多个字符排序的安全多方计算

李顺东[1)] 亢 佳[1)] 杨晓艺[1)] 窦家维[2)] 刘 新[3)]

[1)](陕西师范大学计算机科学学院 西安 710119)

[2)](陕西师范大学数学与信息科学学院 西安 710119)

[3)](内蒙古科技大学信息工程学院 内蒙古 包头 014010)

摘 要 安全多方计算是国际密码学界的研究热点之一,是信息社会隐私保护的核心技术.密码学者已经研究了很多安全多方计算问题,但是还有更多问题有待研究和解决.该文研究如何保密地将多个字符按照字典序排序,这是一个全新的问题,目前尚没有见到关于这个问题的解决方案.它可以提高数据库保密查询的效率,在信息安全领域有重要的实际意义和广泛的应用前景.为了保密地判断多个字符按照字典序排序的位置关系,该文首先设计了一种新的编码方法,并结合 Paillier 加法同态加密算法、椭圆曲线加法同态加密算法、秘密分割和门限解密算法,设计了三个能够抵抗合谋攻击的多个字符保密排序的高效而简单的协议.利用安全多方计算普遍采用的模拟范例证明了协议在半诚实模型下是安全的,并且分析了协议的正确性,同时给出了协议计算复杂性和通信复杂性的理论分析与实验验证.这些协议都跳出了两两比较进行排序的传统思维框架,具有更高的保密性.最后将保密的字符排序问题的协议应用于解决安全多方数据排序问题上,拓展了可比较数据的范围.

关键词 密码学;安全多方计算;字符排序;云计算;同态加密;秘密分割;门限解密

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2018.01172

Secure Multiparty Characters Sorting

LI Shun-Dong[1)] KANG Jia[1)] YANG Xiao-Yi[1)] DOU Jia-Wei[2)] LIU Xin[3)]

[1)](*School of Computer Science, Shaanxi Normal University, Xi'an* 710119)

[2)](*School of Mathematics and Information Science, Shaanxi Normal University, Xi'an* 710119)

[3)](*School of Information and Engineering, Inner Mongolia University, Baotou, Inner Mongolia* 014010)

Abstract The rapid development of information science and technology has brought us into the information age. In the information age, information that exists in the form of data has become an important strategic resource and digital wealth. Different organizations, institutions, enterprises and persons own different private data. The data has great importance for scientific research, business, human health, personal service, social management etc. To make full use of the data, it is necessary for different parties to cooperatively perform some computation on their private data, but this will pose great challenge to the privacy-preserving of the private data. Secure multiparty computation which was introduced in 1980s is a core technology to protect the privacy in both cooperative computation and network world. It has been applied to electronic commerce, secure electronic voting, private data mining, privacy preserving statistical analysis etc. It is also a research focus in the international cryptographic community in recent years. Cryptographic scholars have studied many secure multiparty computation problems such as secure scientific

收稿日期:2017-07-09;在线发表日期:2017-12-05.本课题得到国家自然科学基金(61272435)、内蒙古自然科学基金项目(2017MS0602)、内蒙古自治区高等学校科学研究项目(NJZY17164)资助.**李顺东**,男,1963 年生,博士,教授,博士生导师,主要从事密码学与信息安全研究. E-mail: shundong@snnu.edu.cn. **亢 佳**,女,1992 年生,硕士研究生,主要研究方向为密码学与信息安全研究. **杨晓艺**,女,1993 年生,博士研究生,主要研究方向为密码学与信息安全研究. **窦家维**,女,1963 年生,博士,副教授,主要研究方向为应用数学与密码学. **刘 新**,男,1983 年生,博士,讲师,主要研究方向为密码学与信息安全.

computation, secure data mining, secure computational geometry, secure statistical analysis, and propose solutions to these problems, but there are more problems need to be studied and solved. This work studies how to privately sort multiple characters lexicographically. This is a completely new problem. To the best of our knowledge, this problem has not been investigated. It has great practical significance and extensive application prospect in information security field. For example, it can improve the efficiency of private database query, and it can sort private numbers. To determine the lexicographic position of multiple characters, we first design a new encoding scheme to encode a private character as a vector. A string or multiple characters can be encoded as a matrix with this scheme. This encoding scheme makes us reduce the private sorting of multiple characters to the private addition of elements of a matrix, and the private sorting can be implemented with additively homomorphic cryptosystems. Then we use the Paillier additively homomorphic cryptosystem, elliptic curve cryptosystem, secret cutting method and threshold decryption cryptosystem, to design three simple, efficient private sorting protocols. The first protocol is constructed with the Paillier cryptosystem and can resist limited collusion attacks. The construction of the second protocol is similar to that of the first one, but added secret cutting technology makes it be able to resist collusion attack of any subset of the parties. The third protocol is constructed based threshold decryption elliptic curve cryptosystem. This protocol can also resist collusion attack of any subset of the parties with shorter keys and lower computational complexity. These protocols jump out the traditional pair-wise comparison framework to sort characters. Therefore they do not leak unnecessary information of private characters, and are appropriate for any characters provided the alphabet of characters is fixed. We use the simulation paradigm which is introduced by Goldreich and is well-accepted in secure multiparty computation research to prove that these protocols are secure in the semi-honest model. Finally, we analyze the correctness, computational complexity and communication complexity of these protocols, and provide an experimental result to show our analysis. As the application of these protocols, we show how to use them to securely sort multiple private data. This shows that our protocols are universal.

Keywords　cryptography; secure multi-party computation; characters sorting; cloud computing; homomorphic encryption; secret sharing; threshold decryption

1　引　言

随着信息技术的飞速发展，需要处理的数据呈爆炸式增长，对数据处理与计算的能力提出了严峻的挑战. 现代社会中，人们对生活质量的追求也需要更精确、更复杂的计算，需要更多的计算资源和更强大的计算能力. 但个体为解决自己的计算问题投资更多的计算资源是不合适的，为了解决这个问题，个体通常要将数据外包给云服务器来存储和计算. 然而，在云计算技术支持的外包计算模型中，云端可能破坏数据的隐私性和外包计算结果. 因此，外包计算中的安全和隐私问题非常重要. 为了避免用户数据的泄露，用户的数据必须经过加密才能存储在云服务器中.

试想如下场景：在医疗云系统中，大量的个人健康信息无法存储于本地设备，需要将用户信息加密后发送给远程医疗服务机构的云服务器. 当主治医生要访问某个病人的健康信息时，常规的做法是将要查询病人的查询条件进行加密，然后利用密文与云存储数据库中所有病人信息记录的密文进行有关的查询操作. 当云服务器上存储的病人信息很多的时候，保密查询的计算复杂性很高，查询效率很低. 如何能从云存储数据库中更加快速、准确地查询到保密数据呢？可以对云服务器上的病人信息进行分类（分类标准可以为病人的姓名、各项指标的排序等），比如说按照病人姓名的每个汉字的首字母排序分类（26 个英文字母的排列顺序，张三的汉字首字母为 ZS，李四的汉字首字母为 LS，那么张三的健康

信息要排在李四之后),这样就可以在主治医生查询某个病人信息的时候,直接将病人按照姓名的排序定位.利用这种分类方法来进行病人信息查询,效率将会大大提高.但是在对云存储服务器上的病人信息进行分类的时候,病人的所有信息都是保密的,假如将病人的姓名看作一个字符串,那么对云存储服务器上的病人信息进行分类的过程可以看成是对多个字符串进行保密排序的过程.所以为了提高查询效率,保密地判断多个字符按照字典序排列的位置关系就很有必要.本文就如何保密地判断多个字符按照字典序排列的位置关系问题设计了相关的解决方案.

安全多方计算(Secure Multiparty Computation, SMC)作为隐私保护和信息安全的关键技术,已成为国内外密码学界的研究热点[1-8].安全多方计算保证在不泄露参与者私有数据的前提下,使这些私有数据能够被参与者利用,并进行保密计算,从而使人们能够最大限度地利用私有数据,而不破坏数据的机密性.安全多方计算问题由 Yao[9] 首先提出,Goldreich、Micali 和 Wigderson 等人[10] 对其进行了深入的研究,奠定了安全多方计算的理论基础[11-13],推动了安全多方计算的发展.

安全多方计算在计算科学中占有重要的地位.很多学者致力于安全多方计算问题的研究,提出了各种安全多方计算问题及其解决方案.所研究的问题可以分为以下几大方向:(1)保密的科学计算问题[14-19];(2)保密的计算几何问题[20-23];(3)保密的数据挖掘问题[24];(4)保密统计分析;(5)保密的数据库查询问题;(6)其他安全多方计算问题.在这些问题中,保密的科学计算问题是最重要的问题之一,也是安全多方计算协议的基本模块.

安全多方数据排序问题是百万富翁问题的自然延伸,这与保密比较大小问题相同,都是保密的科学计算中的重要问题,已有文献提出基于加法同态加密算法的百万富翁问题解决方案[25],然后多次调用基本协议解决了安全多方排序问题.同时还提出了基于模糊贴近度的安全多方协议(只针对数据排序).但是协议可能泄露参与者拥有数据的相对位置.文献[26]虽然利用 ElGamal 加密算法和一个新编码方案实现了安全多方排序,但是对于数据相等的情况,可能会得到不正确的排序结果.文献[27]提出了基于离散对数假设和费马定理的多方保密排序协议,但是该协议安全性较低,不能抵抗两个或者两个以上参与者的合谋攻击.文献[28]基于秘密共享设计了一个高效的解决保密的数据排序问题的方案.该协议在不诚实的参与者个数 $t\geq(n-1)/2$ 时,存在信息泄露的问题.

总的来说,目前对保密的排序问题方面的研究仍然处于初始阶段,还存在以下问题:(1)对于当前安全多方排序问题来说,所提出的大部分解决方案都是基于多次调用提出的百万富翁协议来实现多方排序,会泄露参与者拥有的信息;(2)通过同态加密算法和特殊编码方式已经实现的多方数据排序方案的计算复杂性和通信复杂性仍然较高;(3)仅限于对数据的比较.例如保密地判断多个字符按照字典序排序的位置问题还没有研究.但实际上保密地判断多个字符排序的问题不仅在数据挖掘,数据库查询方面有广泛的实用,也可以解决安全多方数据排序的问题.所以研究字符保密排序问题是非常有意义的.

本文的主要贡献如下:

(1)针对字符排序问题设计了一种全新的编码方法,为解决安全多方数据排序问题提供了新思路,达到了简化问题,降低计算复杂性的目的.

(2)借助于新的编码方法设计了基于 Paillier 加法同态加密算法和秘密分割的可以抵抗不同程度合谋攻击的方案,并对其协议的安全性和正确性进行证明.

(3)设计了基于椭圆曲线加密方案,门限解密的安全性更高的协议,并对协议的安全性和正确性进行证明.

(4)在保密地判断多个字符按照字典序排序位置关系协议的基础上,提出了较大数据的保密排序问题的协议.

本文第 2 节介绍协议中用到的预备知识和安全性定义;第 3~5 节设计具体的协议解决字符保密排序问题;第 6 节对协议的正确性和安全性进行了分析,同时利用模拟实验验证协议的复杂性和效率;第 7 节提出保密数据排序问题的协议;第 8 节对文章进行总结.

2 预备知识

2.1 安全性定义

理想保密计算协议.理想保密计算协议假设有一个值得信赖的第三方(Trusted Third Party,

TTP)，无论在什么情况下，它都决不会泄露信息和恶意传递虚假信息. 因此在可信第三方存在的情况下，参与者 $P_1,\cdots,P_n$ 分别将自己的私有数据 $X_1,\cdots,X_n$ 告诉 TTP，由 TTP 计算 $f(X_1,\cdots,X_n)=(f_1(X_1,\cdots,X_n),\cdots,f_n(X_1,\cdots,X_n))$，然后将结果 $f_1(X_1,\cdots,X_n),\cdots,f_n(X_1,\cdots,X_n)$ 分别告诉参与者 $P_1,\cdots,P_n$，而不泄露任何其他私有信息. 由于参与者 $P_1,\cdots,P_n$ 无法从协议中获得除 $f(X_1,\cdots,X_n)$ 之外的信息，所以该协议是安全性最高的安全多方计算协议，被称为“理想协议”. 任何计算 $f(X_1,\cdots,X_n)$ 的实际安全多方计算协议的安全性都无法超出这个理想协议.

