安全多方向量计算

周素芳[1] 窦家维[2] 郭奕旻[1),3)] 毛 庆[1] 李顺东[1]

[1)](陕西师范大学计算机科学学院 西安 710062)

[2)](陕西师范大学数学与信息科学学院 西安 710062)

[3)](中国科学院软件研究所可信计算与信息保障实验室 北京 100190)

摘 要 安全多方计算是密码学一个重要研究方向,是国际密码学界的热点.文中研究向量问题的安全多方计算.一个向量通常由多个分量组成,每个分量可以表示不同的物理意义,因此对向量的计算,相当于同时对具有不同物理意义的分量分别计算.对向量进行高效保密计算,具有重要的理论与实际意义,因此安全多方向量计算成为安全多方计算的一个重要问题.但是该问题现在还没有直接的解决方案,现有的相关方案都是一些朴素的解决方案,即利用加法同态加密算法对向量的每个分量分别加密,然后计算所有向量分量的和,进而实现向量的计算,其效率比较低.文中利用哥德尔编码将向量和自然数一一对应,并借助语义安全乘法同态加密算法设计了一个可以直接对向量进行计算的高效保密计算方案.文中进一步将向量与多项式对应,利用 NTRU 加密算法设计了一种可能抵抗量子攻击的高效向量计算方案.使用安全多方计算普遍采用的模拟范例证明方法证明了这些方案在半诚实模型下是安全的.作为方案的应用,文中提出了高效的安全统计方案和高效的安全电子选举方案.

关键词 密码学;安全多方计算;向量计算;安全统计;安全电子选举

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2017.01134

Secure Multiparty Vector Computation

ZHOU Su-Fang[1] DOU Jia-Wei[2] GUO Yi-Min[1),3)] MAO Qing[1] LI Shun-Dong[1]

[1)](*School of Computer Science, Shaanxi Normal University, Xi'an* 710062)

[2)](*School of Mathematics and Information Science, Shaanxi Normal University, Xi'an* 710062)

[3)](*Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing* 100190)

Abstract Secure multiparty computation is an important field of cryptography a focus of international cryptographic community. This paper studies secure multiparty vector computation. A vector is often composed of multiple components, and each component may has different practical meaning. Therefore vector computation is equivalent to compute the components with different physical meanings at the same time. Privately and efficiently performing vector computation is of theoretical and practical significance. Therefore, secure multiparty vector computation is an important problem of secure multiparty computation. However, to the best of our knowledge, there is no direct and efficient solution to this problem. Existing protocols are trivial. These protocols implement the vectors computation using additively homomorphic encryption scheme to encrypt each component of vectors, and then to privately add the corresponding components of vectors. Such protocols are inefficient. In this study, we use Gödel number to encode a vector into a natural number, and a semantically secure and multiplicatively homomorphic encryption scheme, and then design an efficient scheme enabling direct computation of linear combinations of

收稿日期:2015-05-05;在线出版日期:2016-04-10.本课题得到国家自然科学基金面上项目(61272435)和中央高校基本科研业务费专项资金(2016TS061)资助.**周素芳**,女,1990 年生,博士研究生,主要研究方向为信息安全与密码学.E-mail:zhousufang@snnu.edu.cn.**窦家维**(通信作者),女,1963 年生,博士,副教授,主要研究方向为应用数学与应用密码学.E-mail:jiawei@snnu.edu.cn.**郭奕旻**,女,1992 年生,博士研究生,主要研究方向为信息安全与密码学.**毛 庆**,男,1973 年生,博士研究生,讲师,主要研究方向为安全多方计算.**李顺东**,男,1963 年生,博士,教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为密码学与信息安全.

vectors privately. In order to make secure multiparty vector computation scheme secure against quantum attack, we further devise an efficient protocol based on the NTRU cryptosystem, and mapping a vector to a polynomial. These schemes are proven to be secure in the semi-honest model using the simulation paradigm which is widely used in secure multiparty computation. As the applications of these protocols, we demonstrate how to use them to perform secure statistics and secure electronic elections.

Keywords　cryptography; secure multi-party computation; vector computation; secure statistic; secure electronic election

1 引　言

网络的迅速发展为多个参与者的合作计算提供了巨大的机会,同时也给参与者的信息安全带来了巨大的挑战.由于网络环境的复杂性,参与者之间互不信任,他们需要保护各自消息的隐私性,这使得安全多方计算(Secure Multiparty Computation,SMC)越来越受到人们的关注.安全多方计算是指两个或多个参与者联合进行的秘密计算,计算结束后,各个参与者除了得到既定的输出结果外,输入信息没有任何泄露.很多密码学问题如秘密共享、密钥协商、零知识证明、不经意传输、盲签名等都需要不同参与者之间进行交互计算,因此都可以看作是安全多方计算的一个特例.Goldwasser[1]预言安全多方计算将成为计算科学中一个必不可少的组成部分.

1982 年 Yao[2]提出了两个参与者的安全多方计算.1988 年 Ben-Or 和 Goldwasser 等人[3]给出了多个参与者的安全多方计算.随后 Goldreich 等人[4-5]从理论上证明了任意的安全多方计算问题都是可解的,并给出了通用的解决方案,但指出从计算效率方面考虑对具体的问题应该研究具体的解决方案.他们还利用比特承诺[6]和零知识证明[7]设计了一个编译器,给定该编译器一个对于半诚实参与者安全的安全多方计算协议可以生成一个对于恶意参与者也安全的安全多方计算协议,这表明研究基于半诚实模型下的安全多方计算问题具有重要的价值,Goldreich 还在文献[5]中专门论述了研究半诚实模型下安全多方计算的理论与实际意义.Goldreich 和 Goldwasser 等人的研究激励着人们研究各种各样的安全多方计算问题,这些问题有百万富翁问题[8-9]、保密的计算几何[10]、保密的信息比较[11]、保密的集合问题[12]、保密的远程访问[13]、保密拍卖[14]、保密的数据挖掘[15]等.但由于安全多方计算研究范围非常广泛,目前还有很多安全多方计算问题有待进一步研究.

一个向量中往往含有多个分量,不同的分量代表不同的含义,对向量计算相当于对其内的多个分量分别做相应的计算,由于向量计算的这种特殊性质,使其在很多方面都有重要的应用,同时向量计算问题也是安全多方计算中的一个重要问题.向量计算在保密选举[16]、保密统计[17]、电子投票[18]、统计分布[19]、隐私保护的数据挖掘[20]等问题中有着重要的应用.关于向量计算问题已有的研究主要集中在保密的向量求和、保密的数据求和与向量内积的计算,但他们都只是安全多方向量计算的特例.

文献[21-22]中的方案是对只有一个分量的多个向量进行安全求和,即安全的数据求和[23].文献[21]利用拆分的方法给出了安全数据求和的解决方案,但方案中的拆分方法需要借助于哈密尔顿回路.为了抵抗 $2k-1$ 个参与者的合谋攻击,需要参与者之间的数据形成 k 个特殊的哈密尔顿回路 EDHC(任意两个回路都不能有公共边).而哈密尔顿回路问题是一个 NPC 问题,找到一个哈密尔顿回路是困难的,找到 k 个没有相交边的哈密尔顿回路问题比哈密尔顿回路问题更难,因此文献[21]的方法消耗的工作量很大,并不实用.文献[22]使用加入随机数的方法解决数据安全求和问题,但该方案中如果参与者 P_{j-1} 与参与者 P_{j+1} 合谋,可以恢复出参与者 P_j 拥有的数据,不能抵抗合谋攻击,因此仅适用于对数据安全性要求不高的场合.

文献[18-19,24-25]中的方案虽可以看作是向量安全求和,但它们没有实现真正意义上的向量安全求和,需要对向量的各个分量分别应用协议,如果是 n 维向量则需要调用协议 n 次,计算复杂性与通信复杂性都较高.

