上述保密计算最值的方案[25−27]中, 一部分方案[26,27]只能得到最值的明文, 如果用来计算极差、极值和, 就会泄露中间结果; 文献 [25] 的方案可以得到最值的密文, 但它的协议是基于乘法同态密码系统设计的, 得到最值密文不解密就不能计算出极差、极值和的密文. 也就是说, 文献 [25−27] 直接用于计算极差、极值和, 会泄露中间结果. 而我们的方案可以保密计算出极差、极值和, 且不泄露任何中间结果. 文献 [28] 只是将文献 [25−27] 中在半诚实模型下设计的保密计算最值协议转化为恶意模型下的协议. 总之, 针对极差、极值和保密计算问题的研究工作仍有以下挑战: (1) 目前保密计算最值的方法通常具有一定局限性, 大多只能解决所有数据都在一个确定范围内的最值, 不适用于计算数据范围很大的某个稀疏集的最值. 已有方法在这种情况下计算最值的效率很低, 因此更不适用于解决该情况下保密计算极差、极值和问题. (2) 若采用先求最值再求最值的差、和的思路计算极差、极值和, 将会泄露最值. 安全多方计算要求只能泄露极差和极值和, 任何中间结果都不应该被泄露. 本文提出了解决上述两个新的保密计算问题的新方案, 可以在不泄露任何中间结果的情况下保密计算一组分布式数据的极差, 保密计算一组分布式数据的极值和. 本文主要贡献如下.

(1) 提出并研究了保密科学计算中的新问题, 即极值和的保密计算问题, 给出了安全多方计算极差的方案. 拓展和丰富了保密科学计算的研究内容.

(2) 巧用编码方法和转化技巧, 解决多方参与、每人拥有一个私密数据场景下极差、极值和的保密计算问题. 本文首先提出了一种编码方法, 编码方法与 Lifted ElGamal 密码系统相结合, 解决多方参与、每人拥有一个私密数据场景下分布式数据集的极差的保密计算问题、极值和的保密计算问题.

(3) 使用问题转化技巧, 解决了两方参与、每方拥有多个数据场景下极差、极值和的保密计算问题. 将编码方法稍作改动后与 Paillier 密码系统相结合, 解决两方参与、每方拥有多个数据场景下分布式数据集并集的极差的保密计算问题以及相同场景下另一个保密计算问题, 即分布式数据集并集的极值和的保密计算问题. 为解决其他问题提供了新的思路.

(4) 本文协议安全高效、具有广泛实用性. 本文用模拟范例方法证明了协议在半诚实模型下的安全性. 本文协议不仅适用于数据集范围比较小的情况, 同样可以解决数据范围很大情况下分布式稀疏集上的极差、极值和这两种保密计算问题. 模拟实验测试表明本文协议简单高效.

2 基础知识

2.1 半诚实模型及其安全性定义

半诚实参与者按照协议要求执行协议, 但可能保存协议执行过程中的中间数据, 在协议执行结束后利用收集的所有数据推导其他参与者的私密信息. 若模型中所有参与者都是半诚实的, 称其为半诚实模型[29].

设 n 个参与者 $P_i\ (i=1,\ldots,n)$ 的输入数据分别为 x_i, 记 $X=(x_1,\ldots,x_n)$. 参与者共同执行协议 π 计算 n 元函数 $f(X)$. 把执行协议中 P_i 得到的信息序列记作 $view_i^{\pi}(X)=(x_i,r_i,m_i^1,\ldots,m_i^k,f_i(X))$. 其中, r_i 是 P_i 选择的随机数, $m_i^j(j\in[1,k])$ 是 P_i 收到的第 j 个消息, $f_i(X)$ 是 P_i 得到的计算结果. 对于部分参与者集合 $I=\{P_s,\ldots,P_t\}\subset\{P_1,\ldots,P_n\}(1<s<t<n)$, 记 $view_I^{\pi}(X)=(I,view_s^{\pi}(X),\ldots,view_t^{\pi}(X),f_I(X))$.

定义 1. 在参与者都是半诚实参与者的情况下, 若存在概率多项式时间算法 S, 可以使得对于任意 $I=\{P_s,\ldots,P_t\}\subset\{P_1,\ldots,P_n\}(1<s<t<n)$, 均有下式成立:

$$\{S(I,(x_s,\ldots,x_t),f_I(X))\}_X\overset{c}{\equiv}\{view_I^{\pi}(X)\}_X \tag{1}$$

则称 π 保密地计算了函数 $f(X)$, 其中 $\overset{c}{\equiv}$ 表示计算不可区分.

定义 2. 特别地, 当两方参与者 (不妨设为 P_1, P_2), 执行协议 π 保密计算二元函数 $f(x_1,x_2)$ 时, 若存在概率多项式时间算法 S_1, S_2 使得以下两式成立:

$$\{S_1(x_1,f_1(x_1,x_2))\}_{x_1,x_2}\overset{c}{\equiv}\{view_1^{\pi}(x_1,x_2)\}_{x_1,x_2} \tag{2}$$

$$\{S_2(x_2,f_2(x_1,x_2))\}_{x_1,x_2}\overset{c}{\equiv}\{view_2^{\pi}(x_1,x_2)\}_{x_1,x_2} \tag{3}$$

则称 π 保密地计算 $f(x_1,x_2)$.

上述通过构造模拟器证明协议安全性的方法称为模拟范例方法[30]. 本文所有协议的安全性采用该方法证明.

2.2 Lifted ElGamal 门限密码系统

在有 n 个参与者的门限密码体制中, 公钥由 n 个参与者联合生成, 每个参与者都可以使用公钥加密信息; 密钥被分割给多个参与者, 解密必须有 $t\ (1<t<n)$ 个持有部分密钥的参与者共同合作才能完成, 这样的密码系统被称为 (t,n) 门限密码系统. 在安全多方计算中, 为了抵抗尽可能多的参与者的合谋, 通常采用 (n,n) 门限. 本文中协议采用 Lifted ElGamal 系统构造门限密码系统.

Lifted ElGamal 门限密码系统[31]基于 ElGamal 密码系统[32], 将原系统中的 m 用 g^m 进行替换后构造门限. 具体构造过程如下.

● 密钥生成. 依据安全参数 κ, 系统选取一个 κ 比特的大素数 p, 以及 Z_p^* 的生成元 g. 参与者 P_i 随机选取 k_i 作为各自的私钥, 计算并公布 $h_i=g^{k_i}\bmod p$. 所有参与者联合生成公钥:

$$h=\prod_{i=1}^{n}h_i \bmod p=g^{\sum_{i=1}^{n}k_i} \bmod p.$$

● 加密. 加密明文信息$m\in Z_p$, 选取一个随机数$r\in Z_p^*$, 加密得到对应密文:

$$C=(C_1,C_2)=(g^r \bmod p, g^m h^r \bmod p).$$

● 解密. 解密密文信息$C=(C_1,C_2)$, 所有参与者按照下式联合解密, 最终解密结果记为d.

$$d=\frac{C_2}{\prod_{i=1}^{n}C_1^{k_i}} \bmod p.$$

● 加法同态性. 对于$m_1,m_2\in Z_n$, 假设:

$E(m_1)=(C_1,C_2)=(g^{r_1} \bmod p, g^{m_1}h^{r_1} \bmod p), E(m_2)=(C_1',C_2')=(g^{r_2} \bmod p, g^{m_2}h^{r_2} \bmod p)$, Lifted ElGamal 密码系统具有以下性质:

$$E(m_1)E(m_2)=(g^{r_1+r_2} \bmod p, g^{m_1+m_2}h^{r_1+r_2} \bmod p)=E(m_1+m_2 \bmod p),$$
$$E(m_1)^{m_2} \bmod p=(g^{r_1m_2} \bmod p, g^{m_1m_2}h^{r_1m_2} \bmod p)=E(m_1m_2 \bmod p).$$

在 Lifted ElGamal 密码系统中, 加密的信息$m\in\{0,1\}$时, g^m的值很容易确定不需要额外计算, 故加密一个明文$m\in\{0,1\}$需要两次模指数运算; 解密时因为需要联合解密, 故解密一个密文信息需要n次模指数运算.

此外要注意, 对密文C联合解密后获得的是$d=g^m \bmod p$而不是直接获得原明文m. 当解密后d为$1,g,g^2$等相对较小的数时获得原明文m很容易. 但当d很大时, 若想获得m, 需要额外进行对数计算. 为了对数计算方便, 一般取$g=2, g^m<p$. 如果$g\neq 2$, 就需要提前建立一张关系表$Relation=(m,g^m \bmod p)$, 解密后查表$Relation$获得原明文.

注意, 在 3 个及以上参与者的安全协议中, 为了防止合谋攻击通常会采用门限密码系统. 但在两个参与者的安全协议中, 不会发生合谋的情况, 通常不采用门限密码系统. 在两方参与情况下, 本文采用 Paillier 密码系统.

2.3 Paillier 密码系统

Paillier 密码系统简述如下[33].