半诚实参与者[29]. 所谓半诚实参与者就是参与者按照协议的规定执行协议，并且不会中途退出协议，也不会欺骗和泄露信息，但是它们可能会试图通过分析和利用协议执行过程中自己所得到的信息来推断额外的信息.

本文假设协议的所有参与者都是半诚实的. 由于研究基于半诚实模型下的安全多方计算是研究恶意模型下安全多方计算的基础，所以绝大部分研究建立在半诚实模型上. 只要能够设计出对半诚实参与者安全的保密计算函数 f 的协议 π，就可以使用文献[29]中基于比特承诺和零知识证明理论设计的编译器将 π 转换为对恶意参与者安全的协议. 在转换后的协议中，一个恶意的参与者将被迫按照半诚实的参与者行事，否则将会被发现. 在研究恶意模型下安全协议的时候，人们总是从半诚实参与者模型入手，寻找防止恶意攻击的方法，并将其添加到协议中，最终形成恶意模型下的安全协议.

假设参与保密计算的参与者为 $P_1,\cdots,P_n$，分别拥有保密数据 $X_1,\cdots,X_n$.

(1) 假设 π 表示计算 f 的协议，协议执行过程中输入为 $X=(X_1,\cdots,X_n)$，下式代表概率多项式时间函数，

$$f(X)=(f_1(X),\cdots,f_n(X)).$$

在执行协议的过程中参与方 P_i 得到的信息序列记为 $view_i^{\pi}(X)=(X_i,r^i,m_1^i,\cdots,m_i^i)$，其中 r^i 表示第 i 个参与方独立的硬币抛掷结果；m_j^i 表示第 P_i 第 j 次收到的信息. 对于部分参与者构成的集合

$$I=\{P_{i1},\cdots,P_{ia}\}\subseteq\{P_1,P_2,\cdots,P_n\}$$

记为 $view_I^{\pi}(X)=(view_{i1}^{\pi}(X),\cdots,view_{ia}^{\pi}(X))$.

定义 1. 半诚实参与者的保密性[29]. 假设协议 π 计算函数 f，如果存在概率多项式时间算法 S，使得对于任意的 $I=\{P_{i1},\cdots,P_{ia}\}\subseteq\{P_1,P_2,\cdots,P_n\}$，满足如下：

$$\{S(I,(X_{i1},\cdots,X_{ia}),f_I(X))\}_{X\in(\{0,1\}^*)^n}\overset{c}{\equiv}\{view_I^{\pi}(X)\}_{X\in(\{0,1\}^*)^n}\quad(1)$$

式中 $\overset{c}{\equiv}$ 表示计算上不可区分，则认为 π 保密计算函数 f.

2.2 Paillier 同态加密算法

同态加密的概念在文献[30]中被首次提出，它可以保证在不影响明文数据机密性的情况下，直接操作密文来完成对明文的计算. 简单来说，对密文的计算等价于明文计算之后再加密.

Paillier 方案[31]的具体过程如下：

密钥生成. 首先选取两个素数 p,q，其中 $N=p\times q$，$\lambda=lcm(p-1,q-1)$ 是 $p-1$ 和 $q-1$ 的最小公倍数. 随机选择一个 $g\in Z_N^*$，使得 $gcd(L(g^{\lambda}\bmod N^2),N)=1$，其中 $L(x)=\dfrac{x-1}{N}$. 算法的公钥为 (g,N)，私钥为 λ.

加密. 选择一个随机数 r，$r<N$，计算

$$c=g^m r^N\bmod N^2.$$

解密.

$$m=\frac{L(c^{\lambda}\bmod N^2)}{L(g^{\lambda}\bmod N^2)}\bmod N^2.$$

该算法是概率加密算法，具有加法同态性. 假设密文

$$E(m_1)=g^{m_1}r_1^N\bmod N^2,$$
$$E(m_2)=g^{m_2}r_2^N\bmod N^2.$$

那么

$$E(m_1+m_2)=g^{m_1+m_2}(r)^N\bmod N^2,$$
$$E(m_1)\times E(m_2)=g^{m_1+m_2}(r_1r_2)^N\bmod N^2=g^{m_1+m_2}(r)^N\bmod N^2.$$

该算法满足如下性质：

$$E(m_1)\times E(m_2)=E(m_1+m_2).$$

2.3 椭圆曲线同态加密算法

椭圆曲线密码体制 ECC(Elliptic Curve Cryptography)是 1985 年由 Miller 和 Koblitz 共同提出的. 其理论基础是定义在有限域上的某一椭圆曲线上的整数点与无穷远点可构成有限交换群. 如果该群的阶包含一个较大的素因子，则其上的离散对数问题是困难的. 与 RSA 算法相比，ECC 具有计算量小、密钥短、对带宽和处理器要求低等优点.

基于椭圆曲线实现 ElGamal 密码体制[32]描述如下：

在使用椭圆曲线密码体制之前,必须设计把明文信息编码到椭圆曲线上的点的编码方法.

明文消息嵌入到椭圆曲线的编码方法[33]

(1)选择一个具有 n 个点的椭圆曲线.假设消息可以用字母表{0,1,…,9,A,…,Z}上的字符来表示.这些字符可以用{0,1,…,35}来代替.

(2)把消息转化成一个每一个数都在 0～35 之间的数字序列.

(3)选择一个辅助基本参数 k,比如设 $k=20$(加密解密双方达成一致)

(4)对于每一个 m, For $i=1$ to $k-1$,令 $x=mk+i$,利用椭圆曲线方程求 y.如果找到就停止.如果找不到就令 $i\leftarrow i+1$,继续找,直到找到为止.实际上,可以找到一点(x,y).这个点就是消息 m 的编码.

(5)解码.点(x,y)解码为$\lfloor (x-1)/k \rfloor$($(x-1)/k$ 的值向下取整).

密钥生成.选定一条椭圆曲线 $EC(a,b)$与其上的一个基点 G,在 Z_n^* 上任意选择一个随机数 h,计算 $H=hG$.(H,G)就是公钥,h 是私钥.

加密.消息编码到 $EC(a,b)$上一点 M,并产生一个随机整数 r,计算密文$\langle C_1,C_2\rangle=\langle m+rH,rG\rangle$.

解密.对密文$\langle C_1,C_2\rangle$,用私钥 h 解密得到明文为

$$M=C_1-hC_2,$$

因为

$$\begin{aligned}C_1-hC_2&=M+rH-h(rG)\\&=M+rH-r(hG)=M.\end{aligned}$$

再对点 M 进行解码就可以得到明文.

该算法是概率加密算法,具有加法同态性.假设密文

$$E(m_1)=\langle M_1+r_1H,r_1G\rangle,$$
$$E(m_2)=\langle M_2+r_2H,r_2G\rangle,$$

那么

$$E(m_1+m_2)=\langle (M_1+M_2)+r_3H,r_3G\rangle,$$
$$\begin{aligned}E(m_1)+E(m_2)&=\langle (M_1+M_2)+(r_1+r_2)H,(r_1+r_2)G\rangle\\&=\langle (M_1+M_2)+r_3H,r_3G\rangle\\&=E(m_1+m_2).\end{aligned}$$

该算法满足如下性质:

$$E(m_1)+E(m_2)=E(m_1+m_2).$$

2.4 门限解密

门限解密[34-35]是安全多方计算中对抗合谋攻击的一个重要工具.在门限解密密码体系中,n 个参与者共同生成一个公钥,每个人持有一部分解密密钥.在加密过程中可以直接使用共同拥有的公钥加密消息,但是需要多个参与者共同合作才能对密文进行解密.如果解密一个消息至少需要 t 个人合作才能解密,少于 t 个人合作时将不能得到解密消息,这种密码体制被称为(t,n)门限密码体制.本文需要朴素的(n,n)门限密码系统抵抗 $n-1$ 个参与者的合谋攻击,可以利用椭圆曲线密码系统构造如下:

密钥生成.选定一条椭圆曲线 $EC(a,b)$与其上的一个基点 G,每个参与者 P_i在 Z_n^* 上任意选择一个私钥 h_i,计算 $H_i=h_iG$,共同生成公钥

$$H=\sum_{i=1}^{n}H_i=\sum_{i=1}^{n}h_iG.$$

加密.将消息编码到 $EC(a,b)$上一点 M,并在 Z_n^* 上任意选择一个随机数 r:$1\leq r\leq n-1$,计算密文$\langle C_1,C_2\rangle=\langle M+rH,rG\rangle$.

解密.对密文$\langle C_1,C_2\rangle$,通过下面解密过程得到明文:

$$M=C_1-\sum_{i=1}^{n}h_iC_2.$$

3 基于 Paillier 加密算法的多个字符保密排序方案

问题描述.假设 n 个参与者 $P_1,\cdots,P_n$分别拥有字符 $s_1,\cdots,s_n$,各个参与者希望能比较各自拥有的字符 $s_1,\cdots,s_n$按照字典序排序的位置关系,并要求在排序结束后各个参与方仅知道自己拥有的字符在整个序列中的次序,而不知道其它参与者拥有的字符的任何信息.

方案思想.多个字符排序的问题即为判断多个字符按照字典序排序的问题.本文将参与者拥有的字符 s_i通过一种编码方式表示成一个 m 维的 0-1 向量,n 个参与者的向量合在一起,构成一个 $n\times m$ 维矩阵,每个参与者分别计算自己拥有的字符 s_i所在列之前的所有列元素值(不包括自己字符所在列的元素值)的累加和并加 1,这个值就代表字符 s_i按照字典序排序的位置.

通过编码方式表示保密字符:在此方案中,假设字典的字母表为 $U=\{u_1,\cdots,u_m\}$($1\leq j\leq m$)且满足 $u_1<\cdots<u_m$.P_i拥有的字符 $s_i\in U$ 且在字母表 U 中的位置记为$(s_i)_{\text{ord}}$.每个参与者将各自的字符 s_i借助于以下编码方法编码成一个向量

$$\boldsymbol{T}_i=(t_{i1},\cdots,t_{im}) \tag{2}$$

具体编码规则如下:

```
for(i=1;i≤n;i++){
  for(j=1;j≤m;j++){
    if(s_i==u_j)   t_ij=1
    else           t_ij=0
  }
}
```

参与者将编码得到的 n 个向量 $\boldsymbol{T}_1,\cdots,\boldsymbol{T}_n$ 按顺序排列组合成如下 $n\times m$ 矩阵 $\boldsymbol{T}$：

$$T=\begin{bmatrix}\boldsymbol{T}_1\\ \boldsymbol{T}_2\\ \cdots\\ \boldsymbol{T}_i\\ \cdots\\ \boldsymbol{T}_n\end{bmatrix}=\begin{bmatrix}t_{11} & t_{12} & \cdots & t_{1j} & \cdots & t_{1m}\\ t_{21} & t_{22} & \cdots & t_{2j} & \cdots & t_{2m}\\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\ t_{i1} & t_{i2} & \cdots & t_{ij} & \cdots & t_{im}\\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\ t_{n1} & t_{n2} & \cdots & t_{nj} & \cdots & t_{nm}\end{bmatrix}.$$

每个参与者分别计算自己拥有的字符所在列之前的所有列元素值(不包括自己字符所在列的元素值)的累加和并加 1,即 $(s_i)_{\mathrm{ord}}$ 列之前所有元素值和再加 1,并将结果记为 w_i.

对于 $1\leq i\leq n$:

$$w_i=\sum_{i=1}^{n}\sum_{j=1}^{(s_i)_{\mathrm{ord}}}t_{ij}=\sum_{i=1}^{n}t_{i1}+\sum_{i=1}^{n}t_{i2}+\cdots+\sum_{i=1}^{n}t_{i((s_i)_{\mathrm{ord}}-1)}-1 \tag{3}$$

最后得到的结果 w_i 的值就代表字符 s_i 按照字典序排序的位置. 如果结果中出现相同的 w_i,那么说明这些值对应的字符是相同的,即这些字符按照字典序排序的位置是一样的.

事实 1.　对于 n 个参与者拥有的字符 $s_1,\cdots,s_n$,均按照式(2)构造向量 $\boldsymbol{T}_i$,并按照式(3)计算 w_i. 如果结果中没有出现相同的 w_i,那么 w_i 的值就代表字符 s_i 按照字典序排序的位置. 如果结果中出现相同的 w_i,那么说明这些值对应的字符是相同的.

证明.　因为对于每个字符来说,通过编码方式编码后,只有在自己出现的位置为 1,其余位置均编码为 0,想要判断自己字符的位置,就是将编码后所在位置之前所有元素相加再加 1 即可得到字符的排列位置.