文献[25]借助 Paillier 公钥加密算法和可信的第三方给出了向量安全求和方案.该方案需要用 Paillier 公钥加密算法加密向量中的每个分量,并根据 Paillier 算法的加法同态性质计算出所有向量和

的密文,然后对向量和密文中的每个分量分别解密,从而得到所有向量的和.为了抵抗参与者的合谋攻击,需要可信的第三方将解密密钥在多个权威中心之间共享,并由多个权威中心联合对向量和密文中的每个分量解密,来实现向量和的安全计算.使用Paillier加密算法较难实现门限的加密算法,每次解密需要计算 $2k$(k 是门限值)次指数运算,因此方案的效率较低.同时,在复杂的网络环境中,寻找一个可信的第三方是非常困难的;可信的第三方也是需要成本的;如果很多协议都借助于某个可信的第三方,这个第三方可能成为攻击者攻击的主要目标,也会形成协议的瓶颈.因此借助于可信第三方来抵抗合谋攻击的实用性较差.

文献[18,24]中的解决方案借助的是改进的ElGamal[26]公钥加密算法(将要加密的 M 作为指数,原本算法中加密的消息 M 表示为 T^M),但该方法解密后得到的是 T^M.由于离散对数困难性问题,没有高效计算 M 的方法,只能通过在 M 取值范围内穷举或借助 Pollard 的方法进行计算,效率较低,而且需要对向量和密文中的每个分量解密.为了抵抗参与者之间的合谋攻击,需要多个参与者共同生成密钥并进行联合解密.

在文献[19]的方案中,利用公钥加密算法解决了多组数据求和问题.每个参与者都需要对上一个参与者发送过来的密文集合内的所有元素逐个解密,解密后加入自己的秘密值并用下一个参与者的公钥加密,然后将新构成的密文集合发送给下一个参与者,如此重复直到形成一个环,从而求出所有参与者的数组和,方案中加密解密的次数较多.文献[27-28]主要研究的是向量内积的计算.

本文研究的是向量线性组合计算.借助哥德尔编码将一个向量与一个自然数建立一一对应的关系,各个参与者分别将拥有的向量编码成一个对应的自然数,然后利用乘法同态性质加密算法实现加法性质的向量保密计算,降低了向量计算的计算复杂性和通信复杂性.本文又借助 NTRU[29]公钥加密算法,提出了高效且可能抵抗量子攻击的方案.为了抵抗合谋问题,本文提出了密文拆分方法,即将参与者数乘向量的密文分成任意需要的 k 份并发送给 m 个参与者中的 k 个.根据本文的密文拆分方法,不需要将密文直接进行因子分解,且可以拆分成任意的份额,同时只需要做简单的模乘运算,效率较高.复数的线性组合计算一直是密码学多方保密计算领域中的困难性问题,根据向量的计算可以计算若干个复数的线性组合,只需要将复数看作一个二维的向量.同时本文给出的方案实现了真正意义上的安全多方向量计算.作为向量计算方案的应用,本文给出了高效且安全的统计方案和选举方案.

本文贡献如下:

(1)设计了一种新的、高效的抵抗合谋攻击的密文拆分方法.

(2)借助于哥德尔编码和密文拆分方法,用语义安全乘法同态加密算法实现了向量线性组合的安全多方计算,并利用模拟范例证明了方案的安全性.向量线性组合的安全多方计算问题目前还没有其他解决方案.

(3)利用 NTRU 公钥加密算法,给出了可能抵抗量子攻击的向量线性组合安全多方计算方案.

(4)将向量安全多方计算方案应用于实际问题,解决了可能抵抗量子攻击的多项数据的安全统计和 n 选 k 的安全选举这两个实际应用问题.

2 预备知识

2.1 向量计算问题

设 m 个参与者 $P_1,\cdots,P_m$ 分别拥有向量 $\boldsymbol{X}_1=(x_{11},\cdots,x_{1n}),\cdots,\boldsymbol{X}_m=(x_{m1},\cdots,x_{mn})$,他们想要共同计算

$$\boldsymbol{X}=a_1\boldsymbol{X}_1+\cdots+a_m\boldsymbol{X}_m=(x_1,\cdots,x_n),$$

但不想泄露各自拥有向量的任何信息,这就是本文研究的安全多方向量计算问题.

2.2 理想模型

假设有一个可信的第三方(Trusted Third Party,TTP),他在任何情况下都不会撒谎,也绝不会泄露任何不该泄露的信息.理想的安全多方计算即为多个参与者借助于可信的第三方进行的安全计算.若有 m 个参与者 $P_1,\cdots,P_m$,他们分别将各自的秘密消息 $\boldsymbol{X}_1,\cdots,\boldsymbol{X}_m$ 发送给可信的第三方,由可信的第三方自己单独计算函数 f:

$$f(\boldsymbol{X}_1,\cdots,\boldsymbol{X}_m)=(f_1(\boldsymbol{X}_1,\cdots,\boldsymbol{X}_m),\cdots,f_m(\boldsymbol{X}_1,\cdots,\boldsymbol{X}_m)),$$

然后将计算结果

$$f_1(\boldsymbol{X}_1,\cdots,\boldsymbol{X}_m),\cdots,f_m(\boldsymbol{X}_1,\cdots,\boldsymbol{X}_m)$$

分别发送给参与者 $P_1,\cdots,P_m$.因此 $P_1,\cdots,P_m$ 除了从协议中得到可信第三方发送给自己的计算结果外得不到任何其他信息.该协议是一个理想的安全多方计算协议.理想的安全多方计算协议虽然简单,但其安全性却是最高的安全多方计算协议,任何一个函数 $f(\boldsymbol{X}_1,\cdots,\boldsymbol{X}_m)$ 的实际安全多方计算协议的安

全性都不可能超过这个协议，因此实际安全计算协议可以通过和对应的理想安全多方计算协议比较来检验其安全性. 但理想安全多方计算协议存在的问题是，在遍布世界的网络环境中要找到一个可信的第三方不是一件容易的事.

2.3　半诚实模型的安全性

本文假设所有的参与者都是半诚实的参与者. 对于半诚实的参与者，在协议的执行过程中，他们都会按照协议要求忠实的履行协议，执行协议后，除协议的执行结果外没有任何信息泄露，但他们同时可能会记录下来协议执行过程中收集到的所有信息，并试图根据收集到的信息(多个参与者时我们应该考虑多个参与者在执行协议中收集到的信息，而不是一个参与者收集到的信息[5])推算出其他参与者的输入. 所以半诚实模型又称为诚实但好奇模型或被动模型.

密码学不同领域中采用的安全证明方法不同. 基于可证明安全理论的证明适用于密码学中的加密与数字签名领域，而模拟范例是目前密码学半诚实模型下研究安全多方计算时广泛接受、普遍采用的证明方法. 这种方法的原理是将一个实际安全多方计算协议的安全性与一个理想安全多方计算协议的安全性进行对比，因为理想的安全多方计算协议是安全多方计算所能达到的最高安全水平，如果一个实际的安全多方计算协议不比一个理想的安全多方计算协议泄露更多的信息，这个安全多方计算协议就认为是安全的. 理想的安全多方计算协议如下：

假设有一个绝对可信的第三方 Trent. Alice 和 Bob 要安全计算 $f(x,y)$，可以这么做：(1) Alice 和 Bob 分别把 x,y 发给 Trent；(2) Trent 计算 $f(x,y)$；(3) Trent 把计算结果告诉 Alice 和 Bob.

因为安全多方计算方案大多是利用公钥加密算法构造的，所以安全多方计算协议的安全性证明一般是在假设有关公钥加密算法安全的条件下，具体的安全多方计算协议不会比理想安全多方计算协议泄露更多的信息，实际上是把安全多方计算协议的安全性归约到公钥加密算法的安全性，而不是进一步归约到某个计算困难性问题. 也就是说假设公钥加密算法是安全的，那么一个参与者利用理想的安全多方计算协议得到的计算结果就可以模拟实际安全多方计算的过程(构造一个模拟器)，而这个模拟过程和实际的安全多方计算过程是计算不可区分的，从而说明不比理想安全多方计算协议泄露更多的信息. 这个思想的形式化描述就是模拟范例. 而本文协议的安全性证明就是采用这种普遍接受的模拟范例进行证明的.