● 密钥生成. 依据安全参数κ, 选取两个κ比特的大素数p, q, 并令$n=pq$, $\lambda=lcm(p-1,q-1)$. 随机选取$g\in Z_{n^2}^*$, 使其满足$gcd(L(g^\lambda \bmod N^2),n)=1$, 其中$L(x)$被定义为$L(x)=\frac{x-1}{n}$. 系统的公钥$pk=(g,n)$, 私钥$sk=\lambda$.

● 加密. 加密明文信息$m\in Z_n$, 选取一个随机数$r\in Z_n^*$, 加密得到对应密文:

$$E(m)=g^m r^n \bmod n^2.$$

● 解密. 解密密文信息$C\in Z_{n^2}^*$, 解密得到对应明文:

$$D(C)=\frac{L(C^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n.$$

● 加法同态性. 对于$m_1,m_2\in Z_n$, 假设$E(m_1)=g^{m_1}r_1^n \bmod n^2$, $E(m_2)=g^{m_2}r_2^n \bmod n^2$, Paillier 密码系统具有以下性质:

$$E(m_1)E(m_2)=g^{m_1+m_2}(r_1r_2)^n \bmod n^2=E(m_1+m_2 \bmod n).$$

进一步, 明文m_2已知时, 还可以得到下述性质:

$$E(m_1)^{m_2} \bmod n^2=g^{m_1m_2}(r_1^{m_2})^n \bmod n^2=E(m_1m_2 \bmod n).$$

在 Paillier 密码系统中, $g=1+Qn$(Q为正整数)时, 有$E(m)=(1+Qn)^m r^n \bmod n^2=(1+Qnm)r^n \bmod n^2$, 故加密一个明文信息仅需要一次模指数运算. 解密时由于g^λ可以重复利用, 故解密一个密文信息也仅需要一次模指数运算. 为了方便计算, 本文中采用 Paillier 密码系统时都选取$g=1+Qn$.

3 多方参与极差、极值和的保密计算

3.1 问题描述及编码方法

● 问题描述. n个参与者P_i $(i=1,\ldots,n)$的私有数据分别为$x_i\in U=\{u_1,u_2,\ldots,u_m\}$, 已知其中$u_1<u_2<\ldots<u_m$,

且$|U|=m$, $u_1<x_i<u_m$. 不妨记分布式集合$X=\{x_1,\ldots,x_m\}$. 很多时候这个标准集U是客观存在的[27], 如成绩集合(百分制不超过 100 分, 高考单科成绩不超过 150 分), 年龄集合 (目前不超过 150 岁), 某系列产品集合 (系列号都在生产档案中), 候选人集合 (所有候选人都提前登记在册), 人群集合等等, 都有着客观全集范围.

问题 1: 参与者希望计算出$g_1(X)=\max(X)-\min(X)$ ($g_1(X)$可简记为g_1), 且不泄露各自的私密数据.

问题 2: 参与者希望计算出$g_2(X)=\max(X)+\min(X)$ ($g_2(X)$可简记为g_2), 且不泄露各自的私密数据.

注意: 问题 1 与问题 2 是相互独立的两个不同的安全多方计算问题, 为了书写简洁, 此处一并给出描述与相关定义.

● 编码方法 1.

(1) 参与者P_1根据x_1初始化行向量$x'=(x'_1,\ldots,x'_m,x'_{m+1})$和$x''=(x''_1,\ldots,x''_m,x''_{m+1})$:

$$x'_j(x''_j)=\begin{cases}1, & u_j\leqslant x_1\\ 0, & u_j>x_1\end{cases},$$

其中, $j\in[1,m+1]$. 然后将x',x''发送给P_2.

(2) 对于$i\in[2,n]$, 参与者P_i分别依次执行以下操作.

(2.1) P_i根据自己的数据x_i计算并更新向量$x'\leftarrow(x'_1,\ldots,x'_m,x'_{m+1})$, $x''\leftarrow(x''_1,\ldots,x''_m,x''_{m+1})$. 具体计算如下:

$$x'_j=\begin{cases}1, & u_j\leqslant x_i\\ x'_j, & u_j>x_i\end{cases},\ x''_j=\begin{cases}x''_j, & u_j\leqslant x_i\\ 0, & u_j>x_i\end{cases},$$

其中, $j\in[1,m+1]$.

(2.2) P_i将更新后的x'和x''发送给下一位参与者, 直到P_n计算并更新x'和x''后不再向下一位参与者发送数据.

(3) P_n进行以下操作.

(3.1) 根据向量x', 计算得到新向量$y'=(y'_1,\ldots,y'_m)$. 其中, $y'_j=x'_j-x'_{j+1}$ (此处$j\in[1,m]$). 然后计算$x_{\max}=\sum_{j=1}^{m}y'_j\cdot u_j$.

(3.2) 根据向量x'', 计算得到新向量$y''=(y''_1,\ldots,y''_m)$. 其中, $y''_j=x''_j-x''_{j+1}$ (此处$j\in[1,m]$). 然后计算$x_{\min}=\sum_{j=1}^{m}y''_j\cdot u_j$. 容易发现以下结论.

● 事实 1. 分布式集合$X=\{x_1,\ldots,x_m\}$的极差$g_1=x_{\max}-x_{\min}$.

证明: 我们不妨令$\min(x_1,\ldots,x_n)=u_k$, $\max(x_1,\ldots,x_n)=u_l$, 极差$g_1=u_l-u_k$. 根据编码方法 1 可知, 所有参与者操作结束后, 向量x'中最后一个 1 元素所在的位置 (不妨记为第l位) 代表集合X中最大值u_l在全集U中所在的位置. 且在向量x'中, l位及之前位元素全为 1, l位之后的元素全为 0. 对x'向量的分量从左到右逐位做差, 即计算$y'_j=x'_j-x'_{j+1}$ ($j\in[1,m]$) 得到新向量y'. 故向量y'中唯一 1 元素所在的位置即为第l位, 也就是集合X中最大值u_l在全集U中所在的位置, 其余分量全部为 0. 那么$x_{\max}=\sum_{j=1}^{m}y'_j\cdot u_j=u_l$.

类似地, 向量y''中唯一 1 元素所在的位置即为第k位, 也就是集合X中最小值u_k在全集U中所在的位置, 其余分量全部为 0. 故$x_{\min}=\sum_{j=1}^{m}x''_j\cdot u_j=u_k$.

那么分布式集合$X=\{x_1,\ldots,x_m\}$的极差$g_1=x_{\max}-x_{\min}$. 事实 1 即证.

● 事实 2. 分布式集合$X=\{x_1,\ldots,x_m\}$的极值和$g_2=x_{\max}+x_{\min}$.

证明: 基于事实 1 的正确性, 易证分布式集合$X=\{x_1,\ldots,x_m\}$的极值和$g_2=x_{\max}+x_{\min}$.

例 1: 在参与者的数据范围很大的情况下, 用上述方法计算极差. 已知人数$n=4$, $m=8$, 全集$U=\{1,40,400,860,10000,30420,40380,70760\}$. 当参与者各自私有数据分别为$x_1=30420$, $x_2=40$, $x_3=10000$, $x_4=40380$时, 共同执行编码方法 1 计算这组分布式数据的极差的具体过程如下.

(1) P_1根据$x_1=30420$初始化行向量$x'=(1,1,1,1,1,1,0,0,0)$和$x''=(1,1,1,1,1,1,0,0,0)$. 然后将x',x''发送给P_2.

(2) P_2根据自己的数据$x_2=40$, 进行以下操作.

(2.1) 计算并更新向量x': 小于等于x_2的分量改为 1, 大于x_2的分量保持不变, 故得到$x'\leftarrow(1,1,1,1,1,1,0,0,0)$.

(2.2) 计算并更新向量x'': 大于x_2的分量改为 0, 小于等于x_2的分量保持不变, 故得到$x''\leftarrow(1,1,0,0,0,0,0,0,0)$.

然后将x',x''发送给下一位参与者.

(3) P_3, P_4根据自己的私有数据x_3, x_4执行步骤 (2) 中类似操作, 最后P_4可以得到$x' \leftarrow (1,1,1,1,1,1,1,0,0)$, $x'' \leftarrow (1,1,0,0,0,0,0,0,0)$.

下面用表 1 和表 2 说明以上各步骤中的数据变化情况.

表 1 编码方法 1 更新x'过程

U	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	
初始化x'	1	1	1	1	1	1	0	0	0
P_2更新x'	1	1	1	1	1	1	0	0	0
P_3更新x'	1	1	1	1	1	1	0	0	0
P_4更新x'	1	1	1	1	1	1	1	0	0

表 2 编码方法 1 更新x''过程

U	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	
初始化x''	1	1	1	1	1	1	0	0	0
P_2更新x''	1	1	0	0	0	0	0	0	0
P_3更新x''	1	1	0	0	0	0	0	0	0
P_4更新x''	1	1	0	0	0	0	0	0	0

(4) P_4进行以下操作.

(4.1) 按照编码方法 1 步骤 (3.1) 中所述, 将向量x'的各维数据从左到右 (下标从小到大) 的方向依次逐位相减, 得到$y' = (0,0,0,0,0,0,1,0)$. 然后计算$x_{\max} = \sum_{j=1}^{8} y'_j \cdot u_j = 1 \cdot u_7 = 40380$.