根据这个事实,上面提出的编码方法和 w_i 的值是我们判断参与者拥有的字符按照字典序排序的依据. 如果直接按照矩阵 $\boldsymbol{T}$ 通过上述描述得到字符的位置序列显然没有保密性可言,所以不能用明文的矩阵完成这项任务,我们选用 Paillier 加密算法来加密矩阵.

以下是利用上述编码方法判断多个字符排序的位置关系的实例(实例 1),为了更清楚地表达,在实例 1 中给出了判断 4 个字符的位置关系的过程.

实例 1.　本文假设有 4 个参与者 $P_1,\cdots,P_4$ 分别拥有字符 $s_1=a$, $s_2=f$, $s_3=q$, $s_4=m$,字母表 U 为 26 个英文的字母表. 那么按照编码方式可以将字符 s_1,s_2,s_3,s_4 分别表示为如下向量 $\boldsymbol{T}_1,\boldsymbol{T}_2,\boldsymbol{T}_3,\boldsymbol{T}_4$:

$$\begin{aligned}\boldsymbol{T}_1&=(t_{11},\cdots,t_{1m})\\&=(1,0,0,0,0,0,0,0,0,0,0,0,0,\\&\quad 0,0,0,0,0,0,0,0,0,0,0,0,0),\\\boldsymbol{T}_2&=(t_{21},\cdots,t_{2m})\\&=(0,0,0,0,0,1,0,0,0,0,0,0,0,\\&\quad 0,0,0,0,0,0,0,0,0,0,0,0,0),\\\boldsymbol{T}_3&=(t_{31},\cdots,t_{3m})\\&=(0,0,0,0,0,0,0,0,0,0,0,0,0,\\&\quad 0,0,0,1,0,0,0,0,0,0,0,0,0),\\\boldsymbol{T}_4&=(t_{41},\cdots,t_{4m})\\&=(0,0,0,0,0,0,0,0,0,0,0,0,1,\\&\quad 0,0,0,0,0,0,0,0,0,0,0,0,0),\end{aligned}$$

并由此得到矩阵 $\boldsymbol{T}$:

$$\boldsymbol{T}=\begin{bmatrix}1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\end{bmatrix}.$$

每个参与者分别计算自己 $(s_i)_{\mathrm{ord}}$ 列之前的所有列元素值(不包括自己字符所在列的元素值)的累加和再加 1. 因为 $(s_1)_{\mathrm{ord}}=1$,所以此时 P_1 得到 $w_1=0+1=1$(即为第一列之前列所有元素的累加和加 1),同理 P_2 得到 $w_2=1+0\times 19+1=2$(即为前 $(s_2)_{\mathrm{ord}}=6$ 列所有元素的累加和再加 1),P_3 得到 $w_3=1+1+1+61\times 0+1=4$(即为前 $(s_3)_{\mathrm{ord}}=17$ 列所有元素值的累加和加 1),P_4 得到 $w_4=1+1+46\times 0+1=3$(即为前 $(s_4)_{\mathrm{ord}}=13$ 列所有元素值之和).

3.1　具体协议

首先应用具有加法同态性的 Paillier 加密算法设计一种多个字符排序的基本方案,具体实现如协议 1.

协议 1.　n 个参与者保密地判断各自拥有的字符按照字典序排序的位置关系.

输入:n 个参与者 $P_1,\cdots,P_n$ 分别拥有字符 $s_1,\cdots,s_n$

输出:参与者 $P_1,\cdots,P_n$ 分别得到自己拥有的字符 s_i 按照字典序排列的位置关系 w_i

1. 假设 (G,E,D) 是 Paillier 同态加密方案,ξ 是设定的安全参数. 参与者 P_1 运行 $G(\xi)$ 生成 Paillier 加密算法的私钥

和公钥,并将公钥公开,私钥保留.

2. 参与者 $P_1,\cdots,P_n$ 分别做如下运算:

(1) P_i 将自己拥有的字符 s_i 借助上述式(2)编码方法将其编码成如下向量

$$\boldsymbol{T}_i=(t_{i1},\cdots,t_{im}).$$

(2) 参与者 P_i 用公钥加密 $\boldsymbol{T}_i$,加密后的数据记为

$$E(\boldsymbol{T}_i)=(E(t_{i1}),\cdots,E(t_{im})).$$

3. 参与者 $P_i(1\leq i\leq n-1)$ 分别做以下计算(此处的向量相乘为 $E(\boldsymbol{T}_i)$ 和 $E(\boldsymbol{T}_{i+1})$ 中对应分量相乘):

for($i=1$; $i\leq n-1$; $i++$){

P_i sends $E(\boldsymbol{T}_i)$ to P_{i+1};

P_{i+1} computes $E(\boldsymbol{T}_{i+1})\leftarrow E(\boldsymbol{T}_i)\times E(\boldsymbol{T}_{i+1})$.

}

4. 参与者 P_n 公布最后得到的向量.

$$E(\boldsymbol{T}_n)=(E(t_{n1}),\cdots,E(t_{nm})).$$

5. $P_i(1\leq i\leq n-1)$ 首先各自选取一个随机数 r_i,并用公钥加密得到 $E(r_i)$. 然后分别按照各自字符的位置 $(s_i)_{\text{ord}}$,计算 $E(\boldsymbol{T}_n)$ 中前 $(s_i)_{\text{ord}}$ 个分量与 $E(r_i)$ 累乘的结果,记为 q_i,并将 q_i 发送给 P_1.

6. P_1 用私钥解密,并将解密结果 $D(q_i)$ 告诉 P_i.

7. P_i 在得到解密结果后,计算

$$w_i=D(q_i)-r_i+1.$$

w_i 即为字符 s_i 按照字典序排序的位置.

3.2 协议 1 性能分析

上述协议的正确性可由事实 1 得到保证. 因为 P_1 拥有私钥,只有他可以解密,所以协议 1 可以抵抗除 P_1 之外的合谋攻击. 但是当 P_1 参与合谋攻击时,比如,参与者 P_1 和 P_4 合谋,会恢复出所有的字符. 为了抵抗各种情形下的合谋攻击,下面分别用秘密分割和门限密码解密系统来设计多个字符的保密排序协议. 协议 1 的安全性分析和证明详见附录 1.

4 基于秘密分割的多个字符保密排序方案

方案思想. 在本文协议 1 中,参与者 P_i 将 $E(\boldsymbol{T}_i)$ 先发送给 P_{i+1},P_{i+1} 计算 $E(\boldsymbol{T}_i)\times E(\boldsymbol{T}_{i+1})$,并将其发送给下一个参与者,以此类推,直到 P_n 得到最后的每个参与者加密向量对应分量的乘积 $E(\boldsymbol{T}_n)$. P_n 将 $E(\boldsymbol{T}_n)$ 宣布,如果 P_1 参与合谋攻击时,比如,参与者 P_1 和 P_4 合谋,会恢复出所有的字符. 所以需要借助秘密分割的方法达到抵抗合谋攻击的目的.

协议 2 的思想与协议 1 基本相同,在协议 2 中,P_1 产生 Paillier 加密算法的公钥和私钥,保留私钥,公布公钥. 每个参与者 P_i 都将各自密文 $E(\boldsymbol{T}_i)$ 中的每个分量值 $E(t_{ij})$ 随机分割成 $k_i(k_i\leq n)$ 份,构成 $\boldsymbol{T}_i$ 的 k_i 份密文 $E(\boldsymbol{T}_i)_1,\cdots,E(\boldsymbol{T}_i)_{k_i}$,即

$$E(\boldsymbol{T}_i)_1:E(t_{i1})_1,\cdots,E(t_{im})_1$$
$$E(\boldsymbol{T}_i)_2:E(t_{i1})_2,\cdots,E(t_{im})_2$$
$$\cdots$$
$$E(\boldsymbol{T}_i)_{k_i}:E(t_{i1})_{k_i},\cdots,E(t_{im})_{k_i}$$

其中 $E(t_{ij})=E(t_{ij})_1\cdots E(t_{ij})_{k_i}$. 然后 P_i 将 $E(\boldsymbol{T}_i)_1,\cdots,E(\boldsymbol{T}_i)_{k_i}$ 发送给 n 个参与者中的 k_i 个参与者进行计算. 每个参与者收到所有份额的数据后,将收到向量的各自分量对应相乘. 后续的工作与协议 1 基本相同.

密文分割方法. 参与者 P_i 需要将自己加密后的数据 $E(\boldsymbol{T}_i)$ 分割成 $k_i(k_i\leq n)$ 份,在 Paillier 加密算法中,加密后的密文 $E(t_{ij})=g^{t_{ij}}r_{ij}^{\mathrm{N}}\bmod N^2$,密文属于 Z_{N^2}. 因为 Paillier 加密算法具有加法同态性,所以为了保证密文分割后还能够合成原来的密文,能够正常解密,分成的 $k_i(k_i\leq n)$ 份密文应满足:

$$E(t_{ij})=E(t_{ij})_1\cdots E(t_{ij})_{k_i}.$$

但是将 $E(\boldsymbol{T}_i)$ 通过因子分解而得到 k_i 个因子存在一定的困难,所以我们利用模运算的性质,选择 k_i 个随机数 $r_1,\cdots,r_{ki}$,使其满足如下:

$$r_1r_2\cdots r_{ki}\equiv 1\bmod N^2,$$

然后将 $E(t_{ij})_1=r_1E(t_{ij})$ 发送给选中的 k_i 个参与者中的第一个,将 $E(t_{ij})_2=r_2$ 发送给第二个,以此类推,将 $E(t_{ij})_{k_i}=r_{k_i}$ 发送给第 k_i 个参与者,直到 k_i 个参与者全部得到密文份额为止. 这样 k_i 个参与者获得的密文乘积为

$$E(t_{ij})_1\cdot\cdots\cdot E(t_{ij})_{k_i}=r_1E(t_{ij})\cdot r_2\cdot\cdots\cdot r_{ki}=E(t_{ij}).$$

由上式可知这样分割得到的 k_i 份密文相乘等于未分割前的密文.

4.1 具体协议

协议 2. n 个参与者保密地判断各自拥有的字符按照字典序排序的位置关系.

输入:n 个参与者 $P_1,\cdots,P_n$ 分别拥有的字符 $s_1,\cdots,s_n$

输出:参与者 $P_1,\cdots,P_n$ 分别得到自己拥有的字符 s_i 按照字典序排列的位置关系 w_i

在以下步骤中向量对应相乘即为向量对应的每个分量相乘.

1. 假设 (G,E,D) 是 Paillier 同态加密方案,ξ 是设定的安全参数. 参与者 P_1 运行 $G(\xi)$ 生成 Paillier 加密算法的私钥和公钥,并将公钥公开,私钥保留.

2. 每个参与者 $P_1,\cdots,P_n$ 分别做如下运算:

(1) P_i 将自己拥有的字符 s_i 借助上述式(2)编码方法将其编码成如下向量

$$\boldsymbol{T}_i=(t_{i1},\cdots,t_{im}),\ 1\leq k\leq m.$$

(2) 参与者 P_i 用公钥加密 $\boldsymbol{T}_i$,加密后的数据记为

$$E(\boldsymbol{T}_i)=(E(t_{i1}),\cdots,E(t_{im})).$$

(3) P_i将密文$E(\boldsymbol{T}_i)$分成k_i份分别发送给n个参与者中的k_i个.

(4) P_i把自己收到的所有密文向量对应分量相乘得到新的密文向量

$$E(\boldsymbol{T}_i')=(E(t_{i1}'),\cdots,E(t_{im}')).$$

(5) P_i将$E(\boldsymbol{T}_i')$发送给P_1.

3. 参与者P_1计算所有参与者发送的密文向量对应的分量积

$$\begin{aligned}E(\boldsymbol{T})&=E(\boldsymbol{T}_1')\times\cdots\times E(\boldsymbol{T}_n')\\&=((E(t_{11}')\times E(t_{21}')\times E(t_{n1}')),\cdots,\\&\quad E(t_{1m}')\times E(t_{2m}')\times E(t_{nm}'))).\end{aligned}$$

并且将$E(\boldsymbol{T})$公布.

4. $P_i(1\leq i\leq n-1)$首先各自选取一个随机数r_i,并用公钥加密得到$E(r_i)$.然后分别按照各自字符的位置$(s_i)_{\text{ord}}$,计算$E(\boldsymbol{T})$中前$(s_i)_{\text{ord}}$个分量与$E(r_i)$累乘的结果,记为q_i,并将q_i发送给P_1.