设有 m 个参与者 $P_1,\cdots,P_m$ 分别拥有向量 $\boldsymbol{X}_1,\cdots,\boldsymbol{X}_m$，他们想要借助于协议 Π 安全地计算函数 $f(\boldsymbol{X}_1,\cdots,\boldsymbol{X}_m)$. 令 $\overline{\boldsymbol{X}}=(\boldsymbol{X}_1,\cdots,\boldsymbol{X}_m)$，在协议执行过程中，$P_j(j=1,\cdots,m)$得到的信息序列记为

$$view_j^{\Pi}(\overline{\boldsymbol{X}})=(\boldsymbol{X}_j,r_j,M_j^1,\cdots,M_j^t),$$

其中 $M_j^i(i=1,\cdots,t)$表示 P_j 第 i 次收到的信息. 对于 $I=\{j_1,\cdots,j_s\}\subseteq\{P_1,\cdots,P_m\}$，我们令

$$view_I^{\Pi}(\overline{\boldsymbol{X}})=(view_{j_1}^{\Pi}(\overline{\boldsymbol{X}}),\cdots,view_{j_s}^{\Pi}(\overline{\boldsymbol{X}})).$$

根据 Goldreich[5] 给出的关于安全多方计算半诚实模型下的安全定义，我们给出如下定义.

定义 1(半诚实参与者的安全性).　在参与者都是半诚实的情况下，我们说协议 Π 安全地计算 m 元函数 f，如果存在概率多项式时间算法 S 对于每个集合 $I\subseteq\{P_1,\cdots,P_m\}$ 均使得下式成立：

$$\{S(X_I,f_I(\overline{\boldsymbol{X}}))\}_{\overline{\boldsymbol{X}}\in(\{0,1\}^*)^m}\overset{c}{\equiv}\{(view_I^{\Pi}(\overline{\boldsymbol{X}}))\}_{\overline{\boldsymbol{X}}\in(\{0,1\}^*)^m}\tag{1}$$

其中，$X_I=(X_{j_1},\cdots,X_{j_s})$，$\overset{c}{\equiv}$ 表示计算上不可区分.

2.4　哥德尔编码

哥德尔编码是哥德尔为证明哥德尔不完备定理而引入的，其将一个非负整数序列和一个自然数建立起一一对应的关系. 有穷序列$(a_1,a_2,\cdots,a_n)$借助素数序列$(p_1,p_2,\cdots,p_n)$(其中 $p_1,p_2,\cdots,p_n$是任意 n 个不同的素数，为了使计算简单，一般选取从 2 开始的 n 个连续素数)建立如下的对应关系：

$$[a_1,a_2,\cdots,a_n]=\prod_{i=1}^{n}p_i^{a_i},$$

$[a_1,a_2,\cdots,a_n]$称作有穷序列$(a_1,a_2,\cdots,a_n)$的哥德尔数. 由于 $p_1,p_2,\cdots,p_n$是已知的 n 个素数，因此对哥德尔数$[a_1,a_2,\cdots,a_n]$是容易进行因子分解的，从而可以根据$[a_1,a_2,\cdots,a_n]$得到序列$(a_1,a_2,\cdots,a_n)$. 在本文中，我们用哥德尔编码建立起向量和自然数之间的一一对应关系.

2.5　同态加密算法

一个传统的公钥加密方案 $\mathcal{E}$ 包括 3 个算法：$\mathrm{KeyGen}_{\mathcal{E}}$，$\mathrm{Encrypt}_{\mathcal{E}}$，$\mathrm{Decrypt}_{\mathcal{E}}$.

(1) $\mathrm{KeyGen}_{\mathcal{E}}$. 给定一个安全参数 λ，$\mathrm{KeyGen}_{\mathcal{E}}$ 算法输出一个私钥 sk 和对应的公钥 pk，并给定明文空间 $\mathcal{P}$ 和密文空间 $\mathcal{C}$，即

$$(sk,pk,\mathcal{P},\mathcal{C})\leftarrow\mathrm{KeyGen}_{\mathcal{E}}(\lambda).$$

(2) $\mathrm{Encrypt}_{\mathcal{E}}$. 对于给定的公钥 pk 和明文 $M\in\mathcal{P}$ 输出相应的密文 $C\in\mathcal{C}$，即

$$C\leftarrow \mathrm{Encrypt}_{\varepsilon}(pk,M).$$

(3) $\mathrm{Decrypt}_{\varepsilon}$. 根据密文 $C\in\mathcal{C}$ 和私钥 sk 输出相应的明文 $M\in\mathcal{P}$,即

$$M\leftarrow \mathrm{Decrypt}_{\varepsilon}(sk,C).$$

一个同态加密算法 $\mathcal{E}$ 除了以上 3 种算法外还包括一个高效的 $\mathrm{Evaluate}_{\varepsilon}$ 算法. 如果 C_i 是 M_i 用公钥 pk 加密的密文,为算法 $\mathrm{Evaluate}_{\varepsilon}$ 输入公钥 pk,一种操作 S 和一个密文集合 $C=\langle C_1,\cdots,C_m\rangle$,算法 $\mathrm{Evaluate}_{\varepsilon}$ 输出密文 $C'(C'\in\mathcal{C})$,其中 C' 是 $S(M_1,\cdots,M_m)$ 的密文,即

$$\mathrm{Encrypt}_{\varepsilon}(pk,\mathrm{S}(M_1,\cdots,M_m))\leftarrow \mathrm{Evaluate}_{\varepsilon}(pk,\mathrm{S},C).$$

本文的方案中,我们用到的加密算法主要有 ElGamal 和 NTRU 两种公钥加密算法,他们均具有同态性质.

(1) ElGamal 公钥加密算法是一种具有乘法同态性质的加密算法. 具体性质如下:

① KeyGen. 给定安全参数 λ,KeyGen 产生一个大素数 p,一个生成元 g,随机地选取一个私钥 $x(x\in Z_p^*)$ 并计算其对应的公钥 h,其中

$$h=g^x \bmod p.$$

② Encrypt. 为加密消息 $M(M\in Z_p^*)$,选择一个随机数 $r(r\in Z_p^*)$,密文为

$$E(M)=(c_1,c_2)=(g^r \bmod p, Mh^r \bmod p).$$

③ Decrypt. 对于密文 $E(M)=(c_1,c_2)$,

$$\begin{aligned}M&=c_2\cdot c_1^{-x} \bmod p\\&=Mh^r\cdot(g^r)^{-x} \bmod p\\&=M\cdot(g^x)^r\cdot(g^r)^{-x} \bmod p.\end{aligned}$$

④ Evaluate. 给定消息 M_1 和 M_2 的密文 $E(M_1)$ 和 $E(M_2)$,可以计算:

$$\begin{aligned}E(M_1)\times E(M_2)&=(g^{r_1},M_1h^{r_1})\times(g^{r_2},M_2h^{r_2})\\&=(g^{r_1+r_2},(M_1\times M_2)h^{r_1+r_2})\\&=E(M_1\times M_2).\end{aligned}$$

所以 ElGamal 公钥加密算法是一种具有乘法同态性质的加密算法.

(2) NTRU 公钥加密算法是一种加法同态加密算法. 具体如下:

① KeyGen. 给定安全参数 λ,KeyGen 产生 3 个整数 (N,p,q) 和 4 个次数为 $N-1$ 的整系数多项式集合 L_f,L_g,L_r,L_M,其中 N 为素数,q 大于 p 且 $gcd(p,q)=1$. 随机地选取两个多项式 $f,g\in L_g$,多项式 f 在模 p 与 q 时必须有逆多项式 f_p,f_q,若没有逆多项式,需要舍弃重新选取(只有少数情况 f 不存在逆多项式);计算

$$h=p\cdot f_q\cdot g \bmod q.$$

则公钥为 (N,p,q,h),私钥为 (f,f_p).

② Encrypt. 随机地选取一个多项式 $r\in L_r$ 加密消息 $M(M\in L_M)$ 为

$$E(M)=rh+M(\bmod q).$$

③ Decrypt.