(4.2) 按照编码方法 1 步骤 (3.2) 中所述, 将向量x''的各维数据从左到右 (下标从小到大) 的方向依次逐位相减, 得到$y'' = (0,1,0,0,0,0,0,0)$. 然后计算$x_{\min} = \sum_{j=1}^{8} y''_j \cdot u_j = 1 \cdot u_2 = 40$.

(4.3) 这组数据的极差$g_1 = x_{\max} - x_{\min} = 40380 - 40 = 40340$.

例 2: 参与者的数据范围很大的情况下计算极值和. 求例 1 中数据的极值和只需把例 1 最后一步 (4.3) 中操作修改为如下 (4.3):

(4.3) 这组数据的极值和$g_2 = x_{\max} + x_{\min} = 40380 + 40 = 40420$.

3.2 多方参与极差 (极值和) 的保密计算协议

协议 1. 多方参与极差 (极值和) 的保密计算协议.

输入: $P_1,\ldots,P_n$分别输入$x_1,\ldots,x_n$;

输出: $g_1(X)(g_2(X))$.

准备: n个参与者运行 Lifted ElGamal 门限密码系统, 保密各自的私钥并公布联合生成的公钥.

(1) 参与者P_1根据x_1初始化行向量$x' = (x'_1,\ldots,x'_{m+1})$和$x'' = (x''_1,\ldots,x''_{m+1})$.

$$x'_j(x''_j) = \begin{cases} 1, & u_j \leqslant x_1 \\ 0, & u_j > x_1 \end{cases},$$

其中, $j \in [1,m+1]$. 然后对向量x',x''进行加密得到C', C''.

$$C' = (c'_1,\ldots,c'_{m+1}) = E(x') = (E(x'_1),\ldots,E(x'_{m+1})), C'' = (c''_1,\ldots,c''_{m+1}) = E(x'') = (E(x''_1),\ldots,E(x''_{m+1})),$$

并将C', C''发送给P_2.

(2) 对于$i \in [2,n]$, 参与者P_i分别依次执行以下操作:

(2.1) 根据自己的私有数据x_i计算并更新$C' \leftarrow (c'_1,\ldots,c'_{m+1})$, $C'' \leftarrow (c''_1,\ldots,c''_{m+1})$. 具体计算如下:

$$c'_j \leftarrow \begin{cases} E(1), & u_j \leqslant x_i \\ c'_j \cdot E(0), & u_j > x_i \end{cases}, \quad c''_j \leftarrow \begin{cases} c''_j \cdot E(0), & u_j \leqslant x_i \\ E(0), & u_j > x_i \end{cases},$$

其中, $j \in [1,m+1]$.

(2.2) 将更新后的C',C''发送给P_{i+1}, 直到P_n计算及更新操作完成后停止发送数据.

(3) P_n进行以下操作:

(3.1) 根据C', 计算$Y' = (Y'_1,\ldots,Y'_m)$. 其中, $Y'_j = c'_j \cdot c'^{-1}_{j+1}$(此处$j \in [1,m]$). 然后计算$X_{\max} = \prod_{j=1}^{m} Y'^{u_j}_j$.

(3.2) 根据C'', 计算$Y'' = (Y''_1,\ldots,Y''_m)$. 其中, $Y''_j = c''_j \cdot c''^{-1}_{j+1}$(此处$j \in [1,m]$). 然后计算$X_{\min} = \prod_{j=1}^{m} Y''^{u_j}_j$.

(3.3) 计算 $G_1 = X_{\max} \cdot X_{\min}^{-1}$ (计算 $G_2 = X_{\max} \cdot X_{\min}$).

(4) 所有参与者对 G_1 (G_2) 进行联合解密, 根据事实 1 (事实 2) 得到分布式集合 X 的极差 (极值和).

3.3 协议 1 的正确性

依据 Lifted ElGamal 密码系统的加法同态性, 我们容易证明: $X_{\max} = \prod_{j=1}^{m} Y_j'^{u_j} = E(\sum_{j=1}^{m} y_j' \cdot u_j)$, $X_{\min} = \prod_{j=1}^{m} Y_j''^{u_j} = E(\sum_{j=1}^{m} y_j'' \cdot u_j)$, $G_1 = X_{\max} \cdot X_{\min}^{-1} = E(x_{\max} - x_{\min})$, $G_2 = X_{\max} \cdot X_{\min} = E(x_{\max} + x_{\min})$. 再基于事实 1 和事实 2 的正确性可知协议 1 正确.

3.4 协议 1 的安全性

定理 1. 协议 1 在半诚实模型下是安全的.

证明: 应用模拟范例证明定理 1. 协议 1 采用了 Lifted ElGamal 密码系统, 由于 Lifted ElGamal 密码系统中, 解密需要所有参与者共同参与才能完成, 再由 Lifted ElGamal 密码系统的语义安全性可知, 在安全性方面 n 个参与者的地位是完全平等的. 以 P_1 为例, 由于在协议 1 中 P_1 受到的最大攻击是其他全部参与者 $I = \{P_2, \ldots, P_n\}$ 合谋以获得 x_1, 若 P_1 的信息对最大攻击合集 I 是安全的, 那么对最大攻击集合的任意子集也是安全的. 若能证明协议对 P_1 的信息是安全的, 那么就能说明协议对所有参与者的信息都是安全的, 故仅证明协议 1 中 P_1 的所有私密信息对合谋者 $I = \{P_2, \ldots, P_n\}$ 是安全的即可.

首先, 在协议 1 执行中:

$$view_I^{\pi}(x_1, \ldots, x_n) = \{(x_2, \ldots, x_n), (R_2, \ldots, R_n), C, G_1^{(1)}(G_2^{(1)}), f_I(x_1, \ldots, x_n)\},$$

其中, $R_2, \ldots, R_n$ 分别为协议执行过程中参与者 $P_2, \ldots, P_n$ 用到的随机数集合; $G_1^{(1)}$ 为 P_1 对密文 G_1 的部分解密结果, $G_2^{(1)}$ 为 P_1 对密文 G_2 的部分解密结果. $f_I(x_1, \ldots, x_n)$ 表示合谋者 $I = \{P_2, \ldots, P_n\}$ 联合执行 n 元函数 $f(x_1, \ldots, x_n)$ 得到的结果.

然后构造相应的模拟器 S, S 按照如下方式运行.

(1) S 接收到 $(I, (x_2, \ldots, x_n), f_I(x_1, \ldots, x_n))$ 输入后, 选取任意一个使得 $f_I(x_1^*, x_2, \ldots, x_n) = f_I(x_1, \ldots, x_n)$ 成立的 x_1^*.

(2) S 根据 x_1^* 初始化行向量 $x'^* = (x_1'^*, \ldots, x_m'^*, x_{m+1}'^*)$ 和 $x''^* = (x_1''^*, \ldots, x_m''^*, x_{m+1}''^*)$:

$$x'^*(x''^*) = \begin{cases} 1, & u_j \leqslant x_1^* \\ 0, & u_j > x_1^* \end{cases},$$

其中, $j \in [1, m+1]$. 然后对向量 x'^*, x''^* 进行加密得到 C'^*, C''^*.

$$C'^* = (c_1'^*, \ldots, c_m'^*, c_{m+1}'^*) = E(x'^*) = (E(x_1'^*), \ldots, E(x_m'^*), E(x_{m+1}'^*)),$$

$$C''^* = (c_1''^*, \ldots, c_m''^*, c_{m+1}''^*) = E(x''^*) = (E(x_1''^*), \ldots, E(x_m''^*), E(x_{m+1}''^*)).$$

(3) 对于 $i \in [2, n]$, S 依次根据数据 x_i 计算并更新 $C'^* \leftarrow (c_1'^*, \ldots, c_m'^*, c_{m+1}'^*)$, $C_1''^* \leftarrow (c_1''^*, \ldots, c_m''^*, c_{m+1}''^*)$. 具体如下:

$$c_j'^* \leftarrow \begin{cases} E(1), & u_j \leqslant x_i \\ c_j'^* \cdot E(0), & u_j > x_i \end{cases}, \quad c_j''^* \leftarrow \begin{cases} c_j''^* \cdot E(0), & u_j \leqslant x_i \\ E(0), & u_j > x_i \end{cases},$$

其中, $j \in [1, m+1]$.

(4) S 进行以下操作.

(4.1) 根据 C'^*, 计算 $Y'^* = (Y_1'^*, \ldots, Y_m'^*)$. 其中, $Y_j'^* = c_j'^* \cdot c_{j+1}'^{*-1}$ (此处 $j \in [1, m]$). 再计算 $X_{\max}^* = \prod_{j=1}^{m} Y_j'^{*u_j}$.

(4.2) 根据 C''^*, 计算 $Y''^* = (Y_1''^*, \ldots, Y_m''^*)$, 有 $Y_j''^* = c_j''^* \cdot c_{j+1}''^{*-1}$ (此处 $j \in [1, m]$). 再计算 $X_{\min}^* = \prod_{j=1}^{m} Y_j''^{*u_j}$.