5. P_1用私钥解密,并将解密结果$D(q_i)$告诉P_i.

6. P_i在得到解密结果后,计算$w_i=D(q_i)-r_i+1$.

w_i即为字符s_i按照字典序排序的位置.

4.2　协议 2 性能分析

每个参与者P_i将$E(\boldsymbol{T}_i)$分割成k_i份随机发给n个参与者中的k_i个,分发完成后,每个参与者将自己收到的密文对应分量相乘,得到

$$E(\boldsymbol{T}_i')=\{E(t_{i1}'),\cdots,E(t_{im}')\}.$$

由 Paillier 算法的加法同态性可得到以下事实.

事实 2.　在协议 2 中,对于每个参与者P_i,当$1\leq j\leq m$均有下式成立:

$$\prod_{i=1}^{n}E(t_{i1}')\equiv E(\sum_{i=1}^{n})\bmod N^2.$$

正确性分析. 根据协议 2,每个参与者都将拥有的字符经过式(2)编码为一个特殊的向量,又经过式(3)得到最后的排序结果.由事实 1 可知w_i的值就代表P_i拥有的字符按照字典序排序的位置.虽然协议 2 利用秘密分割的方法抵抗合谋攻击,但是由事实 2 可知

$$E(t_{1j}')\cdot\cdots\cdot E(t_{nj}')\equiv E(t_{1j}+\cdots+t_{nj})\bmod N^2.$$

所以仍可得到w_i的值就代表P_i拥有的字符s_i按照字典序排序的位置.

安全性分析. 关于协议 2 的安全性,有如下定理.

定理 1.　半诚实模型下,基于秘密分割的多个字符保密排序协议 2 是安全的.

证明.　在协议 2 中,每个参与者P_i都将自己的密文随机分成k_i份,自己保留其中一份.假设某个参与者能获得P_i加密后向量的k_i-1份密文,即使拥有私钥,也不能够得到任何关于$E(\boldsymbol{T}_i)$的消息.因为$E(t_{ij})=E(t_{ij})_1\cdots E(t_{ij})_{k_i}$,只有将分割的$k_i$份密文相乘才能得到$E(t_{ij})$.就算拥有私钥,也只能解密得到$E(t_{ij})_1\cdots E(t_{ij})_{k_i-1}$,不能够得到$E(t_{ij})$.

P_i将自己的密文随机分成k_i份,k_i是不确定的,每个参与者所选定的k_i也可能不同.将密文份额发送给n个参与者中的任意k_i个参与者,可能自己会留一份,也可能全部发送给别的参与者,其他参与者不知道P_i将这些密文份额具体发送给哪些参与者.攻击者想要获得关于$E(\boldsymbol{T}_i)$的消息,需要和所有收到$E(\boldsymbol{T}_i)$份额的参与者合谋.如果P_i将分成的密文份额自己留了一份,那么攻击者即使与P_i之外的其他参与者合谋都不能够得到任何关于$E(\boldsymbol{T}_i)$的消息.因此,协议 2 是安全的.

以下是关于协议 2 安全性的具体分析:

假设所有参与者为集合$P=\{P_1,\cdots,P_n\}$,合谋参与者集合为$I=\{P_{i1},\cdots,P_{ia}\}$.

(1) 共谋 1. P_1不参与合谋,I想要得到参与者$P_i\in P\backslash I$的字符s_i.

s为所有参与者拥有字符的集合且$s=(s_1,\cdots,s_n)$.令$s_I=(s_2,\cdots,s_{i-1},s_{i+1},\cdots,s_n)$通过构造使下式成立的概率多项式时间模拟器$S$来证明此情况下协议的安全性:

$$\{S(I,s_I,f_I(s))\}_{s\in(\{0,1\}^*)^n}\overset{c}{\equiv}\{view_I^{\pi}(s)\}_{s\in(\{0,1\}^*)^n}.$$

S的模拟过程如下:

① 给定输入$(I,s_I,f_I(s))$,即$(I,s_I,f_I(s))=(I,(s_2,\cdots,s_{i-1},s_{i+1},\cdots,s_n),f_I(s))$,$S$随机选择两个字符$s_1'$和$s_i'$,使得$f_I(s)=f_I(s')$,在这里$s'=(s_1',s_2,\cdots,s_{i-1},s_i',s_{i+1},\cdots,s_n)$.

② 模拟器S将s'中的每个字符s_i'编码为向量$\boldsymbol{T}_1^*,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_i^*,\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n$.然后模拟器$S$利用公钥加密$\boldsymbol{T}_1^*,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_i^*,\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n$得到密文向量

$$E(\boldsymbol{T}_1^*),\cdots,E(\boldsymbol{T}_{i-1}),E(\boldsymbol{T}_i^*),E(\boldsymbol{T}_{i+1}),\cdots,E(\boldsymbol{T}_n).$$

③ 模拟器S将密文向量

$$E(\boldsymbol{T}_1^*),\cdots,E(\boldsymbol{T}_{i-1}),E(\boldsymbol{T}_i^*),E(\boldsymbol{T}_{i+1}),\cdots,E(\boldsymbol{T}_n).$$

随机分成k_i份,按照协议 2 发给其他参与者.

④ 模拟器S将收到的新的密文份对应分量相乘得到新的密文向量

$$E(\boldsymbol{T}_1^*),\cdots,E(\boldsymbol{T}_{i-1}^*),E(\boldsymbol{T}_i^*),E(\boldsymbol{T}_{i+1}^*),\cdots,E(\boldsymbol{T}_n^*).$$

⑤ 模拟器S按照协议 2 对所有的密文向量的对应分量相乘,得到$E(\boldsymbol{T}')$,选取随机数r_i',并用公钥加密得到$E(r_i')$.然后分别按照各自字符的位置$(s_i')_{\text{ord}}$,计算$E(\boldsymbol{T}')$中前$(s_i')_{\text{ord}}$个分量与$E(r_i')$累乘

的结果，记为 q_i'，并将 q_i' 发送给 P_1.

⑥因为 $P_1 \notin I$，S 没有解密密钥，不能对密文解密.

在协议 2 中：

$$\begin{aligned}view_I^{\pi}(s)=&\{view_2^{\pi}(s),\cdots,view_{i-1}^{\pi}(s),\\&view_{i+1}^{\pi}(s),\cdots,view_n^{\pi}(s)\}\\=&\{(s_2,\cdots,s_{i-1},s_{i+1},\cdots,s_n),\\&(\boldsymbol{T}_2,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n),\\&(E(\boldsymbol{T}_2),\cdots,E(\boldsymbol{T}_{i-1}),E(\boldsymbol{T}_{i+1}),\cdots,E(\boldsymbol{T}_n)),\\&(E(\boldsymbol{T}_2'),\cdots,E(\boldsymbol{T}_{i-1}'),E(\boldsymbol{T}_{i+1}'),\cdots,E(\boldsymbol{T}_n')),\\&E(\boldsymbol{T}),D(Q),(w_2,\cdots,w_{i-1},w_{i+1},\cdots,w_n)\},\end{aligned}$$

$$\begin{aligned}S(I,(s_2,\cdots,s_n),f_I(s))=&\{I,(s_2,\cdots,s_{i-1},s_{i+1},\cdots,s_n),\\&(\boldsymbol{T}_2,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n),\\&(E(\boldsymbol{T}_2),\cdots,E(\boldsymbol{T}_{i-1}),E(\boldsymbol{T}_{i+1}),\cdots,E(\boldsymbol{T}_n)),\\&(E(\boldsymbol{T}_2^*),\cdots,E(\boldsymbol{T}_{i-1}^*),E(\boldsymbol{T}_{i+1}^*),\cdots,E(\boldsymbol{T}_n^*)),\\&E(\boldsymbol{T}'),D(Q'),(w_2',\cdots,w_{i-1}',w_{i+1}',\cdots,w_n')\},\end{aligned}$$

因为 $(w_2,\cdots,w_{i-1},w_{i+1},\cdots,w_n)=(w_2',\cdots,w_{i-1}',w_{i+1}',\cdots,w_n')$，而且 Paillier 加密算法加密后的数据是计算不可区分的，所以上述模拟过程得到的消息序列与实际执行过程中得到的消息序列是计算不可区分的，即

$$\{S(I,s_I,f_I(s))\}_{s\in(\{0,1\}^*)^n}\overset{c}{\equiv}\{view_I^{\pi}(s)\}_{s\in(\{0,1\}^*)^n}.$$

由此可知：半诚实模型下，基于秘密分割的多个字符保密排序协议 2 是安全的.

(2)共谋 2. P_1 参与合谋，I 想要得到 $P_i\in P\backslash I$ 的字符 s_i.

P_i 将 s_i 编码为向量 $\boldsymbol{T}_i$，将加密后得到的 $E(\boldsymbol{T}_i)$ 随机分成 k_i 份发送给 n 个参与者中的任意 k_i 个，其他参与者不知道 P_i 将这些密文份额具体发送给哪些参与者. I 中合谋者不知道收到的密文份额是不是 $E(\boldsymbol{T}_i)$ 的全部，所以合谋者不能得到关于 s_i 的任何信息. 在协议 2 执行完之后，除 P_i 之外的所有参与者合作可以得到 $D(q_{(s_i)_{\mathrm{ord}}})$ 的值，但是由于随机数 r_i 是只有参与者 P_i 自己知道的，所以即使其他参与者知道 $D(q_{(s_i)_{\mathrm{ord}}})$ 的值，也不能得到 w_i，继而无法知道关于 s_i 的任何信息.

同样，存在概率多项式时间算法 S，使得下式成立：

$$\{S(I,s_I,f_I(s))\}_{s\in(\{0,1\}^*)^n}\overset{c}{\equiv}\{view_I^{\pi}(s)\}_{s\in(\{0,1\}^*)^n}.$$

S 的模拟过程如下：

①给定输入 $(I,s_I,f_I(s))$，即 $(I,s_I,f_I(s))=(I,(s_1,\cdots,s_{i-1},s_{i+1},\cdots,s_n),f_I(s))$，$S$ 随机选择一个字符 s_i'，使得 $f_I(s)=f_I(s')$，在这里 $s'=(s_1,s_2,\cdots,s_{i-1},s_i',s_{i+1},\cdots,s_n)$.

②模拟器 S 将 s' 中的每个字符 s_i' 编码为向量 $\boldsymbol{T}_1,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_i^*,\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n$. 然后模拟器 S 利用公钥加密 $\boldsymbol{T}_1,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_i^*,\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n$ 得到密文向量

$$E(\boldsymbol{T}_1),\cdots,E(\boldsymbol{T}_{i-1}),E(\boldsymbol{T}_i^*),E(\boldsymbol{T}_{i+1}),\cdots,E(\boldsymbol{T}_n).$$

③模拟器 S 将密文向量

$$E(\boldsymbol{T}_1),\cdots,E(\boldsymbol{T}_{i-1}),E(\boldsymbol{T}_i^*),E(\boldsymbol{T}_{i+1}),\cdots,E(\boldsymbol{T}_n).$$

随机分成 k_i 份，按照协议 2 发给其他参与者.

④模拟器 S 将收到的新的密文份额对应分量相乘得到新的密文向量

$$E(\boldsymbol{T}_1^*),\cdots,E(\boldsymbol{T}_{i-1}^*),E(\boldsymbol{T}_i^*),E(\boldsymbol{T}_{i+1}^*),\cdots,E(\boldsymbol{T}_n^*).$$

⑤模拟器 S 按照协议 2 对所有的密文向量的对应分量相乘，得到 $E(\boldsymbol{T}')$，选取随机数 r_i'，并用公钥加密得到 $E(r_i')$. 然后分别按照各自字符的位置 $(s_i')_{\mathrm{ord}}$，计算 $E(\boldsymbol{T}')$ 中前 $(s_i')_{\mathrm{ord}}$ 个分量与 $E(r_i')$ 累乘的结果，记为 q_i'，并将 q_i' 发送给 P_1.

在协议 2 执行完之后，假设 I 中合谋者收到的密文份额是 $E(\boldsymbol{T}_i)$ 的全部，除 P_i 之外的所有参与者合作可以得到 $D(q_i)$ 的值，但是由于随机数 r_i 是只有 P_i 自己知道的，所以即使其他参与者知道 $D(q_i)$ 的值，也不能得到 w_i，继而无法知道任何关于 s_i 的信息. 假设 I 中合谋者收到的密文份额不是 $E(\boldsymbol{T}_i)$ 的全部，那么即使 $P_1\in I$，也不能得到 $D(q_i)$，因而无法得到关于 s_i 的任何信息.