(a) 先计算:

$$\begin{aligned}f\cdot E(M)(\bmod q)&=f(rh+M)(\bmod q)\\&=f(rpf_qg+M)(\bmod q)=prg+fM(\bmod q)\end{aligned}$$

(因为 $ff_q \bmod q=1$);

(b) 再将上述结果乘以 f_p 并模 p 得

$$\begin{aligned}M&=f_p(prg+fM(\bmod q))\bmod p\\&=ff_pM \bmod p\\&=M \bmod p\ (\text{因为 } ff_p \bmod p=1).\end{aligned}$$

④ Evaluate. 给定密文 $E(M_1)$ 和 $E(M_2)$ 可以计算:

$$\begin{aligned}E(M_1)+E(M_2)&=(r_1h+M_1(\bmod q))+\\&\quad(r_2h+M_2(\bmod q))\\&=(r_1+r_2)h+(M_1+M_2)(\bmod q)\\&=E(M_1+M_2).\end{aligned}$$

所以 NTRU 加密算法是一种加法同态加密算法,同时 NTRU 算法效率很高[30],是公钥加密算法中效率最高的算法,加密速度至少是 RSA 的 100 倍,而且被认为是可能抵抗量子攻击[31]的公钥加密算法.

3 基于语义安全乘法同态加密算法的方案

设 m 个参与者 $P_1,\cdots,P_m$ 分别拥有向量

$$\boldsymbol{X}_1=(x_{11},\cdots,x_{1n}),\cdots,\boldsymbol{X}_m=(x_{m1},\cdots,x_{mn}),$$

他们共同对 m 个向量进行安全计算

$$\begin{aligned}\boldsymbol{X}&=f(\boldsymbol{X}_1,\cdots,\boldsymbol{X}_m)=a_1\boldsymbol{X}_1+\cdots+a_m\boldsymbol{X}_m\\&=(x_1,\cdots,x_n).\end{aligned}$$

当 $a_j=1(j=1,\cdots,m)$ 时,$f(\boldsymbol{X}_1,\cdots,\boldsymbol{X}_m)$ 简化成 m 个向量安全求和;当向量 $\boldsymbol{X}_j(j=1,\cdots,m)$ 只有一个分量时,$f(\boldsymbol{X}_1,\cdots,\boldsymbol{X}_m)$ 简化成安全求多个数的加权和与加权平均值;当向量 $\boldsymbol{X}_j(j=1,\cdots,m)$ 只有一个分量且 $a_j=1$ 时,$f(\boldsymbol{X}_1,\cdots,\boldsymbol{X}_m)$ 退化成 m 个数安全求和. 数的求和与向量求和都是向量计算的特例,根据向量计算的算法可以很容易实现,反之则实现起来比较复杂.

我们借助具有加法同态性质的 Paillier 公钥加密算法[32]给出一种简单的向量安全计算方案,具体如协议 1.

协议 1.　简单的向量安全计算.

输入：$P_1,\cdots,P_m$ 各自的秘密向量 $\boldsymbol{X}_1,\cdots,\boldsymbol{X}_m$

输出：$\boldsymbol{X}=a_1\boldsymbol{X}_1+\cdots+a_m\boldsymbol{X}_m$

1. P_1 公布 Paillier 公钥加密算法的公钥 pk，并保留私钥 sk.

2. 参与者 $P_j(j=1,\cdots,m)$ 计算如下：

(a) 用公钥 pk 将拥有的数乘向量 $a_j\boldsymbol{X}_j=(a_jx_{j1},\cdots,a_jx_{jn})$ 加密为

$$E(a_j\boldsymbol{X}_j)=(E(a_jx_{j1}),\cdots,E(a_jx_{jn})).$$

(b) 将密文向量 $E(a_j\boldsymbol{X}_j)$ 随机地分成 k_j 份（$k_j\in[1,m]$，具体分割方法见附录 1），P_j 自己保留一个份额，将剩下的 k_j-1 个份额分别发送给其他 $m-1$ 个参与者中的 k_j-1 个.

(c) 把所有收到的密文向量与自己保留的密文向量对应的分量相乘得到新的密文向量 $E(\boldsymbol{X}'_j)$，并发送给 P_1.

3. 参与者 P_1 计算如下：

(a) 将所有收到的密文向量对应的分量相乘，得到

$$E(\boldsymbol{X})=(E(a_1x_{11}+\cdots+a_mx_{m1}),\cdots,E(a_1x_{1n}+\cdots+a_mx_{mn})).$$

(b) 用私钥 sk 解密 $E(\boldsymbol{X})$ 得到 $\boldsymbol{X}$.

(c) 公布 $\boldsymbol{X}=a_1\boldsymbol{X}_1+\cdots+a_m\boldsymbol{X}_m$.

在该方案中，敌手即使获得关于 $E(a_j\boldsymbol{X}_j)$ 的 k_j-1 个份额，也无法求出 P_j 的数乘向量 $a_j\boldsymbol{X}_j$. 不妨设敌手已经得到了 $E(a_j\boldsymbol{X}_j)_1,E(a_j\boldsymbol{X}_j)_2,\cdots,E(a_j\boldsymbol{X}_j)_{k_j-1}$，因为

$$E(a_j\boldsymbol{X}_j)=\prod_{s=1}^{k_j-1}E(a_j\boldsymbol{X}_j)_s\cdot E(a_j\boldsymbol{X}_j)_{k_j},$$

敌手不知道方程中的 $E(a_j\boldsymbol{X}_j)$ 和 $E(a_j\boldsymbol{X}_j)_{k_j}$，得到的是一个有两个未知数的不定方程，无法求出 $E(a_j\boldsymbol{X}_j)$；即使敌手有私钥 sk，方程中仍有两个未知数，也无法求出 $a_j\boldsymbol{X}_j$. 如果某个敌手想要获得 $a_j\boldsymbol{X}_j$，需要得到 $E(a_j\boldsymbol{X}_j)$ 的全部份额和私钥 sk. 由于参与者 P_j 在密文分发过程中自己保留了一个 $E(a_j\boldsymbol{X}_j)$ 的份额，在协议执行过程中，敌手即使和除 P_j 之外的其他全部参与者合作也得不到 $E(a_j\boldsymbol{X}_j)$；只有在协议执行后，敌手和除 P_j 之外的其他全部参与者合作才可以推算出 $a_j\boldsymbol{X}_j$，但这和借助可信第三方的理想模型的效果一样. 因此参与者将密文分量随机地分成不确定的 k_j 份，自己保留一份并将剩下的 k_j-1 份密文分别发送给其他 $m-1$ 个参与者中的 k_j-1 个，可以抵抗其他参与者的合谋.

上述方案虽然可以抵抗参与者之间的合谋攻击，但是参与者需要将自己向量中的每个分量都加密，然后再分成 k 份发送给 k 个参与者，最终 P_1 还需要将 $E(\boldsymbol{X})$ 中的每个分量分别解密，方案中的加密、解密及通信量都与向量分量的个数 n 相关.

在公钥密码学中，为了保证要加密数的安全性，计算通常在一个非常大的群 Z_p^* 中进行，要加密的数也属于 Z_p^*. 因此协议 1 适用的范围很大，只要向量的线性组合在 Z_p^* 中. 而在实际应用中，我们要保密计算的数值通常都比较小，即使要计算的数很大，人们通常都会通过改变具体物理量的度量单位来方便计算. 比如我们说到某个百万富翁的时候通常只说他有 6 百万元，而不关心他的财富是 6 829 543 元，还是 6 824 635 元，这样把需要 7 位十进制表示的数变成了需要 1 位十进制表示的数. 对于大多数向量中的分量较小的情况，本文将向量和自然数利用哥德尔编码建立起一一对应关系，通过对自然数做乘法运算实现向量的加法运算，提高了协议的效率，同时利用密文分割的方法使协议可以抵抗参与者之间的合谋攻击.

3.1　基本原理

为了便于理解，以两个向量和的计算为例进行说明. 用哥德尔编码将向量 $\boldsymbol{X}=(x_1,\cdots,x_n)$，$\boldsymbol{Y}=(y_1,\cdots,y_n)$ 分别编码为自然数 x,y，

$$x=[x_1,\cdots,x_n]=p_1^{x_1}\cdots p_n^{x_n},$$

$$y=[y_1,\cdots,y_n]=p_1^{y_1}\cdots p_n^{y_n},$$

其中 $p_1,\cdots,p_n$ 是 n 个不同的素数. 则

$$x\cdot y=p_1^{x_1+y_1}\cdots p_n^{x_n+y_n}=[x_1+y_1,\cdots,x_n+y_n],$$

即 $\boldsymbol{X}+\boldsymbol{Y}$ 的哥德尔数等于 $\boldsymbol{X}$ 的哥德尔数与 $\boldsymbol{Y}$ 的哥德尔数之积. 利用算术基本定理，将 $x\cdot y$ 因子分解可以得到 $(x_1+y_1,\cdots,x_n+y_n)=\boldsymbol{X}+\boldsymbol{Y}$.