(4.3) 计算 $G_1^* = X_{\max}^* \cdot X_{\min}^{*-1}$ (计算 $G_2 = X_{\max}^* \cdot X_{\min}^*$).

(5) S 对 G_1^* (G_2^*) 进行解密, 根据事实 1 (事实 2) 得到分布式集合 $\{x_1^*, x_2, \ldots, x_n\}$ 的极差 (极值和).

在模拟器 S 执行中, 令:

$$S(I, (x_2, \ldots, x_n), f_I(x_1, \ldots, x_n)) = \{(x_2, \ldots, x_n), (R_2^*, \ldots, R_n^*), C^*, G_1^{*(1)}(G_2^{*(1)}), f_I(x_1^*, x_2, \ldots, x_n)\}.$$

由于随机数 $R_2, \ldots, R_n$ 与 $R_2^*, \ldots, R_n^*$ 不能区分; I 不能解密密文, 根据 Lifted ElGamal 门限密码系统的语义安全性, 有 $C^* \overset{c}{\equiv} C$, $G_1^{*(1)} \overset{c}{\equiv} G_1^{(1)}$, $G_2^{*(1)} \overset{c}{\equiv} G_2^{(1)}$; 又由于 $f_I(x_1^*, x_2, \ldots, x_n) = f_I(x_1, \ldots, x_n)$, 因此:

$$\{view_I^{\pi}(x_1,\ldots,x_n)\} \overset{c}{\equiv} \{S(I,(x_2,\ldots,x_n),f_I(x_1,\ldots,x_n))\}.$$

4 两方参与极差、极值和的保密计算

4.1 问题描述及编码方法

● 问题描述. Alice 拥有私密集合 $A=\{a_1,\ldots,a_q\}$, Bob 拥有私密集合 $B=\{b_1,\ldots,b_w\}$. $A,B\subseteq U=\{u_1,u_2,\ldots,u_m\}$, 其中 $u_1<u_2<\ldots<u_m$ 且 $|U|=m$.

问题 1: 参与者双方希望计算出 $g_1(A\cup B)=\max(A\cup B)-\min(A\cup B)$, 且不泄露各自的私密数据.

问题 2: 参与者双方希望计算出 $g_2(A\cup B)=\max(A\cup B)+\min(A\cup B)$, 且不泄露各自的私密数据.

需要注意, 问题 1 与问题 2 是相互独立的两个不同的安全多方计算问题, 为了书写简洁, 此处一并给出相关定义.

● 编码方法 2. 不妨记 $a_{\max}=\max(A)$, $a_{\min}=\min(A)$, $b_{\max}=\max(B)$, $b_{\min}=\min(B)$, $x_{\max}=\max(A\cup B)$, $x_{\min}=\min(A\cup B)$. 以编码方法 1 为基础进行改动, 得到编码方法 2. 具体如下:

(1) Alice 根据 $a_{\max}$ 和 $a_{\min}$ 初始化行向量 $x'=(x'_1,\ldots,x'_m)$ 和 $x''=(x''_1,\ldots,x''_m)$. 其中:

$$x'_j(x''_j)=\begin{cases}1, & u_j\leqslant a_{\max}(a_{\min})\\ 0, & u_j>a_{\max}(a_{\min})\end{cases},$$

然后将 x' 和 x'' 发送给 Bob.

(2) Bob 根据自己的数据 $b_{\max},b_{\min}$ 进行以下操作.

(2.1) 根据 $b_{\max}$ 在全集 U 中的位序 (位置) $index_1(1\leqslant index_1\leqslant m)$, 选择向量 x' 中第 $index_1$ 维分量 x'_{index_1}, 并记为 $\alpha=x'_{index_1}$; 计算 $\overline{\alpha}=1-\alpha$.

(2.2) 根据 $b_{\min}$ 在全集 U 中的位序 (位置) $index_2(1\leqslant index_2\leqslant m)$, 选择向量 x'' 中第 $index_2$ 维分量 x''_{index_2}, 并记为 $\beta=x''_{index_2}$; 计算 $\overline{\beta}=1-\beta$.

(2.3) 将 $(\alpha,\overline{\alpha})$, $(\beta,\overline{\beta})$ 分别随机置换后发送给 Alice.

(3) Alice 进行以下操作.

(3.1) 计算 $d_1=\alpha\cdot a_{\max}$, $d'_1=\overline{\alpha}\cdot a_{\max}$; 计算 $d_2=\beta\cdot a_{\min}$, $d'_2=\overline{\beta}\cdot a_{\min}$.

(3.2) 将 (d_1,d'_1), (d_2,d'_2) 发送给 Bob.

(4) Bob 进行以下操作.

(4.1) 计算 $x_{\max}=\alpha\cdot a_{\max}+(1-\alpha)\cdot b_{\max}=d_1+(1-\alpha)\cdot b_{\max}$.

(4.2) 计算 $x_{\min}=(1-\beta)\cdot a_{\min}+\beta\cdot b_{\min}=d'_2+\beta\cdot b_{\min}$. 容易发现以下结论.

● 事实 3. 分布式集合 $A\cup B$ 的极差 $g_1=(\alpha\cdot a_{\max}+(1-\alpha)\cdot b_{\max})-((1-\beta)\cdot a_{\min}+\beta\cdot b_{\min})$.

证明: 首先, $g_1(A\cup B)=\max(A\cup B)-\min(A\cup B)=\max(a_{\max},b_{\max})-\min(a_{\min},b_{\min})=x_{\max}-x_{\min}$. 使得原问题得到转换.

其次, 由编码方法 2 可知, 当 $a_{\max}\geqslant b_{\max}$ 时 $\alpha=1$, 此时 $x_{\max}=\alpha\cdot a_{\max}+(1-\alpha)\cdot b_{\max}=a_{\max}$; 当 $a_{\max}<b_{\max}$ 时 $\alpha=0$, 此时 $x_{\max}=\alpha\cdot a_{\max}+(1-\alpha)\cdot b_{\max}=b_{\max}$.

另一方面, 当 $a_{\min}\geqslant b_{\min}$ 时 $\beta=1$, 此时 $x_{\min}=(1-\beta)\cdot a_{\min}+\beta\cdot b_{\min}=b_{\min}$; 当 $a_{\min}<b_{\min}$ 时 $\beta=0$, 此时 $x_{\min}=(1-\beta)\cdot a_{\min}+\beta\cdot b_{\min}=a_{\min}$.

进一步可知 $g_1=x_{\max}-x_{\min}=(\alpha\cdot a_{\max}+(1-\alpha)\cdot b_{\max})-((1-\beta)\cdot a_{\min}+\beta\cdot b_{\min})$. 事实 3 即证.

● 事实 4. 分布式集合 $A\cup B$ 的极值和 $g_2=(\alpha\cdot a_{\max}+(1-\alpha)\cdot b_{\max})+((1-\beta)\cdot a_{\min}+\beta\cdot b_{\min})$.

证明: 基于事实 3 的正确性, 进一步可知 $g_2=x_{\max}+x_{\min}=(\alpha\cdot a_{\max}+(1-\alpha)\cdot b_{\max})+((1-\beta)\cdot a_{\min}+\beta\cdot b_{\min})$. 事实 4 即证.

例 3: 已知 $m=8$, 全集 $U=\{10,20,30,869,1000,6990,7000,7010\}$. 当 Alice 的集合 $A=\{30,869,1000,7000\}$, Bob 的集合 $B=\{20,30,869,6990\}$, 共同执行编码方法 2 计算分布式集合 $A\cup B$ 的极差的具体过程如下.

(1) Alice 根据自己的数据 $a_{\max}=7000$, $a_{\min}=30$ 初始化行向量 $x'=(1,1,1,1,1,1,1,0)$, $x''=(1,1,1,0,0,0,0,0)$.

然后将 x',x'' 发送给 Bob.

(2) Bob 根据自己的数据 $b_{\max}=6990$, $b_{\min}=20$, 进行以下操作.

(2.1) 根据 $b_{\max}=6990$ 在全集 U 中的位序 6, 选择 x' 中第 6 维分量, 并记为 $\alpha=1$; 计算 $\overline{\alpha}=1-\alpha=0$.

(2.2) 根据 $b_{\min}=20$ 在全集 U 中的位序 2, 选择 x'' 中第 2 维分量, 并记为 $\beta=1$; 计算 $\overline{\beta}=1-\beta=0$.

(2.3) 将 $(\alpha,\overline{\alpha})$, $(\beta,\overline{\beta})$ 分别随机置换后发送给 Alice.

(3) Alice 进行以下操作.

(3.1) 计算 $d_1=\alpha\cdot a_{\max}=7000$, $d_1'=\overline{\alpha}\cdot a_{\max}=0$; 计算 $d_2=\beta\cdot a_{\min}=30$, $d_2'=\overline{\beta}\cdot a_{\min}=0$.

(3.2) 将 $(d_1=7000,d_1'=0)$, $(d_2=30,d_2'=0)$ 发送给 Bob.

(4) Bob 进行以下操作.