在协议 2 中：

$$\begin{aligned}view_I^{\pi}(s)=&\{view_1^{\pi}(s),\cdots,view_{i-1}^{\pi}(s),\\&view_{i+1}^{\pi}(s),\cdots,view_n^{\pi}(s)\}\\=&\{(s_1,\cdots,s_{i-1},s_{i+1},\cdots,s_n),\\&(\boldsymbol{T}_1,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n),\\&(E(\boldsymbol{T}_1),\cdots,E(\boldsymbol{T}_{i-1}),E(\boldsymbol{T}_{i+1}),\cdots,E(\boldsymbol{T}_n)),\\&(E(\boldsymbol{T}_1'),\cdots,E(\boldsymbol{T}_{i-1}'),E(\boldsymbol{T}_{i+1}'),\cdots,E(\boldsymbol{T}_n')),\\&E(\boldsymbol{T}),D(Q),(w_1,\cdots,w_{i-1},w_{i+1},\cdots,w_n)\},\end{aligned}$$

$$\begin{aligned}S(I,(s_1,s_2,\cdots,s_n),f_I(s))=&\\\{&I,(s_1,\cdots,s_{i-1},s_{i+1},\cdots,s_n),\\&(\boldsymbol{T}_1,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n),\\&(E(\boldsymbol{T}_1),\cdots,E(\boldsymbol{T}_{i-1}),E(\boldsymbol{T}_{i+1}),\cdots,E(\boldsymbol{T}_n)),\\&(E(\boldsymbol{T}_1^*),\cdots,E(\boldsymbol{T}_{i-1}^*),E(\boldsymbol{T}_{i+1}^*),\cdots,E(\boldsymbol{T}_n^*)),\\&E(\boldsymbol{T}'),D(Q'),(w_1',\cdots,w_{i-1}',w_{i+1}',\cdots,w_n')\}.\end{aligned}$$

因为 $(w_2,\cdots,w_{i-1},w_{i+1},\cdots,w_n)=(w_2',\cdots,w_{i-1}',w_{i+1}',\cdots,w_n')$，而且 Paillier 加密算法加密后的数据是计算不可区分的，所以上述模拟过程得到的消息序列与实际执行过程中得到的消息序列是计算不可区分的，即

$\{S(I,s_I,f_I(s))\}_{s\in(\{0,1\}^*)^n}\overset{c}{\equiv}\{view_I^{\pi}(s)\}_{s\in(\{0,1\}^*)^n}$.

因此可知在此情况下基于秘密分割的多个字符保密排序协议 2 是安全的.

(3) 共谋 3. P_1 参与合谋，I 想要得到集合 $P\backslash I$ 中参与者们拥有的字符.

很明显(2)中即为集合 $P\backslash I$ 中只有一个元素的情况，当集合 $P\backslash I$ 中有两个或者更多的元素时，可按照上述(2)中的方式分析. 在这种情况下，完全类似于上述两种情况，存在概率多项式时间算法 S，使得下式成立：

$\{S(I,s_I,f_I(s))\}_{s\in(\{0,1\}^*)^n}\overset{c}{\equiv}\{view_I^{\pi}(s)\}_{s\in(\{0,1\}^*)^n}$.

综合上述 3 种情况，可得出半诚实模型下，基于秘密分割的多个字符保密排序协议 2 是安全的.

5　基于门限解密的多个字符保密排序方案

在方案 3 中利用门限椭圆曲线加密系统设计了一个更安全，更高效的协议.

5.1　具体协议

协议 3.　n 个参与者保密地判断各自拥有的字符按照字典序排序的位置关系.

输入：n 个参与者 $P_1,\cdots,P_n$ 分别拥有的字符 $s_1,\cdots,s_n$

输出：参与者 $P_1,\cdots,P_n$ 分别得到自己拥有的字符 s_i 按照字典序排列的位置关系 w_i

1. n 个参与者 $P_1,\cdots,P_n$ 首先选定一条椭圆曲线 $EC(a,b)$，G 为其上基点. 每个参与者分别选择一个私钥 h_i：其中 $2\le h_i\le n-1$ 且 $1\le i\le n$，计算 $H_i=h_iG$，共同生成公钥

$$H=\sum_{i=1}^{n}H_i=\sum_{i=1}^{n}h_iG.$$

将公钥 H 公开，私钥 h_i 各自保留.

2. 每个参与者 $P_1,\cdots,P_n$ 分别做如下运算：

(1) P_i 将自己拥有的字符 s_i 借助 0-1 编码将其编码成如下向量

$$\boldsymbol{T}_i=\{t_{i1},\cdots,t_{im}\},\ 1\le j\le m.$$

(2) 参与者 P_i 将向量 $\boldsymbol{T}_i$ 借助明文消息嵌入到椭圆曲线的编码方法编码到 $EC(a,b)$ 上作为椭圆曲线上的点

$$M_i=\{M_{i1},\cdots,M_{im}\}.$$

(3) P_i 在 Z_n^* 上任意选择 m 个随机数 r_{ij} $(1\le j\le m)$：$1\le r_{ij}\le n-1$，加密 M_i，得到如下：

$$E(M_i)=\{\langle M_{i1}+r_{i1}H,r_{i1}G\rangle,\cdots,\langle M_{im}+r_{im}H,r_{im}G\rangle\}.$$

3. 参与者 $P_1,\cdots,P_n$ 将加密后的数据 $E(M_i)$ $(1\le i\le n)$ 表示成如下 $n\times m$ 矩阵 $E(\boldsymbol{M})$：

$$E(\boldsymbol{M})=\begin{bmatrix}\langle M_{11}+r_{11}H,r_{11}G\rangle & \cdots & \langle M_{1m}+r_{1m}H,r_{1m}G\rangle\\ \langle M_{21}+r_{21}H,r_{21}G\rangle & \cdots & \langle M_{2m}+r_{2m}H,r_{2m}G\rangle\\ \cdots & \cdots & \cdots\\ \langle M_{n1}+r_{n1}H,r_{n1}G\rangle & \cdots & \langle M_{nm}+r_{nm}H,r_{nm}G\rangle\end{bmatrix}.$$

4. 参与者 P_i 选择随机数 R_i，用公钥加密得到 $E(R_i)=\langle R_i+r_iH,r_iG\rangle$，计算 $(s_i)_{\mathrm{ord}}$ 列之前的所有列元素值(不包括自己字符所在列的元素值)与 $E(R_i)$ 累加的和，将结果记为 $\langle C_{i1},C_{i2}\rangle$.

5. 参与者 P_i $(1\le i\le n-1)$ 计算 h_iC_{i2}，并将计算结果与 C_{i2} 发送给 P_{i+1}. P_{i+1} 计算 $h_iC_{i2}+h_{i+1}C_{i2}$，发送给下一个参与者，以此类推 最终参与者 P_n 将得到

$$z_i=h_1C_{i2}+\cdots+h_nC_{i2}=(h_1+\cdots+h_n)C_{i2}.$$

6. 参与者 P_n 公布 $Z=\{z_1,\cdots,z_n\}$. P_i $(1\le i\le n)$ 从 Z 中取出自己对应的 z_i，计算 $w_i=C_{i1}-z_i-R_i+1$. w_i 的值就代表字符 s_i 按照字典序排序的位置.

5.2　协议 3 性能分析

正确性分析. 由于椭圆曲线密码体制具有加法同态性质，即对密文做加法运算等于对相应的明文做加法运算后再加密. 参与者 P_i 拥有的字符 s_i 在字母表 U 中的位置是 $(s_i)_{\mathrm{ord}}$，那么 P_i 从加密后的矩阵 $E(\boldsymbol{M})$ 中计算出的结果对于 $1\le i\le n$，有如下性质：P_i $(1\le i\le n)$ 计算 $(s_i)_{\mathrm{ord}}$ 所在列之前的所有矩阵 $E(\boldsymbol{M})$ 的对应元素值与 1(不包括自己的元素值)累加的和就是字符 s_i 按照字典序顺序排列的位置.

$$\langle C_{i1},C_{i2}\rangle=\left\langle\sum_{j=1}^{(s_i)_{\mathrm{ord}}}M_{ij}+R_i+\left(\sum_{j=1}^{(s_i)_{\mathrm{ord}}}r_{ij}+r_i\right)H,\left(\sum_{j=1}^{(s_i)_{\mathrm{ord}}}r_{ij}+r_i\right)G\right\rangle.$$

P_n 将得到

$$\begin{aligned}z_i&=h_1C_{i2}+\cdots+h_nC_{i2}\\&=(h_1+\cdots+h_n)C_{i2}\\&=(h_1+\cdots+h_n)\left(\sum_{j=1}^{(s_i)_{\mathrm{ord}}}r_{ij}+r_i\right)G\\&=\left(\sum_{j=1}^{(s_i)_{\mathrm{ord}}}r_{ij}+r_i\right)H.\end{aligned}$$

参与者 P_n 公布解密结果后，P_i $(1\le i\le n)$ 从 Z 中取出自己对应的 z_i，计算

$$w_i=C_{i1}-z_i-R_i=\sum_{j=1}^{(s_i)_{\mathrm{ord}}}M_{ij}.$$

即参与者 P_i $(1\le i\le n)$ 计算 $(s_i)_{\mathrm{ord}}$ 所在列之前的所有矩阵 $E(\boldsymbol{M})$ 的对应元素值与 1 累加(不包括自己的元素值)的和就是字符 s_i 按照字典序顺序排列的位置.

因此，协议 3 是正确的.

安全性分析. 协议的安全性是基于椭圆曲线加密体制的安全性. 由于门限椭圆曲线的公钥是由所有参与者共同产生的，即 $H=\sum_{i=1}^{n}h_iG$，其中 h_i 是参与者 P_i 所持有的私钥碎片，如果想解密的话，就必须拥有全部参与者的私钥碎片. 所以整个解密的

过程都必须要全部参与者参与,因而可以抵抗合谋攻击.

在计算过程中,每个参与者 P_i 对外仅公布了加密信息 $E(M_i)=\{\langle M_{i1}+r_{i1}H,r_{i1}G\rangle,\cdots,\langle M_{im}+r_{im}H,r_{im}G\rangle\}$. 在解密过程中对外也仅公布了加密信息 C_{i2},由椭圆曲线密码体制的安全性可知,在协议解密过程中如果 P_i 没有参与,将无法解密得到 z_i,也无法得到 w_i. 因此在解密过程中,P_i 的字符 s_i 是完全保密的. 我们给出下面的定理,仅给出证明思路,详细的证明过程省略.

定理 2. 在半诚实模型下,基于门限密码系统的多个字符保密排序协议 3 是安全的.

证明思路. 证明协议的安全性需要构造满足式(1)的模拟器 S. 根据语义安全的同态加密算法的性质,如果没有私钥,应用概率公钥系统加密的任何信息都是计算不可区分的,因此只要有一个参与者不合谋,对其他合谋者来说,他们实际执行协议时获得的 $view$ 和用满足所有字符排序不变的任意一组输入进行模拟所得到的信息序列是计算不可区分的,所以只要在式(1)中令 $S(I,(s_{i1},\cdots,s_{ia}),f_I(s))$ 为模拟过程中的 $view$,即可使式(1)满足.

6 复杂性分析

本文的协议是用同态加密算法解决字符按照字典序顺序排序的问题,基本运算都是模指数运算(忽略协议执行过程中需要用到的乘法运算). 应用 Paillier 加密系统加密或者解密一次需要进行两次模指数运算. 应用椭圆曲线加密系统进行的是模加运算,模加运算的次数与加密过程中所选随机数 r 的二进制位数有关. 文中协议 1、2、3 的性能分析见表 1.

表 1 协议复杂性比较

	本文协议 1	本文协议 2	本文协议 3
计算复杂性	$2n(m+2)$	$2n(m+2)$	$O(nlogr)$
通信复杂性	$O(n)$	$O(n)$	$O(n^2)$
安全性	抵抗除 P_1 外的合谋攻击	抵抗合谋攻击	抵抗合谋攻击

6.1 计算复杂性分析

在本文协议 1 中,每个参与者都需要对编码后的序列元素进行加密,n 个参与者需要进行 $2mn$ 次模指数运算,而且在协议过程中每个需要加密一次随机数,所以协议 1 在加密过程中需要进行 $2n(m+1)$ 次模指数运算. 最后 P_1 对 Q 进行解密,需要 $2n$ 次模指数运算. 所以协议 1 共需要 $2n(m+2)$ 次模指数运算,计算复杂性为 $2n(m+2)$.