因此对向量利用哥德尔编码成的自然数做求积计算可以实现对应向量的求和计算[向量编码成的哥德尔数是一个很大的数，这在日常的计算中会大大增加计算量，但在密码学中这样的增加对计算的影响并不明显，是完全可以接受的（具体原因见 3.3 节的效率分析），因为公钥密码学的计算都是在一个很大的群中进行的，为了取得安全性即使非常小的自然数加密也需要在一个很大的群中进行运算，也大大增加了计算量，这是依靠公钥加密保证安全必须付出的代价]. 利用这个性质和满足 $E(x)\otimes E(y)=E(x\cdot y)$ 的语义安全乘法同态加密算法可以对多个向量的线性组合进行保密计算. 但为了抵抗合谋攻击还需要考虑一些新的策略，这就是每个参与者需要将自己的密文分成若干个份额，发送给不同的参与者.

3.2 具体方案

假设有 m 个参与者 $P_1,\cdots,P_m$，根据哥德尔编码原理，每个参与者 $P_j(j=1,\cdots,m)$ 将数乘向量 $a_j\boldsymbol{X}_j=(a_jx_{j1},\cdots,a_jx_{jn})$ 编码成一个自然数 x_j^*，具体如下：

$$x_j^*=p_1^{a_jx_{j1}}p_2^{a_jx_{j2}}\cdots p_n^{a_jx_{jn}},$$

其中 $p_1,p_2,\cdots,p_n$ 是 n 个不同的素数.

参与者 P_1 公布自己的语义安全乘法同态加密算法(现有的最著名，也最为人熟知的语义安全乘法同态加密算法是 ElGamal 公钥加密算法，对 ElGamal 加密算法的密文做乘法运算得到的是相应明文积的密文，即 ElGamal 加密算法是具有乘法同态性质的加密算法. 为了便于理解，本文采用 ElGamal 算法的同态性对方案进行说明，其他的语义安全乘法同态加密算法也同样适用，只需要在密文拆分的过程中用相应的运算替代密文间的乘法运算即可)的加密公钥 pk，并保留私钥 sk.

参与者 P_j 根据 P_1 公布的公钥 pk 将 x_j^* 加密为 $E(x_j^*)$；然后将 $E(x_j^*)$ 随机地分成非零的 $k_j(k_j\in[1,m])$ 份，即 $E(x_j^*)_1,E(x_j^*)_2,\cdots,E(x_j^*)_{k_j}$，使

$$E(x_j^*)=\prod_{s=1}^{k_j}E(x_j^*)_s$$

(做到这一点是很容易的，具体的分割方法见附录 2). P_j 自己保留一个份额，并将剩下的 k_j-1 个份额发送给其他 $m-1$ 个参与者中的 k_j-1 个. 即使敌手获得关于 x_j^* 的 k_j-1 份密文，如 $E(x_j^*)_1,E(x_j^*)_2,\cdots,E(x_j^*)_{k_j-1}$，因为

$$E(x_j^*)=\prod_{s=1}^{k_j-1}E(x_j^*)_s\cdot E(x_j^*)_{k_j},$$

敌手得到的是一个有两个未知数 $E(x_j^*)$ 与 $E(x_j^*)_k$ 的不定方程，无法求出 $E(x_j^*)$，即使敌手可以得到私钥 sk，解密后方程中仍有两个未知数，也无法求出 x_j^*，因而得不到 P_j 的数乘向量. 如果敌手想要获得 x_j^*，需要得到 $E(x_j^*)$ 的 k_j 份密文和私钥 sk. 由于 P_j 在密文分发过程中自己保留了一个 $E(x_j^*)$ 的份额，在协议执行过程中，敌手即使和除 P_j 外的其他全部参与者合作也得不到 $E(x_j^*)$；只有在协议执行后，敌手和除 P_j 之外的其他全部参与者合作才可以得到 $a_j\boldsymbol{X}_j$，但这和借助于可信第三方的理想模型的效果是一样的. 因此参与者将密文分量随机地分成不确定的 k_j 份，可以抵抗其他参与者的合谋. 每个参与者 P_j 将 $E(x_j^*)$ 的份额分发后，将收到的所有密文和自己保留的密文相乘，得到 $E(x_j')$ 并发送给 P_1.

参与者 P_1 将收到的所有密文相乘，构成 $E(x^*)$，

$$E(x^*)=E(x_1')\cdot E(x_2')\cdots E(x_m').$$

因为任何 $E(x_j^*)$ 被分成 k_j 份并发送给 k_j 个参与者，这 k_j 份之积是 $E(x_j^*)$ 且这 k_j 个参与者在 m 个参与者内，同时所有参与者都是严格按照协议执行的半诚实参与者，所以 P_j 的 k_j 份密文包含在这 k_j 个参与者最终发送给 P_1 的密文中，又由于半诚实参与者不会加入任何额外的信息，因而 P_1 构成的密文 $E(x^*)$ 是所有参与者密文的乘积，即

$$E(x^*)=E(x_1^*)\cdot E(x_2^*)\cdots E(x_m^*).$$

因为加密算法具有乘法同态性，所以

$$E(x^*)=E(x_1^*\cdot x_2^*\cdots x_m^*).$$

P_1 用自己的私钥解密 $E(x^*)$ 得到 x^*，然后借助算术基本定理将 x^* 展开如下：

$$x^*=p_1^{x_1}p_1^{x_2}\cdots p_n^{x_n},$$

其中

$$x_i=\sum_{j=1}^{m}a_jx_{ji},\ i=1,\cdots,n.$$

从而得到 m 个参与者的向量计算结果

$$\boldsymbol{X}=(x_1,\cdots,x_n)=a_1\boldsymbol{X}_1+\cdots+a_m\boldsymbol{X}_m.$$

具体如协议 2.

协议 2. 基于语义安全乘法同态加密算法的向量安全计算.

输入：$P_1,\cdots,P_m$ 各自的秘密向量 $\boldsymbol{X}_1,\cdots,\boldsymbol{X}_m$

输出：$\boldsymbol{X}=a_1\boldsymbol{X}_1+\cdots+a_m\boldsymbol{X}_m$

1. 参与者 P_1 公布加密公钥 pk，并保留私钥 sk.
2. 参与者 $P_j(j=1,\cdots,m)$ 计算如下：
   (a) 根据哥德尔编码将拥有的数乘向量 a_jx_j 编码为秘密数 x_j^*.
   (b) 用 P_1 的公钥 pk 加密 x_j^* 为 $E(x_j^*)$.
   (c) 将密文 $E(x_j^*)$ 随机地分成 $k_j(k_j\in[1,m])$ 份，自己保留一份，将剩下的 k_j-1 份分别发送给其他 $m-1$ 个参与者中的 k_j-1 个.
   (d) 把收到的所有密文和自己保留的密文相乘得到新的密文 $E(x_j')$，并发送给 P_1.
3. 参与者 P_1 计算如下：
   (a) 将所有收到的密文相乘，得到 $E(x^*)$.
   (b) 用私钥 sk 解密 $E(x^*)$ 得到 x^*.
   (c) 根据算术基本定理展开 x^*，得到
   $$\boldsymbol{X}=(x_1,\cdots,x_n)=a_1\boldsymbol{X}_1+\cdots+a_m\boldsymbol{X}_m.$$
   (d) 公布 X.