(4.1) 计算 $x_{\max}=d_1+(1-\alpha)\cdot b_{\max}=7000+(1-1)\cdot 6990=7000$; 计算 $x_{\min}=d_2'+\beta\cdot b_{\min}=0+1\cdot 20=20$.

(4.2) 计算 $g_1=x_{\max}-x_{\min}=7000-20=6980$ 即为分布式集合 $A\cup B$ 的极差.

例 4: 求例 3 中数据的极值和. 只需把例 3 最后 (4.2) 步中操作修改为如下 (4.2):

(4.2) 这组数据的极值和 $g_2=x_{\max}+x_{\min}=7000+20=7020$.

4.2 问两方参与极差 (极值和) 的保密计算协议

协议 2. 两方参与极差 (极值和) 的保密计算协议.

输入: Alice 输入集合 $A=\{a_1,\ldots,a_q\}$, Bob 输入集合 $B=\{b_1,\ldots,b_w\}$;

输出: $g_1(A\cup B)(g_2(A\cup B))$.

准备: Alice 运行 Paillier 密码系统, 保密私钥 sk 并公布公钥 pk.

(1) Alice 进行以下操作.

(1.1) 根据 $a_{\max}$ 和 $a_{\min}$ 初始化行向量 $x'=(x_1',\ldots,x_m')$ 和 $x'=(x_1',\ldots,x_m')$. 其中,

$$x_j'(x_j'')=\begin{cases}1, & u_j\leqslant a_{\max}(a_{\min})\\ 0, & u_j>a_{\max}(a_{\min})\end{cases}.$$

(1.2) 然后对向量 x',x'' 进行加密得到 C', C'':

$$C'=(c_1',\ldots,c_m')=E(x')=(E(x_1'),\ldots,E(x_m')),C''=(c_1'',\ldots,c_m'')=E(x'')=(E(x_1''),\ldots,E(x_m'')),$$

并将 C',C'' 发送给 Bob.

(2) Bob 进行以下操作.

(2.1) 根据 $b_{\max}$ 在全集 U 中的位序 (位置) $index_1$ $(1\leqslant index_1\leqslant m)$, 选择向量 C' 中第 $index_1$ 维分量 c'_{index_1}, 并记 $E_\alpha=c'_{index_1}\cdot E(0)$; 计算 $E_{\overline{\alpha}}=E(1)\cdot E_\alpha{}^{-1}$.

(2.2) 根据 $b_{\min}$ 在全集 U 中的位序 (位置) $index_2$ $(1\leqslant index_2\leqslant m)$, 选择向量 C'' 中第 $index_2$ 维分量 c''_{index_2}, 并记 $E_\beta=c''_{index_2}\cdot E(0)$; 计算 $E_{\overline{\beta}}=E(1)\cdot E_\beta{}^{-1}$;

(2.3) 将 $(E_\alpha,E_{\overline{\alpha}})$, $(E_\beta,E_{\overline{\beta}})$ 分别随机置换后发送给 Alice.

(3) Alice 计算 $D_1=E_\alpha^{a_{\max}}$, $D_1'=E_{\overline{\alpha}}^{a_{\max}}$; 计算 $D_2=E_\beta^{a_{\min}}$, $D_2'=E_{\overline{\beta}}^{a_{\min}}$; 最后将 (D_1,D_1'), (D_2,D_2') 发送给 Bob.

(4) Bob 进行以下操作.

(4.1) 计算 $X_{\max}=D_1\cdot(E(1)\cdot E_\alpha^{-1})^{b_{\max}}$; 计算 $X_{\min}=D_2'\cdot E_\beta^{b_{\min}}$.

(4.2) 计算 $G_1=X_{\max}\cdot X_{\min}^{-1}(G_2=X_{\max}\cdot X_{\min})$.

(4.3) 将 $G_1(G_2)$ 发送给 Alice.

(5) Alice 对 $G_1(G_2)$ 解密得到分布式集合 $A\cup B$ 的极差 $g_1(A\cup B)$ (极值和 $g_2(A\cup B)$).

4.3 协议 2 的正确性

首先, 协议 2 本质上是把原问题规约为: Alice 拥有两个元素 $a_{\max},a_{\min}$, Bob 拥有两个元素 $b_{\max},b_{\min}$, 求这 4 个

元素组成的分布式集合 D 的极差. 由集合运算性质易知, 若令集合 $D=\{d_1,d_2,d_3,d_4\}=\{a_{\max},a_{\min},b_{\max},b_{\min}\}$, 那么 $\max(A\cup B)-\min(A\cup B)=\max D-\min D$. 故该归约正确.

其次, 由 Paillier 加法同态性, 故 $D_1=E_\alpha^{a_{\max}}=E(\alpha\cdot a_{\max})$, $X_{\max}=D_1\cdot(E(1)\cdot E_\alpha^{-1})^{b_{\max}}=E(\alpha\cdot a_{\max}+(1-\alpha)\cdot b_{\max})$. 再由于 $D_2'=E_{\overline{\alpha}}^{a_{\min}}=E((1-\beta)\cdot a_{\min})$, 那么 $X_{\min}=D_2'\cdot E_\beta^{b_{\min}}=E((1-\beta)\cdot a_{\min}+\beta\cdot b_{\min})$. 故我们可以得到 $G_1=X_{\max}\cdot X_{\min}^{-1}=E(x_{\max}-x_{\min})$, $G_2=X_{\max}\cdot X_{\min}=E(x_{\max}+x_{\min})$. 基于事实 3, 事实 4 的正确性, 可知协议 2 正确.

4.4 协议 2 的安全性

协议 2 第 2 步中若 Bob 直接将 E_α,E_β 发送给 Alice, 由于 Alice 拥有解密密钥, E_α 的解密结果会泄露 $a_{\max},b_{\max}$ 的大小关系, E_β 的解密结果会泄露 $a_{\min},b_{\min}$ 的大小关系. 若将 $(E_\alpha,E_{\overline{\alpha}})$, $(E_\beta,E_{\overline{\beta}})$ 分别随机置换后发送给 Alice, 由于 $\alpha,\overline{\alpha},\beta,\overline{\beta}\in\{0,1\}$, 所以即使 Alice 进行恶意攻击解密 Bob 发送的密文, 也不会泄露 $a_{\max},b_{\max}$ 及 $a_{\min},b_{\min}$ 的大小关系. 下面用模拟范例法具体证明协议 2 的安全性.

定理 5. 协议 2 在半诚实模型下是安全的.

证明: 用类似于协议 1 中方法可证, 此处省略.

5 复杂性分析和实验测试

5.1 复杂性分析

解决一个问题的难易程度通常用解决该问题的算法的计算复杂性衡量. 本文中, 计算复杂性以开销最大的模指数运算次数为衡量指标, 通信复杂性以协议参与者之间需要进行信息交互的次数为衡量指标.

● 计算复杂性. 协议 1 主要应用了 Lifted ElGamal 密码系统. 协议 1 中, 生成公钥需要进行 n 次模指数运算. 数据 $x_{ij}\in\{0,1\}$, 故 P_1 加密数据需要 $4(m+1)$ 次模指数运算. 计算过程中, P_2 至 P_{n-1} 分别需要 $4(m+1)$ 次模指数运算; P_n 计算极差时需要 $8m+5$ 次模指数运算, 计算极值和时需要 $8m+4$ 次模指数运算. 联合解密数据时需要 n 次模指数运算. 故协议 1 计算极差时总共需要 $4nm+6n+4m+1$ 次模指数运算, 计算极值和时总共需要 $4nm+6n+4m$ 次模指数运算.

协议 2 主要应用了 Paillier 密码系统. 协议 2 中, 加密数据时 Alice 进行了 $2m$ 次模指数运算. 计算过程中, Alice 有 4 次模指数运算; Bob 计算极差时有 11 次模指数运算, 计算极值和时有 10 次模指数运算. 解密数据时 Alice 有 2 次模指数运算. 故协议 2 计算极差时总共需要 $2m+17$ 次模指数运算, 计算极值和时总共需要 $2m+16$ 次模指数运算.

● 通信复杂性. 协议 1 中, 生成密钥需要 $n-1$ 次通信, 计算过程需要 $n-1$ 次通信, 联合解密需要 $n-1$ 次通信. 故协议 1 执行中共需要 $3(n-1)$ 次通信.

协议 2 中, Alice 向 Bob 总共发送了 2 次信息; Bob 向 Alice 总共发送了 2 次信息. 故协议 2 执行中共需要 4 次通信.

计算复杂性与通信复杂性具体分析结果如表 3 所示.

表 3 协议计算复杂性与通信复杂性

复杂性	协议1 (计算极差)	协议1 (计算极值和)	协议2 (计算极差)	协议2 (计算极值和)
计算复杂性	$4nm+6n+4m+1$	$4nm+6n+4m$	$2m+17$	$2m+16$
通信复杂性	$3(n-1)$	$3(n-1)$	4	4

5.2 实验测试

● 实验环境. Windows 10 64 位操作系统, 处理器参数为 Intel(R) Core(TM) i5-9400 CPU@ 2.90 GHz, 16.0 GB 内存. 应用 Python 3.9.4 语言在 PyCharm 环境上开发并运行实现.