在本文协议 2 中,每个参与者都需要对编码后的序列元素进行加密,n 个参与者需要进行 $2mn$ 次模指数运算;利用第 4 节中的密文分割方法将密文分割成 k_i 部分,在这个过程中只进行了乘法运算. 最后 P_1 对 Q 进行解密,需要 $2n$ 次模指数运算. 所以协议 1 共需要 $2n(m+2)$ 次模指数运算,计算复杂性为 $2n(m+2)$.

在本文协议 3 中,参与者都需要对编码后的序列元素进行加密,而且在协议过程中每个参与者需要加密一次随机数,所以协议 3 共加密 $n(m+1)$ 次. 参与者利用自己的私钥对密文联合解密,共解密 n 次. 加密和解密过程均需要进行模加运算,所以协议 3 的计算复杂性是 $O(n\log r)$ 模加运算(r 表示加密过程中的随机数且 $0\leq r\leq p-1$).

6.2 通信复杂性

衡量通信复杂度的指标一般用协议交换信息的比特数,或者用通信轮数,在安全多方计算研究中通常用轮数.

在协议 1 中,每个参与者需要将加密后的密文发送给 P_n,P_n 将收到的密文做运算后发送给 P_1 解密,在这个过程中需要 n 轮通信. 最后 P_1 解密,并且将解密结果告诉 P_i,需要 $n-1$ 轮通信. 所以协议 1 共需要 $2n-1$ 轮通信,通信复杂性为 $O(n)$.

在协议 2 中,每个参与者 P_i 将自己的密文分成 k_i 份发送给其他 k_i 个参与者,需要 $n\varepsilon k_i$ 轮通信(每个参与者的 k_i 不同). 然后每个参与者将收到的密文份额做运算后发送给 P_1 解密,这个过程中需要 $n-1$ 轮通信. 最后 P_1 解密,并且宣布结果,需要 $n-1$ 轮通信. 所以协议 1 共需要 $n\varepsilon k_i+2n-2$ 轮通信,通信复杂性为 $O(n)$.

在协议 3 中所有参与者构造公钥需要 $n-1$ 轮通信,加密过程宣布 M_i',需要 1 轮通信,解密过程每个参与者将自己得到的 C_{i2} 发送给其他参与者,需要 $n(n-1)$ 轮通信,P_n 宣布 Z,需要 1 次通信,所以协议 3 共需要 n^2 轮通信,通信复杂性为 $O(n^2)$.

6.3 实验数据分析

实验测试环境. Windows 10 64 位操作系统,Intel(R) Core(TM) i5-6600 处理器 CPU@3.30GHz,8.00GB 内存,用 Java 语言在 MyEclipse 上运行实现. 本文所做模拟实验均在此环境下进行.

实验方法. 我们通过模拟实验来测试执行协

议 1、协议 2 和协议 3 所用的时间，可通过协议执行的时间来验证方案的效率. 本实验中，我们的底层协议(Paillier 算法，椭圆曲线加密算法)是使用了现成的开源包(https://www.csee.umbc.edu/~kunliu1/research/Paillier.html，https://sourceforge.net/projects/jecc/files/jecc-alpha/jecc-alpha1.1/). 实验设定 $m=26$，参与者数分别为 $n=3,4,303$. 为使数据准确，对 n 的每个设定值进行 1000 次模拟实验测试，统计协议执行时间的平均值(忽略协议中的预处理时间). 图 1 描述了判断字符排序的执行时间随参与者个数增长的变化规律.

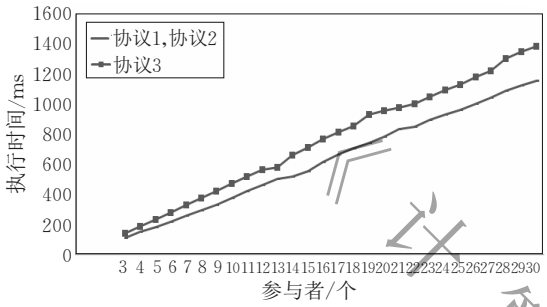

图 1　字符排序的执行时间随参与者个数增长的变化规律

在协议 1 和协议 2 中，底层 Paillier 算法设定素数大小是 512 位比特，生成的大素数 p,q 确为大素数的概率至少为 $1\sim2^{-64}$. 我们基于底层加密算法对协议中编码后的 0,1 进行加密解密运算，执行一次协议 1(协议 2)所需时间包括 $n\times m$ 次加密运算时间，1 次解密运算时间，密文乘法运算时间(因为协议 2 中密文随机分成 k_i 份的过程可看作由预处理过程完成的，所以忽略不计). 在协议 3 中，底层椭圆曲线满足 $y^2=x^3+ax+b(\mathrm{mod}\ p)$，其中 p 为 256 位，大整数 a,b 均为 256 位. 私钥长度最大限值为 273 位. 我们基于底层椭圆曲线加密算法对协议中编码后的 0,1 进行加密解密运算，执行一次协议 3 所需时间包括 $n(m+1)$ 次加密运算时间，n 次解密运算时间.

由图 1 可知字符排序的执行时间随参与者个数增长而线性增加. 基于秘密分割的多个字符保密排序方案效率高于基于门限解密的多个字符保密排序方案.

7　应　用

本节我们将利用多个字符保密排序协议来解决安全多方数据排序的问题. 在日常生活中，年龄、工资、成绩等都属于一定的较小的范围，保密地比较像这些一定范围内数据的排序关系有很大的实用价值[36]. 例如某行业的几个公司想通过员工的工资(员工的工资是分等级的，不同等级的员工工资不同，但是所有的工资均是在一定的范围之内，而且这个范围的域值较小，即这个范围内值的个数较少)来判断员工的业务能力，这些公司的员工工资应该是保密的，而员工工资的排名顺序应该是各个公司所关注的，这就要用到一定范围内数据的保密排序协议来解决. 现有的保密排序方法都无法跳出两两比较的传统框架，因此泄露了许多不应泄露的信息. 而本文设计的协议 4 跳出两两比较的传统思维，不仅高效地解决了以上问题，而且安全性很高，不存在消息的泄露问题，同时也使可排序的数据范围更加灵活，更加实用.

7.1　安全多方数据排序问题

问题描述. 假设 n 个参与者 $P_1,\cdots,P_n$ 分别拥有私有数据 $a_1,\cdots,a_n\in U$，$U=\{u_1,\cdots,u_m\}$(U 的势较小)，并且集合元素满足 $u_1<\cdots<u_m$，如 $U=\{1,2,\cdots,100\}$. 各个参与者希望得到自己的数据在整个序列中的次序，而不知道其他参与者拥有的数据的任何信息.

方案思想. 参与者拥有的保密数据 $a_1,\cdots,a_n\in U$，$|U|$ 不太大. 我们可以将参与者拥有的保密数据看成是特殊类型的字符，就可以将安全多方数据排序问题转化成多个字符保密排序问题，即通过判断字符的位置关系来确定数据的大小关系.

7.2　具体协议

协议 4.　n 个参与者保密地判断各自拥有的数据在整个序列中的位置排序.

输入：n 个参与者 $P_1,\cdots,P_n$ 分别拥有的保密数据 $a_1,\cdots,a_n$

输出：参与者 $P_1,\cdots,P_n$ 分别得到自己拥有的数据在整个序列中的排列顺序 $W=\{w_1,\cdots,w_n\}$

参与者 $P_i(1\leq i\leq n)$ 调用协议 2(协议 3)，利用协议 2 中提及的编码方法对保密数据进行编码，分别执行协议 1 中的步骤来判断数据 a_i 在整个序列排序的位置. $W=\{w_1,\cdots,w_n\}$ 的值就代表 n 个参与者 $P_1,\cdots,P_n$ 分别拥有的保密数据 $a_1,\cdots,a_n$ 在整个序列中的排序位置.

协议效率分析

在协议 4 中需要 $O(n\log N)$ 次模加运算，参与者之间需要进行 $O(n)$ 轮通信.

对于安全多方数据保密排序问题，我们将本文协议 4 与同类协议在效率、安全性等方面做了如下比较(见表 2). 忽略方案中随机数选择的计算开销和双方准备阶段的计算开销，且将四个方案中的模

都统一为 N 进行比较分析.

表 2 协议 4 性能分析与比较

协议	计算复杂性	通信复杂性	抵抗合谋攻击
文献[26]	$O(n^2\log N)$	$O(n^2)$	否
文献[27]	$O(nm\log N)$	$O(n^2m\log N)$	否
文献[28]	$O(n^2t^2)$	$O(n^2)$	否
文献[37]	$O(nN\log N)$	$O(n^2N\log N)$	是
文献[38]	$O(nN)$	$O(n^2N)$	否
文献[39]	$O(m\log N)$	$O(n^2m\log N)$	否
协议 4	$2n(m+2)$	$O(n)$	是

表 2 中 n 表示参与方数目,t 为门限,m 为共享值的长度.

由表 2 可知,本文协议 4 与其他同类协议相比,通信复杂性较低.另外本文协议 4 和文献[37]都可以抵抗合谋攻击,但是协议 4 的计算复杂性和通信复杂性要更低.在适用范围方面,本文的协议 4 使可排序的数据范围更加灵活,更加实用.

8 结 论

多个字符保密排序问题是新的安全多方计算问题,具有重要的研究意义和应用前景,可以提高云存储数据的查询效率. 本文首先提出了一种新的编码方式,并结合 Paillier 加法同态加密算法、椭圆曲线加密体制、秘密分割和门限解密算法设计了多个字符保密排序的安全多方计算协议,可以抵抗不同程度的合谋攻击.所有协议都跳出了两两比较的传统比较框架,具有更强的保密性.同时将保密的字符排序协议应用于解决安全多方数据排序问题.本文研究的问题都是基于半诚实模型的,对于安全多方计算的研究与应用有重要的理论意义,但恶意模型的安全性更高、更具有实际意义,所以如何实现恶意模型下的字符保密排序问题是我们今后研究的问题.

参 考 文 献

[1] Goldwasser S. Multi party computations: Past and present//Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing. New York, USA, 1997: 1-6

[2] Goldreich O. Secure multi-party computation. Manuscript. Preliminary version. Cambridge, USA: Massachusetts Institute of Technology Press, 1998: 86-97

[3] Freedman M J, Nissim K, Pinkas B. Efficient private matching and set intersection//Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques. Interlaken, Switzerland, 2004: 1-19

[4] Lynn B, Prabhakaran M, Sahai A. Positive results and techniques for obfuscation//Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques. Interlaken, Switzerland, 2004: 20-39

[5] Aggarwal G, Mishra N, Pinkas B. Secure computation of the *k*th-ranked element//Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques. Interlaken, Switzerland, 2004: 40-55

[6] Fitzi M, Holenstein T, Wullschleger J. Multi-party computation with hybrid security//Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques. Interlaken, Switzerland, 2004: 419-438

[7] Ishai Y, Kushilevitz E. On the hardness of information-theoretic multiparty computation//Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques. Interlaken, Switzerland, 2004: 439-455

[8] Golle P, Juels A. Dining cryptographers revisited//Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques. Interlaken, Switzerland, 2004: 456-473

[9] Yao A C. Protocols for secure computations//Proceedings of the 23rd Annual Symposium on Foundations of Computer Science. Chicago, USA, 1982: 160-164

[10] Goldreich O, Micali S, Wigderson A. How to play any mental game//Proceedings of the 19th Annual ACM Symposium on Theory of Computing. New York, USA, 1987: 218-229

[11] Yasin S, Haseeb K, Qureshi R J. Cryptography based e-commerce security: A review. International Journal of Computer Science Issues, 2012, 9(2): 132-137

[12] Sharma R. Review paper on cryptography. International Journal of Research, 2015, 2(5): 141-142

[13] Kumar S N. Review on network security and cryptography. International Transaction of Electrical and Computer Engineers System, 2015, 3(1): 1-11

[14] Lin H Y, Tzeng W G. An efficient solution to the millionaires' problem based on homomorphic encryption//Proceedings of the International Conference on Applied Cryptography and Network Security. New York, USA, 2005: 456-466

[15] Li Shun-Dong, Wang Dao-Shun. Efficient secure multiparty computation based on homomorphic encryption. Acta Electronica Sinica, 2013, 41(4): 798-803