3.3 方案分析

正确性分析. 参与者的向量经过哥德尔编码，向量的每个分量成为一个素数的指数. 在乘法同态加密算法中，对密文做乘法运算相当于对明文做乘法运算，进而相当于对素数的指数做加法运算. 经过

乘法同态加密算法运算，解密得到

$$x^{*}=x_1^{*}\cdots x_m^{*}=p_1^{a_1x_{11}}\cdots p_n^{a_1x_{1n}}\cdots p_1^{a_mx_{m1}}\cdots p_n^{a_mx_{mn}}$$
$$=p_1^{a_1x_{11}+\cdots+a_mx_{m1}}\cdots p_n^{a_1x_{1n}+\cdots+a_mx_{mn}}=p_1^{x_1}\cdots p_n^{x_n},$$

所以

$$(x_1,\cdots,x_n)=a_1\boldsymbol{X}_1+\cdots+a_m\boldsymbol{X}_m=\boldsymbol{X}.$$

因此，协议 2 的输出结果是 m 个参与者想要计算的结果.

安全性分析. 利用密码学半诚实模型下安全多方计算研究中广泛接受、普遍采用的模拟范例可以证明协议 2 的安全性. 关于协议 2 的安全性，有下面的定理.

定理 1.　基于语义安全乘法同态加密算法的向量安全计算协议 2(记为 Π)是安全的.

证明.　根据 Π 中合谋的参与者的不同，分以下 3 种情况证明协议的安全性.

(1)参与者 P_1 不参与合谋，其他参与者集合 $I\subseteq\{P_2,\cdots,P_m\}$ 合谋想要推算出另一个参与者 $P_j\notin I$ 的数乘向量 $a_j\boldsymbol{X}_j$. 不妨设 P_1 外的其他参与者 $\{P_2,\cdots,P_{j-1},P_{j+1},\cdots,P_m\}$ 合谋想要推算出 $P_j(j=1,\cdots,m)$ 的数乘向量 $a_j\boldsymbol{X}_j$.

令 $X_I=(\boldsymbol{X}_2,\cdots,\boldsymbol{X}_{j-1},\boldsymbol{X}_{j+1},\cdots,\boldsymbol{X}_m)$. 通过构造使

$$\{S(X_I,f_I(\overline{\boldsymbol{X}}))\}_{\overline{\boldsymbol{X}}\in(\{0,1\}^{*})^m}\overset{c}{\equiv}\{(view_I^{\Pi}(\overline{\boldsymbol{X}}))\}_{\overline{\boldsymbol{X}}\in(\{0,1\}^{*})^m}$$

成立的概率多项式时间模拟器 S 来证明情况(1)的安全性，S 工作过程如下：

① 给定输入 $(X_I,f_I(\overline{\boldsymbol{X}}))$，令

$$(X_I,f_I(\overline{\boldsymbol{X}}))=((\boldsymbol{X}_2,\cdots,\boldsymbol{X}_{j-1},\boldsymbol{X}_{j+1},\cdots,\boldsymbol{X}_m),f_I(\overline{\boldsymbol{X}})).$$

S 随机选择两个向量 $\boldsymbol{X}_1^{\cdot},\boldsymbol{X}_j^{\cdot}$，使得 $f_I(\overline{\boldsymbol{X}})=f_I(\overline{\boldsymbol{X}}')$，其中 $\overline{\boldsymbol{X}}'=(\boldsymbol{X}_1^{\cdot},\boldsymbol{X}_2,\cdots,\boldsymbol{X}_{j-1},\boldsymbol{X}_j^{\cdot},\boldsymbol{X}_{j+1},\cdots,\boldsymbol{X}_m)$.

② S 根据哥德尔编码将数乘向量 $a_1\boldsymbol{X}_1^{\cdot},a_2\boldsymbol{X}_2,\cdots,a_{j-1}\boldsymbol{X}_{j-1},a_j\boldsymbol{X}_j^{\cdot},a_{j+1}\boldsymbol{X}_{j+1},\cdots,a_m\boldsymbol{X}_m$ 分别编码为 $x_1^{\cdot},x_2^{*},\cdots,x_{j-1}^{*},x_j^{\cdot},x_{j+1}^{*},\cdots,x_m^{*}$.

③ S 用语义安全乘法同态加密算法的公钥 pk 加密

$$x_1^{\cdot},x_2^{*},\cdots,x_{j-1}^{*},x_j^{\cdot},x_{j+1}^{*},\cdots,x_m^{*},$$

得到 m 个密文

$$E(x_1^{\cdot}),E(x_2^{*}),\cdots,E(x_{j-1}^{*}),E(x_j^{\cdot}),E(x_{j+1}^{*}),\cdots,E(x_m^{*}),$$

将密文分别随机地分成 $k_1,k_2,\cdots,k_m\in[1,m]$ 份，模拟 Π 中的分发方法将密文份额分发.

④ S 将各个密文份额随机化分发后经 Evaluate 计算得到新的 m 个密文

$$E(x_1^{\circ}),E(x_2^{\circ}),\cdots,E(x_{j-1}^{\circ}),E(x_j^{\circ}),E(x_{j+1}^{\circ}),\cdots,E(x_m^{\circ}).$$

⑤ S 对新的 m 个密文做 Evaluate 计算，得到

$$E(x^{\cdot})=E(x_1^{\circ})\cdot E(x_2^{\circ})\cdots E(x_{j-1}^{\circ})\cdot E(x_j^{\circ})\cdot E(x_{j+1}^{\circ})\cdots E(x_m^{\circ}).$$

⑥ 由于 P_1 不参与合谋，参与者集合 I 不能对 $E(x^{\cdot})$ 进行解密，具有概率多项式时间算法的敌手不能攻破语义安全乘法同态加密算法基于的困难性假设，即不能对 $E(x^{\cdot})$ 解密. 因此概率多项式时间模拟器 S 不能对密文解密，只能在协议执行结束时根据得到的 $\boldsymbol{X}$ 计算出 $x^{\cdot}$，从而得到 $\overline{\boldsymbol{X}}'$.

在本协议中

$$view_I^{\Pi}(\overline{\boldsymbol{X}})=(I,view_{j_1}^{\Pi}(\overline{\boldsymbol{X}}),\cdots,view_{j_s}^{\Pi}(\overline{\boldsymbol{X}}))$$
$$=\{(\boldsymbol{X}_2,\cdots,\boldsymbol{X}_{j-1},\boldsymbol{X}_{j+1},\cdots,\boldsymbol{X}_m),$$
$$(x_2^{*},\cdots,x_{j-1}^{*},x_{j+1}^{*},\cdots,x_m^{*}),$$
$$(E(x_2^{*}),\cdots,E(x_{j-1}^{*}),E(x_{j+1}^{*}),\cdots,E(x_m^{*})),$$
$$(E(x_2'),\cdots,E(x_{j-1}'),E(x_{j+1}'),\cdots,E(x_m')),$$
$$E(x),x,\boldsymbol{X}\},$$

令

$$S(X_I,f_I(\overline{\boldsymbol{X}}))=\{(\boldsymbol{X}_2,\cdots,\boldsymbol{X}_{j-1},\boldsymbol{X}_{j+1},\cdots,\boldsymbol{X}_m),$$
$$(x_2^{*},\cdots,x_{j-1}^{*},x_{j+1}^{*},\cdots,x_m^{*}),$$
$$(E(x_2^{*}),\cdots,E(x_{j-1}^{*}),E(x_{j+1}^{*}),\cdots,E(x_m^{*})),$$
$$E(x_2^{\cdot}),\cdots,E(x_{j-1}^{\cdot}),E(x_{j+1}^{\cdot}),\cdots,E(x_m^{\cdot})),$$
$$E(x^{\cdot}),x^{\cdot},\boldsymbol{X}^{\cdot}\},$$

因为 $\boldsymbol{X}=\boldsymbol{X}^{\cdot}$，语义安全加密算法加密的消息对概率多项式时间算法是不可区分的，密文拆分和拆分后的分发也是随机的，则模拟过程中与实际执行过程中乘法同态加密算法加密的消息、密文拆分和密文拆分后的分发是计算不可区分的，因此上述实际执行过程中得到的消息序列的 $view$ 和模拟过程中得到的消息序列的 $view$ 是计算不可区分的，即

$$\{S(X_I,f_I(\overline{\boldsymbol{X}}))\}_{\overline{\boldsymbol{X}}\in(\{0,1\}^{*})^m}\overset{c}{\equiv}\{(view_I^{\Pi}(\overline{\boldsymbol{X}}))\}_{\overline{\boldsymbol{X}}\in(\{0,1\}^{*})^m},$$

所以 Π 在情况(1)时是安全的.