● 实验方法. 由于协议 1 是多方参与协议, 执行时间与参与者人数 n 和参与者私密数据所属的全集的势 m 有

关. 针对协议 1, 设计两个实验, 分别考察:

(1) m 固定 ($m=50$) 情况下, 协议 1 执行时间随 n 的变化规律. 实际实验中, 全集的势 (大小) $m=50$ 时, 参与者人数分别取 $n=5,10,\ldots,50$. 我们设定 Lifted ElGamal 密码系统的安全参数取定为 1 024, 实验记录 100 次模拟实验所需要的平均执行时间. 实验结果如图 1 所示.

(2) n 固定 ($n=20$) 情况下, 协议 1 执行时间随 m 的变化规律. 实际实验中, 参与者人数 $n=20$ 时, 全集的势分别取 $m=10,20,\ldots,100$. 我们设定 Lifted ElGamal 密码系统的安全参数取定为 1 024, 实验记录 100 次模拟实验所需要的平均执行时间. 实验结果如图 2 所示.

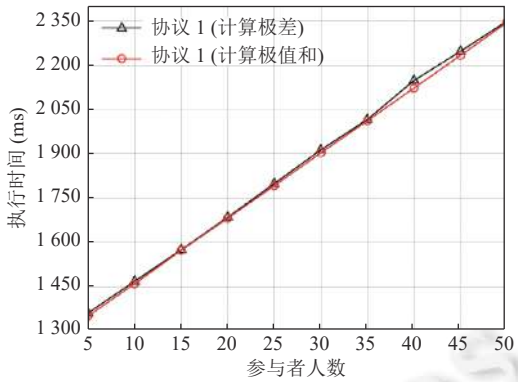

图 1 m 固定时执行时间随 n 的变化规律

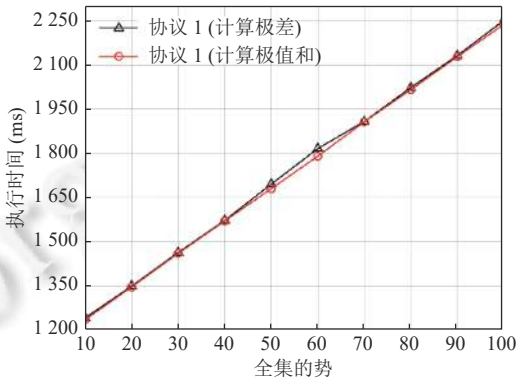

图 2 n 固定时执行时间随 m 的变化规律

协议 2 执行时间仅与 m 有关. 针对协议 2, 设计实验考察.

(3) 协议 2 执行时间随 m 的变化规律. 实际实验中, 数据范围分别取 $m=10,20,\ldots,100$. 我们设定 Paillier 密码系统的安全参数取定为 1 024 比特, 实验记录 100 次模拟实验所需要的平均执行时间. 实验结果如图 3 所示.

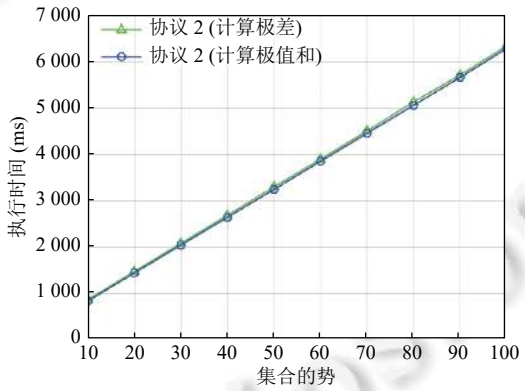

图 3 执行时间随 m 的变化规律

由实验结果可知, 当 m (或 n) 固定时, 协议 1 (协议 2) 的执行时间大致随 n (或 m) 的增加而线性增长.

5.3 方案扩展

从上述协议复杂性分析及实验结果中可以看出, 协议能很好地解决成绩集合、年龄集合等场景下的极差极值和问题. 而在全集 U 的数据范围比较大、个数比较多场景下, 本文协议效率如何? 下面我们讨论该场景下协议 1 的效率和影响效率的因素, 给出一些能提高该场景下协议效率的参考方案.

● 讨论 1. 集合 U 的最大值 $\max(U)=u_m$ 较大时, Lifted ElGamal 密码系统的安全参数 κ 如何选取?

以计算极差为例 (下同), 由于 p 为 κ 比特的数, 则 p 的取值范围为 $2^{\kappa-1}<p<2^{\kappa}$. 我们规定了 $g^M=2^M<p$ (如预

备知识第 2.3 节所述), 那么 $2^M \leqslant 2^{\kappa-1}$, 可知明文 $M \leqslant \kappa-1$. 故选取安全参数时要保证 $\kappa > \max(U)$ (计算极值和时用类似思路可以得到讨论结果 $\kappa > 2\max(U)$).

为了测试 u_m 比较大情况下协议 1 的执行效率, 我们设计实验考察协议 1 执行时间随集合 U 最大值 u_m 的变化规律. 实际实验中, 参与者人数 $n=10$, 集合的势 $m=100$ 时, 集合 U 最大值、对应的 Lifted ElGamal 密码系统的安全参数 κ 分别取 $(u_m,\kappa)=(1000,1024),(2000,2048),\ldots,(8000,8192)$, 将随机数 r 取 256 位. 实验记录 100 次模拟实验所需要的平均执行时间. 实验结果如图 4 中协议 1 ($|r|=256$) 所示.

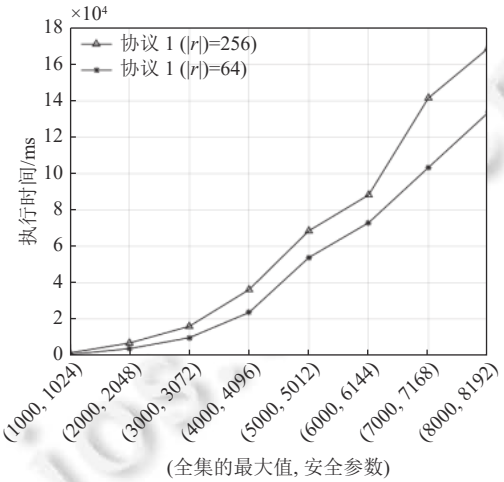

图 4 u_m 较大时执行时间随 (u_m,κ) 的变化规律

怎样更高效率地解决集合数据范围 (即 u_m) 比较大情况下的极差 (极值和) 问题? 我们给出以下 2 个可行解决方案.

● 解决方案 1. 对 Lifted ElGamal 系统中参数 r 进行适当调整, 以提高协议 1 效率. 具体如下.

由于 Lifted ElGamal 密码系统中, 加密明文信息 $M \in Z_p$, 选取一个随机数 $r \in Z_p^*$, 得到对应密文 $C=(C_1,C_2)=(g^r \bmod p, g^M h^r \bmod p)$. 在满足 $r \in Z_p^*$ 的情况下适当减少随机数 r 的位数, 不会影响系统安全性且会减少计算时间. 如上述实验中, 我们选取的 r 的位数为 256, 现我们将 r 的位数改为 64, 其他条件不变情况下再次进行上述实验. 具体实验及比较结果如图 4 所示.

由实验结果可知, 适当降低 Lifted ElGamal 密码系统中随机数 r 的位数, 可以显著降低安全参数 κ 位数过大 (集合 U 数据范围大) 对协议效率带来的影响. 该方案不影响 Lifted ElGamal 系统安全性且可以提高数据范围比较大情况下协议 1 执行效率.

● 解决方案 2. 在全集 U 中数据范围比较大情况下, 对数据进行预处理, 以提高协议 1 效率. 具体方案如下.

在协议 1 输入数据前, 每个参与者 P_i 分别对 x_i 进行预处理操作 $x_i \leftarrow \left\lfloor \frac{x_i}{10} \right\rfloor$ 后再输入数据; 在协议 1 最后加入恢复数据操作, 即真实极差 $g_1 \leftarrow g_1 \cdot 10$. 该方案减小了输入数据的范围, 提高了数据范围比较大情况下协议 1 的执行效率. 另一方面, 该方案存在误差, 但该误差在可接收范围内, 一般小于 0.9% (真实极差 $g_1 > 100$ 情况下).

如本文例 1 中, 参与者数据中的最大值 $x_4=4038$, 最小值 $x_2=4$, 真实极差为 $g_1=x_4-x_2=4034$, 经过预处理后 $x_4 \leftarrow \left\lfloor \frac{x_4}{10} \right\rfloor = 403$, $x_2 \leftarrow \left\lfloor \frac{x_2}{10} \right\rfloor = 0$, 计算极差为 $g_1 \leftarrow (403-0)\cdot 10=4030$, 误差约为 -0.1%.

● 讨论 2. 已知集合 U 的势 $|U|=m$ 比较大情况下, Lifted ElGamal 密码系统的安全参数 κ 如何选取?