[16] Sheikh R, Mishra D K, Kumar B. Secure multiparty computation: From millionaires problem to anonymizer. Information Security Journal: A Global Perspective, 2011, 20(1): 25-33

[17] Grigoriev D, Shpilrain V. Yao's millionaires' problem and decoy-based public key encryption by classical physics. International Journal of Foundations of Computer Science, 2014, 25(4): 409-417

[18] Boudot F, Schoenmakers B, Traore J. A fair and efficient solution to the socialist millionaires' problem. Discrete Applied Mathematics, 2001, 111(1): 23-36

[19] Okamoto T. Provably secure and practical identification schemes and corresponding signature schemes//Proceedings of the 12th Annual International Cryptology Conference. California, USA, 1992: 31-53

[20] Atallah M J, Du W. Secure multi-party computational geometry//Proceedings of the Workshop on Algorithms and Data Structures. New York, USA, 2001: 165-179

[21] Du W, Atallah M J. Secure multi-party computation problems and their applications: A review and open problems//Proceedings of the 2001 Workshop on New Security Paradigms. New York, USA, 2001: 13-22

[22] Li Shun-Dong, Wu Chun-Ying, Wang Dao-Shun, et al. Secure multiparty computation of solid geometric problems and their applications. Information Sciences, 2014, 282(10): 401-413

[23] Li Shun-Dong, Dai Yi-Qi, Wang Dao-Shun, et al. A secure multi-party computation solution to intersection problems of sets and rectangles. Progress in Natural Science, 2006, 16(5): 538-545

[24] Lindell Y, Pinkas B. Privacy preserving data mining. Journal of Cryptology, 2002, 15(3): 177-206

[25] Xiao Qian, Luo Shou-Shan, Chen Ping, et al. Research on the problem of secure multi-party ranking under semi-honest model. Acta Electronica Sinica, 2008, 36(4): 709-714(in Chinese)
(肖倩，罗守山，陈萍等. 半诚实模型下安全多方排序问题的研究. 电子学报，2008, 36(4): 709-714)

[26] Li Shun-Dong, Zhang Xuan-Ping. Secure multi-party ranking problem. Journal of Xi'an Jiaotong University, 2008, 42(2): 231-233(in Chinese)
(李顺东，张选平. 排序问题的多方保密计算. 西安交通大学学报，2008, 42(2): 231-233)

[27] Liu Wen, Luo Shou-Shan, Chen Ping. A study of secure multi-party ranking problem//Proceedings of the Eighth ACIS International Conference. Qingdao, China, 2007, 2: 727-732

[28] Tang Chun-Ming, Shi Gui-Hua, Yao Zheng-An. Secure multi-party computation protocol for sequencing problem. Science China Information Sciences, 2011, 41(7): 789-797 (in Chinese)
(唐春明，石桂花，姚正安. 排序问题的安全多方计算协议. 中国科学：信息科学，2011,41(7): 789-797)

[29] Goldreich O. The Fundamental of Cryptography: Basic Applications. London, UK: Cambridge University Press, 2004: 599-764

[30] Rivest R L, Adleman L, Dertouzos M L. On data banks and privacy homomorphisms. Foundations of Secure Computation, 1978, 4(11): 169-180

[31] Paillier P. Public-key cryptosystems based on composite degree residuosity classes//Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques. Prague, Czech Republic, 1999: 223-238

[32] Yang Bo. Foundations of Modern Cryptography. Beijing: Tsinghua University Press, 2015(in Chinese)
(杨波. 现代密码学基础. 北京：清华大学出版社，2015)

[33] Bh P, Chandravathi D, Roja P P. Encoding and decoding of a message in the implementation of Elliptic Curve cryptography using Koblitz's method. International Journal on Computer Science and Engineering, 2010, 2(5): 1904-1907

[34] Desmedt Y, Frankel Y. Threshold cryptosystems//Proceedings of the International Workshop on the Theory and Application of Cryptographic Techniques. Gold Goast, Australia, 1990: 307-315

[35] Long Y, Chen K, Mao X. New constructions of dynamic threshold cryptosystem. Journal of Shanghai Jiaotong University(Science), 2014, 19: 431-435

[36] Samanthula B K, Jiang W. Secure multiset intersection cardinality and its application to Jaccard coefficient. IEEE Transactions on Dependable and Secure Computing, 2016, 13(5): 591-604

[37] Liu Wen, Luo Shou-Shan, Chen Ping. A study of secure multi-party ranking problem based on ElGamal encryption algorithm. Journal on Communications, 2007, 28(11): 1-5 (in Chinese)
(刘文，罗守山，陈萍. 利用 ElGamal 密码体制解决安全多方多数据排序问题. 通信学报，2007, 28(11): 1-5)

[38] Qiu Mei, Luo Shou-Shan, Liu Wen, Chen Ping. A solution of secure multi-party multi-data ranking problem based on RSA encryption scheme. Acta Electronica Sinica, 2009, 37(5): 1119-1123(in Chinese)
(邱梅，罗守山，刘文，陈萍. 利用 RSA 密码体制解决安全多方多数据排序问题. 电子学报，2009, 37(5): 1119-1123)

[39] Damgård I, Fitzi M, Kiltz E, et al. Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation//Proceedings of the Third Theory of Cryptography Conference. New York, USA, 2006, 3876: 285-304

附录 1.

协议 1 安全性证明.

在协议 1 中，$P_i(1\leq i\leq n-1)$将 q_i 发送给 P_1 解密. 因为 P_1 拥有私钥，只有他可以解密，所以协议 1 可以抵抗除 P_1 之外的合谋攻击. 以下是关于协议 1 安全性的具体分析：

假设所有参与者为集合 $P=\{P_1,\cdots,P_n\}$，合谋参与者集合为 $I=\{P_{i1},\cdots,P_{i\alpha}\}$.

(1) 共谋 1. P_1 不参与合谋，I 想要得到参与者 $P_i\in P\backslash I$ 的字符 s_i.

s 为所有参与者拥有字符的集合且 $s=(s_1,\cdots,s_n)$. 令 $s_I=(s_2,\cdots,s_{i-1},s_{i+1},\cdots,s_n)$，通过构造使下式成立的概率多项式时间模拟器 S 来证明上述情况的安全性：

$$\{S(I,s_I,f_I(s))\}_{s\in(\{0,1\}^*)^n}\overset{c}{\equiv}\{view_I^{\pi}(s)\}_{s\in(\{0,1\}^*)^n}.$$

S 的模拟过程如下：

① 给定输入 $(I,s_I,f_I(s))$，即 $(I,s_I,f_I(s))=(I,(s_2,\cdots,$

$s_{i-1},s_{i+1},\cdots,s_n),f_I(s))$,$S$ 随机选择两个字符s_1'和 s_i',使得 $f_I(s)=f_I(s')$,在这里 $s'=(s_1',s_2,\cdots,s_{i-1},s_i',s_{i+1},\cdots,s_n)$.

② 模拟器 S 将 s' 中的每个字符 s_i' 编码为序列 $\boldsymbol{T}_1^*,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_i^*,\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n$. 然后模拟器 S 利用公钥加密 $\boldsymbol{T}_1^*,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_i^*,\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n$ 得到密文向量

$$E(\boldsymbol{T}_1^*),\cdots,E(\boldsymbol{T}_{i-1}),E(\boldsymbol{T}_i^*),E(\boldsymbol{T}_{i+1}),\cdots,E(\boldsymbol{T}_n).$$

③ 模拟器 S 按照协议 1 将密文向量中对应分量相乘,得到 $E(\boldsymbol{T}_n')$,选取随机数 r_i',并用公钥加密得到 $E(r_i')$. 然后分别按照各自字符的位置$(s_i')_{\text{ord}}$,计算 $E(\boldsymbol{T}_n')$ 中前$(s_i')_{\text{ord}}$个分量与 $E(r_i')$ 累乘的结果,记为 q_i',并将 q_i' 发送给 P_1.

④ 因为 $P_1\notin I$,S 没有 Paillier 加密算法的私钥,不能对密文解密.

在协议 1 中:

$$\begin{aligned}view_1^{\pi}(s)&=\{view_2^{\pi}(s),\cdots,view_{i-1}^{\pi}(s),\\&\quad view_{i+1}^{\pi}(s),\cdots,view_n^{\pi}(s)\}\\&=\{(s_2,\cdots,s_{i-1},s_{i+1},\cdots,s_n),\\&\quad(\boldsymbol{T}_2,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n),\\&\quad(E(\boldsymbol{T}_2),\cdots,E(\boldsymbol{T}_{i-1}),E(\boldsymbol{T}_{i+1}),\cdots,E(\boldsymbol{T}_n)),\\&\quad E(\boldsymbol{T}_n),D(Q),(w_2,\cdots,w_{i-1},w_{i+1},\cdots,w_n)\},\\S(I,(s_2,\cdots,s_n),f_I(s))&=\{I,(s_2,\cdots,s_{i-1},s_{i+1},\cdots,s_n),\\&\quad(\boldsymbol{T}_2,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n),\\&\quad(E(\boldsymbol{T}_2),\cdots,E(\boldsymbol{T}_{i-1}),E(\boldsymbol{T}_{i+1}),\cdots,E(\boldsymbol{T}_n)),\\&\quad(E(\boldsymbol{T}_2^*),\cdots,E(\boldsymbol{T}_{i-1}^*),E(\boldsymbol{T}_{i+1}^*),\cdots,E(\boldsymbol{T}_n^*)),\\&\quad E(\boldsymbol{T}_n'),D(Q'),(w_2',\cdots,w_{i-1}',w_{i+1}',\cdots,w_n')\}.\end{aligned}$$

因为$(w_2,\cdots,w_{i-1},w_{i+1},\cdots,w_n)=(w_2',\cdots,w_{i-1}',w_{i+1}',\cdots,w_n')$,而且 Paillier 加密算法加密后的数据是计算不可区分的,所以上述模拟过程得到的消息序列与实际执行过程中得到的消息序列是计算不可区分的,即

$$\{S(I,s_I,f_I(s))\}_{s\in(\{0,1\}^*)^n}\overset{c}{\equiv}\{view_I^{\pi}(s)\}_{s\in(\{0,1\}^*)^n}.$$

由此可知:在共谋 1 情形下基于秘密分割的多个字符保密排序协议 1 是安全的.

(2) 共谋 2:P_1 参与合谋,I 合谋想要得到参与者 $P_i\in P\backslash I$ 的字符 s_i.

当 $I=P\backslash\{P_i\}$,即 I 为除 P_i 之外的所有参与者时,在协议 1 执行完之后,除 P_i 之外的所有参与者合作可以得到 $D(q_i)$ 的值,继而知道关于 s_i 的信息.

当 $I\neq P\backslash\{P_i\}$,S 的模拟过程如下:

① 给定输入$(I,s_I,f_I(s))$,即$(I,s_I,f_I(s))=(I,(s_1,\cdots,s_{i-1},s_{i+1},\cdots,s_n),f_I(s))$,$S$ 随机选择一个字符s_i',使得 $f_I(s)=f_I(s')$,在这里 $s'=(s_1,s_2,\cdots,s_{i-1},s_i',s_{i+1},\cdots,s_n)$.

② 模拟器 S 将 s' 中的每个字符 s_i' 编码为向量 $\boldsymbol{T}_1,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_i^*,\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n$. 然后模拟器 S 利用公钥加密 $\boldsymbol{T}_1,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_i^*,\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n$ 得到密文向量

$$E(\boldsymbol{T}_1),\cdots,E(\boldsymbol{T}_{i-1}),E(\boldsymbol{T}_i^*),E(\boldsymbol{T}_{i+1}),\cdots,E(\boldsymbol{T}_n).$$

③ 模拟器 S 按照协议 1 将密文向量中对应分量相乘,得到 $E(\boldsymbol{T}_n')$,选取随机数 r_i',并用公钥加密得到 $E(r_i')$. 然后分别按照各自字符的位置$(s_i')_{\text{ord}}$,计算 $E(\boldsymbol{T}_n')$ 中前$(s_i')_{\text{ord}}$个分量与 $E(r_i')$ 累乘的结果,记为 q_i',并将 q_i' 发送给 P_1.