(2)参与者 P_1 和其他参与者集合 $I\subseteq\{P_2,\cdots,P_m\}$ 联合想要推算出参与者 $P_j\notin I$ 的数乘向量 $a_j\boldsymbol{X}_j$. 因为 P_j 将自己的数乘向量编码成自然数 x_j^{*}，并加密为 $E(x_j^{*})$，然后随机地分成 $k_j(k_j\in[1,m])$ 份，自己保留一份，并将剩下的 k_j-1 份发送给其他 $m-1$ 个参与者中的 k_j-1 个，具体这 k_j-1 份发送给了哪 k_j-1 个参与者，其他参与者没有任何信息. 由于 P_j 保留了一份 $E(x_j^{*})$ 的份额，在协议执行过程中，除 P_j 外的全部参与者合谋，也不能得到 $E(x_j^{*})$ 的全部密文，即使有私钥，也不能得到关于 $a_j\boldsymbol{X}_j$ 的信息；只有在协议执行后，除 P_j 外的参与者联合才可以推算出 $a_j\boldsymbol{X}_j$，但这和借助可信第三方的理想模型的结果是一样的. 所以存在概率多项式时间模拟器 S 对

于参与者集合 I 使得下式成立：

$$\{S(X_I, f_I(\overline{\boldsymbol{X}}))\}_{\overline{\boldsymbol{X}}\in(\{0,1\}^*)^m} \overset{c}{\equiv} \{(view_I^{\Pi}(\overline{\boldsymbol{X}}))\}_{\overline{\boldsymbol{X}}\in(\{0,1\}^*)^m}.$$

S 的构造非常简单，只需根据 $X_I, f_I(\overline{\boldsymbol{X}})$ 随机地选择一个输入 $\boldsymbol{X}_j'$ 替代原协议执行过程中 P_j 的输入 $\boldsymbol{X}_j$ 来模拟协议的执行过程，将模拟执行过程中的 $view$ 看作 $S(X_I, f_I(\overline{\boldsymbol{X}}))$ 即可.

(3) 参与者 P_1 与参与者集合 $I\subseteq\{P_1,\cdots,P_m\}$ 合谋想要获得集合 $\bar{I}$($\bar{I}=\{P_1,\cdots,P_m\}-I$)中参与者的数乘向量. 当 $\bar{I}$ 中只有一个元素时，即其他参与者想要得到 P_j 的数乘向量，这与(2)中的情况类似，和借助于理想模型的结果一样. 当 $\bar{I}$ 中有两个元素时，即其他参与者想要推算出 P_j 和 P_{j+1} 的数乘向量，由于参与者 P_j, P_{j+1} 分别保留一个 $E(x_j^*), E(x_{j+1}^*)$ 的份额，在协议执行过程中其他参与者不能根据得到的关于 $E(x_j^*), E(x_{j+1}^*)$ 的密文份额推测出 P_j, P_{j+1} 的数乘向量，只有在协议结束时，其他所有参与者根据最终协议执行的结果，得到一个关于 P_j, P_{j+1} 数乘向量的不定方程，即

$$a_j\boldsymbol{X}_j + a_{j+1}\boldsymbol{X}_{j+1} = \boldsymbol{X} - a_1\boldsymbol{X}_1 - \cdots - a_{j-1}\boldsymbol{X}_{j-1} - a_{j+2}\boldsymbol{X}_{j+2} - \cdots - a_m\boldsymbol{X}_m,$$

得到的是 $a_j\boldsymbol{X}_j + a_{j+1}\boldsymbol{X}_{j+1}$，但求不出 $a_j\boldsymbol{X}_j$ 和 $a_{j+1}\boldsymbol{X}_{j+1}$ 的各个分量. 当 $\bar{I}$ 中有多于两个元素时，即其他参与者想要获得多于两个参与者的数乘向量时，情况类似于其他参与者想要推算出两个参与者中每个参与者数乘向量的情况. 所以 P_1 和其他参与者集合 I 合谋得不到 $\bar{I}$ 的数乘向量，只能在协议执行后得到 $\bar{I}$ 中所有参与者数乘向量的和. 因而可以用概率多项式时间模拟器 S(S 的具体构造如前所述，具体省略)对 I 得到的信息进行模拟，即下式成立：

$$\{S(X_I, f_I(\overline{\boldsymbol{X}}))\}_{\overline{\boldsymbol{X}}\in(\{0,1\}^*I)^m} \overset{c}{\equiv} \{(view_I^{\Pi}(\overline{\boldsymbol{X}}))\}_{\overline{\boldsymbol{X}}\in(\{0,1\}^*)^m}.$$

综上所述，无论任何情况，协议 2 都可以用概率多项式时间模拟器 S 对 I 进行模拟，所以协议 2 是安全的. 证毕.

效率分析. 本文将简单的向量计算方案(协议 1)和协议 2 进行比较.

计算效率分析. 忽略两个协议中进行的 Evaluate 计算. 当有 m 个参与者，每个参与者的向量有 n 个分量时，在协议 1 中，首先，每个参与者要对拥有的数乘向量中的每个分量做一次 Paillier 加密，m 个参与者共需要计算 mn 次加密；然后每个参与者对其收到的密文向量做 Evaluate 计算并发送给 P_1，P_1 对 m 个参与者发送过来的密文向量做 Evaluate 计算；最终，P_1 对 m 个参与者构成的 n 个密文分量分别解密，需要解密 n 次；整个协议中要做 mn 次 Paillier 加密和 n 次 Paillier 解密运算. 在协议 2 中，首先，每个参与者用哥德尔编码将自己的数乘向量编码成一个自然数，并对自然数加密，则 m 个参与者需要加密 m 次；在此之后，每个参与者将收到的密文做 Evaluate 计算，然后将密文积发送给 P_1；最终，P_1 对 m 个参与者发送过来的密文做 Evaluate 计算，最后做一次解密；协议 2 中共做了 m 次加密和 1 次解密运算.

通信效率分析. 在协议 1 与协议 2 中均使用了密文拆分的方法来防止参与者之间的合谋. 协议 1 中，每个参与者将向量中的 n 个分量分别加密后的密文分成 k(参与者密文拆分份数的平均值)份，自己保留一份，并将剩下的 $k-1$ 份分别随机地发送给其他 $m-1$ 个参与者中的 $k-1$ 个，同时每个参与者需要将收到的密文向量和自己保留的密文向量经 Evaluate 计算后发送给 P_1，m 个参与者之间相当于传递了 kmn 个密文. 协议 2 中，每个参与者将自己的密文分成 k 份，自己保留一份，并将剩下的 $k-1$ 份随机地发送给其他 $m-1$ 个参与者中的 $k-1$ 个，m 个参与者之间相当于传递了 $(k-1)m$ 个密文，而每个参与者将收到的密文经 Evaluate 计算后仍需要发送给 P_1，相当于 m 个参与者之间又传递了 m 个密文，协议 2 中共传递了 km 个密文.

协议 1 与协议 2 的具体比较如表 1. 其中 E_E，D_E，$|p|/2$(p 表示协议 2 中使用的乘法同态加密算法的操作是在模 p 的乘法循环群中进行的)分别代表使用语义安全乘法同态加密算法加密的计算量、解密的计算量和密文长度；E_P，D_P，$|N^2|/2$ 分别代表使用 Paillier 公钥加密算法加密的计算量、解密的计算量和密文长度.

表 1 协议 1 与协议 2 的比较

	协议 1	协议 2
计算开销	$mn\cdot E_P + n\cdot D_P$	$m\cdot E_E + D_E$
通信开销	$kmn\cdot\|N^2\|/2$	$km\cdot\|p\|/2$
是否抗合谋	是	是

为了便于说明哥德尔编码对协议 2 效率的影响，以下用 ElGamal 加密算法作为协议 2 中语义安全乘法同态加密算法的实例进行说明.