由于全集 U 中的元素满足 $u_1 < u_2 < \ldots < u_m$, 全集 U 的势 $|U|=m$ 过大会导致 u_m (即 $\max(U)$) 过大, u_m 过大会影响到安全参数 κ 的选择 (u_m 对 κ 具体影响见讨论 1). 为了测试 m 比较大情况下协议 1 的执行效率, 我们设计实验考察 m 比较大情况下, 协议 1 执行时间随集合 U 的势 m 的变化规律. 实际实验中, 参与者人数 $n=10$ 时, 我们将集

合 U 的势 (数据集中数据个数)、对应的 Lifted ElGamal 密码系统的安全参数 κ 分别取为 $(m,\kappa)=(1000,1024),(2000,2048),\ldots,(8000,8192)$, 规定最大值 $u_m=m$, 将随机数 r 取 256 位. 实验记录 100 次模拟实验所需要的平均执行时间. 实验结果如图 5 中协议 1 ($|r|=256$) 所示.

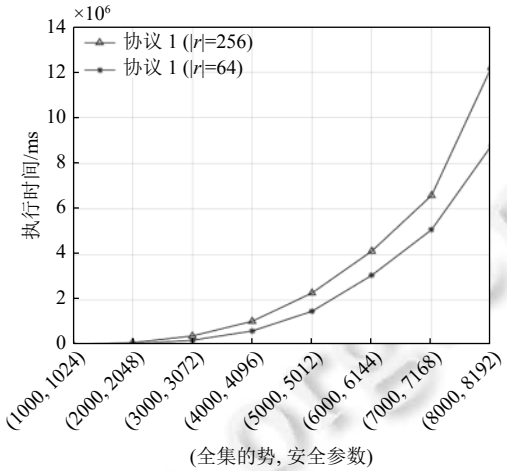

图 5 m 较大时执行时间随 (m,κ) 的变化规律

怎样更高效率地解决集合的势 m 比较大情况下的极差 (极值和) 问题?

● 解决方案. 集合的势 m 比较大情况下也可以用降低随机数 r 的位数的方法降低模指数运算时间, 一定程度上解决 m 较大带来的影响. 如上述实验中, r 的位数为 256, 现将 r 的位数改为 64, 其他条件不变情况下再次进行上述实验. 具体实验及比较结果如图 5 所示.

由实验结果可知, 适当降低 Lifted ElGamal 密码系统中随机数 r 的位数, 可以提高协议效率, 降低集合 U 的势较大、数据范围较大及安全参数 κ 位数过大对协议效率带来的影响.

5.4 方案扩展

据我们了解, 目前还没有关于极值和保密计算问题的研究, 关于极差保密计算问题的研究只有文献 [21]. 但文献 [21] 中最值及极差的计算都借助了外包云手段和可信第三方服务器, 安全平台的构建对技术及计算开销有着较高要求. 此外, 文献 [21] 采用的是全同态加密系统, 目前全同态的实现对各方面限制和要求比较高, 理论上可以实现, 然而实际工程中很难实现, 因此不与文献 [21] 中方案做比较.

理论上极差、极值和这两个不同的问题都可以归约到最大 (最小) 值的保密计算问题, 目前有关安全多方计算最值的方案已有不少文献[25−27]. 基于最大值最小值计算极差极值和, 一方面可能需要修改输入参数重复调用最值协议, 另一方面需要注意不泄露中间结果. 我们将从多个方面将文献 [25−27] 中方案与本文方案进行比较. 首先, 从安全性角度出发, 本文所有协议是抗合谋的, 文献 [25−27] 中一些协议仅抵抗部分合谋, 而安全多方计算总是希望抵抗尽可能多的人参与的合谋. 故我们将选取文献 [25−27] 中抗合谋的协议与本文协议在 (1) 复杂性, (2) 能否用于计算极差极值和方面进行效率比较. 令参与者人数为 n, 全集 U 的势为 m. 具体分析如下.

文献 [25] 中协议 3 以 1-r 编码、ElGamal 门限密码系统为基础给出了保密计算最小值的方案, 计算最大值需要变换编码方法后再次调用协议. 计算复杂性方面, 计算最小值时, 生成公钥需要 n 次模指数运算; 加密需要 $2nm$ 次模指数运算; 解密平均需要 $\frac{1+m}{2}\cdot n$ 次模指数运算; 故计算最小值平均需要 $2.5nm+1.5n$ 次模指数运算. 同时计算出最大最小值, 需要改动输入参数、调用两次协议, 平均需要 $5nm+2n$ 次模指数运算. 通信复杂性方面, 构造公钥及加密过程需要 $n(n-1)$ 次通信, 解密过程平均需要 $\frac{1+m}{2}\cdot(n-1)$ 次通信, 同时计算出最大最小值, 平均需要 $(n-1)(2n+m)$ 次通信. 但由于协议是基于乘法同态的, 算出最大最小值的密文也无法再保密计算极差及极值和.

文献 [26] 在解决保密计算最值问题时, 以 0-1 编码及同态为基础, 用保密替换方法保密计算最值. 计算复杂度

方面, 生成公钥需要 n 次模指数运算; 加密过程中共需要 $2nm$ 次模指数运算; 解密时平均需要 $\frac{1+m}{2}\cdot n$ 次模指数运算. 同时计算出最大最小值需要修改输入参数调用两次协议, 共需要 $5nm+2n$ 次模指数运算. 通信复杂性方面, 生成公钥需要 $n-1$ 次通信, 计算过程中需要 $n-1$ 次通信, 解密时平均需要 $\frac{1+m}{2}\cdot n$ 次通信. 同时计算出最大最小值需要修改输入参数调用两次协议, 共需要 $(n-1)(m+4)$ 次通信. 协议用解密密文数组过程中遇到的首 0 或者首 1 元素的位置标记最值在全集 U 中的位置, 所以用该方案计算极差或者极值和会泄露最大最小值.

文献 [27] 以 0-r 编码及加法同态为基础, 同时计算出最大最小值. 计算复杂性方面, 生成公钥需要 n 次运算, 加密数据共需要 $2nm$ 次运算, 解密时平均需要 nm 次运算, 共需要 $3nm+n$ 次运算. 通信复杂度方面, 生成公钥需要 $n-1$ 次通信, 加密数据需要 $n-1$ 次通信, 解密时平均需要 $\frac{3m}{4}\cdot(n-1)$ 次通信, 共需要 $(0.75m+2)(n-1)$ 次运算. 协议用解密密文数组过程中遇到的首非 0 元素的位置标记最值在全集 U 中的位置, 所以用该方案计算极差或者极值和会泄露最大最小值.

本文协议 1 与文献 [25−27] 中协议的效率比较结果如表 4 所示.

表 4 协议效率比较

协议	计算最大及最小值计算开销	计算最大及最小值通信开销	可否计算极差、极值和
协议1	$4nm+7n+2m$	$3(n-1)$	是
文献[25]协议3	$5nm+2n$	$(2n+m)(n-1)$	否
文献[26]协议3	$5nm+2n$	$(m+4)(n-1)$	否
文献[27]协议2	$3nm+n$	$(0.75m+2)(n-1)$	否

从上述协议效率比较结果中可以看出, 只用于计算最大值和最小值时, 与当前已有的保密计算最大及最小值协议相比, 我们的协议在计算开销方面比大多数方案有优势, 在通信开销方面比已有方案有明显优势. 此外, 我们的协议可以直接用于保密计算极差、极值和, 其他最值协议不能用于计算极差、极值和.

6 总 结

本文提出了新的安全计算问题, 即极值和的保密计算问题, 给出了安全多方计算解决极差的方案. 首先针对极值和的保密计算问题设计了新的编码方法. 新编码方法与 Lifted ElGamal 密码相结合解决多方参与、每方拥有一个数据场景下, 极值和的保密计算问题以及极差的保密计算问题, 该方法适用于解决相同场景下数据范围很大 (稀疏集) 时极差、极值和的保密计算问题以及极值和的保密计算问题. 进一步, 将新编码方法调整后与 Paillier 密码系统相结合设计两方参与、每方拥有多个数据情况下分布式数据集合并集的极差的保密计算协议, 分布式数据集合并集的极值和的保密计算协议. 可以看出, 本文提出的问题具有研究价值, 本文设计的协议简单易行且适用于数据范围很大的稀疏集. 后续研究中, 我们将进一步研究数据个数很多、参与人数很多情况下更高效的极差以及极值和保密计算问题的解决方案及恶意模型下极差、极值和保密计算问题的解决方案.

References:

[1] Yao AC. Protocols for secure computations. In: Proc. of the 23rd Annual Symp. on Foundations of Computer Science. Chicago: IEEE Computer Society, 1982. 160–164. [doi: 10.1109/SFCS.1982.38]

[2] Goldreich O, Micali S, Wigderson A. How to play any mental game. In: Proc. of the 19th Annual ACM Symp. on Theory of Computing. New York: ACM, 1987. 218–229. [doi: 10.1145/28395.28420]

[3] Goldwasser S. Multi party computations: Past and present. In: Proc. of the 16th Annual ACM Symp. on Principles of Distributed Computing. Santa Barbara: ACM, 1997. 1–6. [doi: 10.1145/259380.259405]

[4] Yasin S, Haseeb K, Qureshi RJ. Cryptography based E-commerce security: A review. Int'l Journal of Computer Science Issues, 2012, 9(2): 132–137.