在协议 1 中:

$$\begin{aligned}view_I^{\pi}(s)&=\{view_2^{\pi}(s),\cdots,view_{i-1}^{\pi}(s),\\&\quad view_{i+1}^{\pi}(s),\cdots,view_n^{\pi}(s)\}\\&=\{(s_2,\cdots,s_{i-1},s_{i+1},\cdots,s_n),\\&\quad(\boldsymbol{T}_2,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n),\\&\quad(E(\boldsymbol{T}_2),\cdots,E(\boldsymbol{T}_{i-1}),E(\boldsymbol{T}_{i+1}),\cdots,E(\boldsymbol{T}_n)),\\&\quad E(\boldsymbol{T}_n),D(Q),(w_2,\cdots,w_{i-1},w_{i+1},\cdots,w_n)\},\\S(I,(s_2,\cdots,s_n),f_I(s))&=\{I,(s_2,\cdots,s_{i-1},s_{i+1},\cdots,s_n),\\&\quad(\boldsymbol{T}_2,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n),\\&\quad(E(\boldsymbol{T}_2),\cdots,E(\boldsymbol{T}_{i-1}),E(\boldsymbol{T}_{i+1}),\cdots,E(\boldsymbol{T}_n)),\\&\quad(E(\boldsymbol{T}_2^*),\cdots,E(\boldsymbol{T}_{i-1}^*),E(\boldsymbol{T}_{i+1}^*),\cdots,E(\boldsymbol{T}_n^*)),\\&\quad E(\boldsymbol{T}_n'),D(Q'),(w_2',\cdots,w_{i-1}',w_{i+1}',\cdots,w_n')\}.\end{aligned}$$

因为$(w_2,\cdots,w_{i-1},w_{i+1},\cdots,w_n)=(w_2',\cdots,w_{i-1}',w_{i+1}',\cdots,w_n')$,而且 Paillier 加密算法加密后的数据是计算不可区分的,所以上述模拟过程得到的消息序列与实际执行过程中得到的消息序列是计算不可区分的,即

$$\{S(I,s_I,f_I(s))\}_{s\in(\{0,1\}^*)^n}\overset{c}{\equiv}\{view_I^{\pi}(s)\}_{s\in(\{0,1\}^*)^n}.$$

由此可知:协议 1 不能抵抗除 P_i 之外的所有参与者参与的共谋攻击.

协议 3 安全性证明.

证明思路. 证明协议的安全性需要构造满足式(1)的模拟器 S. 根据语义安全的同态加密算法的性质,如果没有私钥,应用公钥加密的任何信息都是计算不可区分的,因此只要有一个参与者不合谋,对其他合谋者来说,他们实际执行时获得的 $view$ 和用满足所有字符排序不变的任意一组输入进行模拟所得到的信息序列是计算不可区分的,所以只要在式(1)中令 $S(I,(s_{i1},\cdots,s_{ia}),f_I(s))$ 为模拟过程中的 $view$,即可使式(1)满足.

安全性分析关于协议 3 的安全性,有如下定理.

定理 2. 在半诚实模型下,基于门限密码系统的多个字符保密排序协议 3 是安全的.

证明. 协议 3 的安全性是基于椭圆曲线加密体制的安全性. 由于门限椭圆曲线的公钥是由所有参与者共同产生的,即 $H=\sum_{i=1}^{n}h_iG$,其中 h_i 是参与者 P_i 所持有的私钥碎片,如果想解密的话,就必须拥有全部参与者的私钥碎片. 所以整个解密的过程都必须要全部参与者参与,因而可以抵抗合谋攻击.

在计算过程中,每个参与者 P_i 对外仅公布了加密信息

$$\begin{aligned}E(M_i)=\{&\langle M_{i1}+r_{i1}H,r_{i1}G\rangle,\langle M_{i2}+r_{i2}H,r_{i2}G\rangle,\cdots,\\&\langle M_{ij}+r_{ij}H,r_{ij}G\rangle,\cdots,\langle M_{im}+r_{im}H,r_{im}G\rangle\}.\end{aligned}$$

在解密过程中对外也仅公布了加密信息 C_{i2},由椭圆曲线密码体制的安全性可知,在协议执行过程中如果 P_i 没有参与,将无法解密得到 z_i,也无法得到 w_i. 因此在协议执行过程中,P_i 的字符 s_i 是完全保密的.

以下是关于协议 3 安全性的具体分析:

假设所有参与者满足 $P=\{P_1,\cdots,P_n\}$,合谋参与者集合为 $I=\{P_{i1},\cdots,P_{ia}\}$.

(1)共谋 1. $I=P\backslash\{P_i\}$(即除 P_i 外的所有参与者)合谋想要得到参与者 $P_i\in P\backslash I$ 的字符 s_i.

s 为所有参与者拥有字符的集合且 $s=(s_1,s_2,\cdots,s_n)$. 令 $s_I=(s_1,\cdots,s_{i-1},s_{i+1},\cdots,s_n)$,通过构造使下式成立的概率多项式时间模拟器 S 来证明上述情况的安全性:

$$\{S(I,s_I,f_I(s))\}_{s\in(\{0,1\}^*)^n}\overset{c}{\equiv}\{view_I^{\pi}(s)\}_{s\in(\{0,1\}^*)^n}.$$

S 的模拟过程如下:

① 给定输入 $(I,s_I,f_I(s))$,即 $(I,s_I,f_I(s))=(I,(s_1,\cdots,s_{i-1},s_{i+1},\cdots,s_n),f_I(s))$,$S$ 随机选择字符 s_i',使得 $f_I(s)=f_I(s')$,在这里 $s'=(s_1,s_2,\cdots,s_{i-1},s_i',s_{i+1},\cdots,s_n)$.

② 模拟器 S 将 s' 中的每个字符 s_i' 编码为向量 $\boldsymbol{T}_1,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_i^*,\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n$. 并将向量 $\boldsymbol{T}_1,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_i^*,\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n$ 编码到 $EC(a,b)$ 上作为椭圆曲线上的点 $M_1,\cdots,M_{i-1},M_i^*,M_{i+1},\cdots,M_n$. 然后模拟器 S 利用公钥加密 $M_1,\cdots,M_{i-1},M_i^*,M_{i+1},\cdots,M_n$ 得到密文序列 $E(M_1),\cdots,E(M_{i-1}),E(M_i^*),E(M_{i+1}),\cdots,E(M_n)$

③ 模拟器 S 按照协议 3 得到矩阵 $E(\boldsymbol{M}')$,选取随机数 R_i',并用公钥加密得到 $E(R_i')$. 然后分别按照各自字符的位置 $(s_i')_{\mathrm{ord}}$,计算 $E(\boldsymbol{T}')$ 中前 $(s_i')_{\mathrm{ord}}$ 个分量与 $E(R_i')$ 累加的结果,记为 $\langle C_{i1}',C_{i2}'\rangle$,并计算 z_i'.

④ 模拟器 S 宣布 Z'.

因为 $I=\{P_1,\cdots,P_{i-1},P_{i+1},\cdots,P_n\}$,在解密时,无法拥有全部参与者的私钥碎片,不能解密.

在协议 3 中:

$$\begin{aligned}view_I^{\pi}(s)=&\{view_2^{\pi}(s),\cdots,view_{i-1}^{\pi}(s),\\&view_{i+1}^{\pi}(s),\cdots,view_n^{\pi}(s)\}\\=&\{(s_1,\cdots,s_{i-1},s_{i+1},\cdots,s_n),\\&(\boldsymbol{T}_1,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n),\\&(M_1,\cdots,M_{i-1},M_{i+1},\cdots,M_n),\\&(E(M_1),\cdots,E(M_{i-1}),E(M_{i+1}),\cdots,E(M_n)),\\&E(M),Z,(w_1,\cdots,w_{i-1},w_{i+1},\cdots,w_n)\},\end{aligned}$$

$$\begin{aligned}S(I,(s_1,\cdots,s_n),f_I(s))=&\{I,(s_1,\cdots,s_{i-1},s_{i+1},\cdots,s_n),\\&(\boldsymbol{T}_1,\cdots,\boldsymbol{T}_{i-1},\boldsymbol{T}_{i+1},\cdots,\boldsymbol{T}_n),\\&(M_1,\cdots,M_{i-1},M_{i+1},\cdots,M_n),\\&(E(M_1),\cdots,E(M_{i-1}),E(M_{i+1}),\cdots,E(M_n)),\\&E(\boldsymbol{M}'),Z',(w_1',\cdots,w_{i-1}',w_{i+1}',\cdots,w_n')\}.\end{aligned}$$

因为 $(w_1,\cdots,w_{i-1},w_{i+1},\cdots,w_n)=(w_1',\cdots,w_{i-1}',w_{i+1}',\cdots,w_n')$,而且椭圆曲线加密算法加密后的数据是计算不可区分的,所以上述模拟过程得到的消息序列与实际执行过程中得到的消息序列是计算不可区分的,即

$$\{S(I,s_I,f_I(s))\}_{s\in(\{0,1\}^*)^n}\overset{c}{\equiv}\{view_I^{\pi}(s)\}_{s\in(\{0,1\}^*)^n}.$$

由此可知:在共谋 1 情形下,基于门限密码系统的多个字符保密排序协议 3 是安全的.

(2)共谋 2. $I\neq P\backslash\{P_i\}$ 合谋想要得到参与者 $P_i\in P\backslash I$ 的字符 s_i.

由上述共谋 1 情形可知,当 I 为除 P_i 的参与者时,因为在解密时不能构造完整的私钥,所以得不到任何关于 s_i 的信息. 那么当 $I\neq\{P_1,\cdots,P_{i-1},P_{i+1},\cdots,P_n\}\subseteq P$ 因为缺少部分参与者的私钥碎片,所以不能解密. 因此在共谋 1 情形下,基于门限密码系统的多个字符保密排序协议 3 是安全的.

综上所述,协议 3 可以抵抗任意形式的合谋攻击. 即在半诚实模型下,基于门限密码系统的多个字符保密排序协议 3 是安全的.

协议 4 安全性证明.

安全性分析. 关于协议 4 的安全性,有如下定理.

定理 3. 在半诚实模型下,安全多方数据保密排序协议 4 是安全的.

证明. 因为协议 4 是调用协议 2(协议 3)的,所以协议 4 的安全性证明和协议 2(协议 3)类似. 在此不做详细证明.

LI Shun-Dong, born in 1963, Ph.D., professor, Ph. D. supervisor. His main research interests include modern cryptography and information security.

KANG Jia, born in 1992, M. S. candidate. Her main research interests include modern cryptography and information security.

YANG Xiao-Yi, born in 1993, Ph. D. candidate. Her main research interests include modern cryptography and information security.

DOU Jia-Wei, born in 1963, Ph. D., associate professor. Her main research interests include computational mathematics and applied mathematics.

LIU Xin, born in 1983, Ph. D., lecturer. Her main research interests include modern cryptography and information security.

Background

Secure multiparty computation (SMC) plays an important role in information security, and is a pivotal privacy preserving technology both in cyberspaces and in cooperative computation. It is a research focus in the international cryptographic

community in recent years. Many cryptographic scholars have explored various secure multiparty computation problems arising in various fields and proposed their solutions. However, there are many difficult problems needed to be further studied in the future.

In this study, we studied a completely new SMC problem, that is, privately sort characters lexicographically, which has important practical significance and broad application prospects in information security. This problem can further be used to solve the secure multi-party ranking problem to extend the range of numbers to be compared. Although there are protocols for privacy-preserving data sorting, direct using them to solve this problem is impractical and inefficient, and the current research on this problem is still in the initial stage. To the best of our knowledge, this problem of privately sorting multiple characters lexicographically has not been investigated.

To solve this problem, we first encode private numbers and convert them into a matrix through a new encoding scheme, and then we design 3 simple and efficient protocols to privately determine the lexicographical order of private characters, which are based on the additively homomorphic ECC probabilistic encryption algorithm and the Paillier probabilistic encryption algorithm and use secret cutting technology and threshold decryption.

We further use these schemes as building blocks to solve the secure multi-party number sorting problem to extend the range of numbers to be compared. Furthermore, we proved that these protocols are secure in the semi-honest model, and analyzed their correctness. Analysis of the computational complexities and communication complexities indicates that the efficiency of the above protocol is very high.

The study is supported by the National Natural Science Foundation of China under Grant No. 61272435, the Inner Mongolia Natural Science Foundation Project No. 2017MS0602, and the Inner Mongolia Autonomous Region University Scientific Research Project No. NJZY17164. The purpose of this project is to solve all kinds of confidential problems in field of cryptography. Our team has been devoted to exploring and analyzing cryptographic protocols for over 10 years, such as SMC, SMC geometry, 1-out-of-m oblivious transfer, zero knowledge proof. We have published over 60 papers, of which over 30 have been indexed by SCI.