[5] Miyajima H, Shigei N, Miyajima H, Norio S. A proposal of profit sharing method for secure multiparty computation. Int'l Journal of

Innovative Computing, Information and Control, 2018, 14(2): 727–735.

[6] Zhao C, Zhao SN, Zhao MH, Chen ZX, Gao CZ, Li HW, Tan YA. Secure multi-party computation: Theory, practice and applications. Information Sciences, 2019, 476: 357–372. [doi: 10.1016/j.ins.2018.10.024]

[7] Collins MJ. Efficient secure multiparty computation of sparse vector dot products. Journal of Discrete Mathematical Sciences and Cryptography, 2018, 21(5): 1107–1117. [doi: 10.1080/09720529.2018.1453623]

[8] Park H, Moon J. Irregular product coded computation for high-dimensional matrix multiplication. In: Proc. of the 2019 IEEE Int'l Symp. on Information Theory. Paris: IEEE, 2019. 1782–1786. [doi: 10.1109/ISIT.2019.8849236]

[9] Liu XM, Choo KKR, Deng RH, Lu RX, Weng J. Efficient and privacy-preserving outsourced calculation of rational numbers. IEEE Trans. on Dependable and Secure Computing, 2018, 15(1): 27–39. [doi: 10.1109/TDSC.2016.2536601]

[10] Kim M, Yang H, Lee J. Private coded matrix multiplication. IEEE Trans. on Information Forensics and Security, 2020, 15: 1434–1443. [doi: 10.1109/TIFS.2019.2940895]

[11] Chen H, Laine K, Rindal P. Fast private set intersection from homomorphic encryption. In: Proc. of the 2017 ACM SIGSAC Conf. on Computer and Communications Security. Dallas: ACM, 2017. 1243–1255. [doi: 10.1145/3133956.3134061]

[12] Egert R, Fischlin M, Gens D, Jacob S, Senker M, Tillmanns J. Privately computing set-union and set-intersection cardinality via bloom filters. In: Proc. of the 20th Australasian Conf. on Information Security and Privacy. Brisbane: Springer, 2015. 413–430. [doi: 10.1007/978-3-319-19962-7_24]

[13] Li G, Wang YD. An improved privacy-preserving classification mining method based on singular value decomposition. Acta Electronica Sinica, 2012, 40(4): 739–744.

[14] Zhu H, Liu F, Li H. Efficient and privacy-preserving polygons spatial query framework for location-based services. IEEE Internet of Things Journal, 2017, 4(2): 536–545. [doi: 10.1109/JIOT.2016.2553083]

[15] Sun GS, Qian Q. Deep learning and visualization for identifying malware families. IEEE Trans. on Dependable and Secure Computing, 2021, 18(1): 283–295. [doi: 10.1109/TDSC.2018.2884928]

[16] Phong LT, Aono Y, Hayashi T, Wang LH, Moriai S. Privacy-preserving deep learning via additively homomorphic encryption. IEEE Trans. on Information Forensics and Security, 2018, 13(5): 1333–1345. [doi: 10.1109/TIFS.2017.2787987]

[17] Liu C, Zhu LH, He XJ, Chen JJ. Enabling privacy-preserving shortest distance queries on encrypted graph data. IEEE Trans. on Dependable and Secure Computing, 2021, 18(1): 192–204. [doi: 10.1109/TDSC.2018.2880981]

[18] Ge SS, Zeng P, Lu RX, Choo KKR. FGDA: Fine-grained data analysis in privacy-preserving smart grid communications. Peer-to-peer Networking and Applications, 2018, 11(5): 966–978. [doi: 10.1007/s12083-017-0618-9]

[19] Essex A. Secure approximate string matching for privacy-preserving record linkage. IEEE Trans. on Information Forensics and Security, 2019, 14(10): 2623–2632. [doi: 10.1109/TIFS.2019.2903651]

[20] Bag S, Hao F, Shahandashti SF, Ray IG. SEAL: Sealed-bid auction without auctioneers. IEEE Trans. on Information Forensics and Security, 2020, 15: 2042–2052. [doi: 10.1109/TIFS.2019.2955793]

[21] Li ZL, Chen LC, Chen ZH, Liu YR. Secure multiparty computation of the maximum and the minimum in cloud environment and its statistics application. Journal of Cryptologic Research, 2019, 6(2): 219–233 (in Chinese with English abstract). [doi: 10.13868/j.cnki.jcr.000297]

[22] Sciancalepore S, Di Pietro R. PPRQ: Privacy-preserving MAX/MIN range queries in IoT networks. IEEE Internet of Things Journal, 2021, 8(6): 5075–5092. [doi: 10.1109/JIOT.2020.3037115]

[23] Zhang Y, Chen QJ, Zhong S. Efficient and privacy-preserving min and *k*th min computations in mobile sensing systems. IEEE Trans. on Dependable and Secure Computing, 2017, 14(1): 9–21. [doi: 10.1109/TDSC.2015.2432814]

[24] Yao YL, Xiong NX, Park JH, Ma L, Liu JF. Privacy-preserving max/min query in two-tiered wireless sensor networks. Computers & Mathematics with Applications, 2013, 65(9): 1318–1325. [doi: 10.1016/j.camwa.2012.02.003]

[25] Dou JW, Ma L, Li SD. Secure multi-party computation for minimum and its applications. Acta Electronica Sinica, 2017, 45(7): 1715–1721 (in Chinese with English abstract). [doi: 10.3969/j.issn.0372-2112.2017.07.023]

[26] Yang XY, Li SD, Kang J. Private substitution and its applications in private scientific computation. Chinese Journal of Computers, 2018, 41(5): 1132–1142 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2018.01132]

[27] Yang YJ, Li SD, Du RM. Private maximum and minimum computation. Journal of Cryptologic Research, 2020, 7(4): 483 –497 (in Chinese with English abstract). [doi: 10.13868/j.cnki.jcr.000383]

[28] Li SD, Xu WT, Wang WL, Zhang MY. Secure maximum (minimum) computation in malicious model. Chinese Journal of Computers, 2021, 44(10): 2076–2089 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2021.02076]

[29] Goldreich O. Foundations of Cryptography. II: Basic Applications. Cambridge: Cambridge University Press, 2004.

[30] Reimer B, Fried R, Mehler B, Joshi G, Bolfek A, Godfrey KM, Zhao N, Goldin R, Biederman J. Brief report: Examining driving behavior in young adults with high functioning autism spectrum disorders: A pilot study using a driving simulation paradigm. Journal of Autism and Developmental Disorders, 2013, 43(9): 2211–2217. [doi: 10.1007/s10803-013-1764-4]

[31] Liu J, Asokan N, Pinkas B. Secure deduplication of encrypted data without additional independent servers. In: Proc. of the 22nd ACM SIGSAC Conf. on Computer and Communications Security. Denver: ACM, 2015. 874–885. [doi: 10.1145/2810103.2813623]

[32] Elgamal T. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. on Information Theory, 1985, 31(4): 469–472. [doi: 10.1109/TIT.1985.1057074]

[33] Paillier P. Public-key cryptosystems based on composite degree residuosity classes. In: Proc. of the 1999 Int'l Conf. on the Theory and Application of Cryptographic Techniques. Prague: Springer, 1999. 223–238. [doi: 10.1007/3-540-48910-X_16]

附中文参考文献:

[13] 李光, 王亚东. 一种改进的基于奇异值分解的隐私保持分类挖掘方法. 电子学报, 2012, 40(4): 739–744.

[21] 李占利, 陈立朝, 陈振华, 刘娅茹. 云环境下多方保密计算最大值、最小值及其统计学应用. 密码学报, 2019, 6(2): 219–233. [doi: 10.13868/j.cnki.jcr.000297]

[25] 窦家维, 马丽, 李顺东. 最小值问题的安全多方计算及其应用. 电子学报, 2017, 45(7): 1715–1721. [doi: 10.3969/j.issn.0372-2112.2017.07.023]

[26] 杨晓艺, 李顺东, 亢佳. 保密替换及其在保密科学计算中的应用. 计算机学报, 2018, 41(5): 1132–1142. [doi: 10.11897/SP.J.1016.2018.01132]

[27] 杨颜璟, 李顺东, 杜润萌. 最大最小值的保密计算. 密码学报, 2020, 7(4): 483–497. [doi: 10.13868/j.cnki.jcr.000383]

[28] 李顺东, 徐雯婷, 王文丽, 张萌雨. 恶意模型下的最大(小)值保密计算. 计算机学报, 2021, 44(10): 2076–2089. [doi: 10.11897/SP.J.1016.2021.02076]

李顺东(1963－), 男, 博士, 教授, 博士生导师, 主要研究领域为密码学, 信息安全.

赵雪玲(1996－), 女, 硕士生, 主要研究领域为密码学, 信息安全.

家珠亮(1992－), 女, 硕士生, 主要研究领域为密码学, 信息安全.