在协议 2 中，有 m 个参与者，每个参与者的向量中有 n 个分量，ElGamal 公钥加密算法的模数为 p. 参与者 P_j($j=1,\cdots,m$)用哥德尔编码将数乘向量 $a_j\boldsymbol{X}_j=(a_jx_{j1},\cdots,a_jx_{jn})$ 编码成一个自然数

x_j^*，为了保证方案的正确性，需要使 $x_j^* \in [0, \sqrt[m]{p}]$，即 $x_j^* < \log_m p$，经过哥德尔编码成的数仍在 Z_p^* 中. 本方案适用于参与者数量、参与者向量的分量个数及分量值较小的情况. 对于参与者比较多的情况，参与者可以分成不同的小组，首先在小组内计算小组的向量线性组合密文，然后将该密文发送给拥有私钥的 P_1，P_1 先对每个小组的密文进行解密，然后计算所有小组向量线性组合的和，从而得到所有参与者的向量计算结果.

在 ElGamal 公钥加密算法中，消息 $M(M \in Z_p^*)$ 加密后的密文为

$$E(M)=(c_1,c_2)=(g^r \bmod p, Mh^r \bmod p),$$

其中，$r,h,g(r,h,g \in Z_p^*)$ 分别是加密者随机选择的一个随机数、系统公钥与生成元. 协议 2 将要加密的数乘向量编码成哥德尔数 x_j^*，根据 ElGamal 公钥加密算法的加密形式，对 x_j^* 做加密运算，只需要在 $h^r \bmod p$ 的计算基础上再做一次模乘运算. 在 ElGamal 加密运算中主要的计算量是函数 $g^r \bmod p$ 与 $h^r \bmod p$ 的模指数运算，模指数运算通常采用重复平方相乘(repeated squaring and multiplication)计算法[33]，在重复平方的过程中一个小的数经过几次平方运算就变成了一个大数，一个大数可能经过一次平方取模运算就变成了一个小的数，所以在模指数运算中将原有的明文做底数和将明文的哥德尔编码作底数对计算复杂性影响不大，即采用哥德尔编码对平均加密时间没有影响. 实际上本协议在使用 ElGamal 乘法同态加密算法时，根据模重复平方计算方法，每个指数为 r 的模指数运算平均需要做 $\log r$ 次模乘运算(由于 $r \in Z_p^*$，r 的平均取值是 $p/2$，因此每次模指数运算平均需要做 $\log(p/2)$ 次模乘运算)，同时还需要计算 $M \cdot h^r \bmod p$，则一次加密运算平均需要做 $2\log r+1$ 次模乘运算. 1 次 $x_j^* \cdot h^r \bmod p$ 模乘运算相对于 $2\log r$ 次模乘运算($g^r \bmod p$ 与 $h^r \bmod p$ 模指数运算)的计算量是非常小的，因此哥德尔编码不会影响协议 2 的计算效率.

4　基于 NTRU 公钥加密算法的方案

上述基于语义安全乘法同态加密算法的效率相比简单方案有所提高，且可以抵抗参与者之间的合谋，但现有语义安全乘法同态加密算法如 ElGamal 加密算法是基于离散对数困难性假设的. 现有破解离散对数问题的算法是指数复杂性的，这在实际应用中是不可行的，是一个困难性问题，因此现有的很多密码学算法都是基于离散对数困难性假设的. 在量子计算模型下，Shor[34] 给出了关于破解离散对数问题的多项式时间算法，因此在量子计算模型下，求解离散对数问题不再是一个困难性问题. 虽然 Shor 的算法是在量子计算模型下给出的，可能与现实量子计算机中的计算有一些误差，但他同时指出这些误差属于常数范围内的，对计算结果没有决定性影响. 虽然现今仍没有制造出来量子计算机，但已经可以实现特殊问题的量子计算，随着信息技术的迅猛发展，我们将进入量子计算机时代，因此需要寻找量子计算模型下仍不能有效破解的向量计算方法.

NTRU 公钥加密算法的安全性依赖于在某些代数环上将多项式分解为小系数的多项式是困难的，攻破该系统与求解格上的最短向量问题强相关[31]，但现今已知的量子计算模型下仍不能给出破解基于格上困难性问题的多项式时间算法，虽然该问题还没有得到证明，但 NTRU 公钥加密算法仍被认为是可能抵抗量子攻击的算法. NTRU 加密算法的计算效率是公钥加密算法中最高的，加密速度至少是 RSA 的 100 倍，同时该算法具有算法简洁、占用存贮空间小的优点[30]，所以我们根据 NTRU 加密算法设计了可能抗量子攻击的高效保密向量计算方案，至少目前在没有攻破 NTRU 公钥加密算法的多项式时间算法的情况下，该方案是可以抵抗量子攻击的.

4.1　具体方案

假设有 m 个参与者 $P_1,\cdots,P_m$，每个参与者 P_j $(j=1,\cdots,m)$ 将数乘向量 $a_j \boldsymbol{X}_j=(a_jx_{j1}, a_jx_{j2},\cdots, a_jx_{jn})$ 表示为如下多项式：

$$x_j=a_jx_{j1}+a_jx_{j2}y+\cdots+a_jx_{jn}y^{n-1}(x_j \in L_M),$$

用 NTRU 公钥加密算法加密的结果为

$$E(x_j)=r_jh+x_j,$$

其中 h 是系统公钥，r_j 是参与者 P_j 随机选取的多项式. 把这 m 个参与者的密文相加，结果为

$$\begin{aligned}E(x_1)+\cdots+E(x_m)&=r_1h+x_1+\cdots+r_mh+x_m\\&=(r_1+\cdots+r_m)h+(x_1+\cdots+x_m).\end{aligned}$$

$r_1+\cdots+r_m$ 可以看作是一个随机多项式，因此密文和 $E(x_1)+\cdots+E(x_m)$ 是 $x_1+\cdots+x_m$ 的密文，对其解密后可以得到 $x_1+\cdots+x_m$，即

$$\begin{aligned}x_1+\cdots+x_m&=a_1x_{11}+a_1x_{12}y+\cdots+a_1x_{1n}y^{n-1}+\cdots+\\&\quad a_mx_{m1}+a_mx_{m2}y+\cdots+a_mx_{mn}y^{n-1}\\&=(a_1x_{11}+\cdots+a_mx_{m1})+\\&\quad(a_1x_{12}+\cdots+a_mx_{m2})y+\cdots+\\&\quad(a_1x_{1n}+\cdots+a_mx_{mn})y^{n-1}\\&=x_1+x_2y+\cdots+x_ny^{n-1},\end{aligned}$$

所以

$$(x_1,\cdots,x_n)=a_1\boldsymbol{X}_1+\cdots+a_m\boldsymbol{X}_m=\boldsymbol{X}.$$

上述内容是本方案的基本思想,但仅仅这样做不能抵抗协议执行后参与者之间的合谋攻击.因此,我们需要将每个参与者的密文分成若干份额,发送给不同的参与者,来抵抗合谋攻击.

参与者 P_1 公布 NTRU 公钥加密算法的加密公钥 $pk=(N,p,q,h)$ 和系统参数 (L_f,L_g,L_r,L_M) 并保留私钥 $sk=(f,f_p)$,其中 $h=p\cdot f_q\cdot g \bmod q$. N 是一个素数,若是一个合数,很容易恢复密钥与明文;N 也决定着算法中多项式的最大次数与格的维数,因此 N 必须足够大,以阻止格攻击. $(L_f,L_g,L_r,L_M)\subset R[x]$(其中 $R[x]$ 是一个整系数多项式集合且 $R[x]=Z[x]/(x^N-1)$). p 是小模数,但 p 要足够大,使每个多项式 x_j 的系数在 $\left[-\frac{p}{2m},\frac{p}{2m}\right]$ 中,这是为了使所有参与者拥有的秘密多项式之和 x 的系数仍在 $\left[-\frac{p}{2},\frac{p}{2}\right]$,从而可以保证 x 被正确解密. q 是大模数,为了防止解密失败,需要使 q 大于 $f\cdot E(x)$ 系数的最大值,这样使 $f\cdot E(x)$ 在中心化的过程中不会出现溢出性错误;q 要小于 N 且与 N 相当,以增加算法的安全性;p 与 q 还要满足 $gcd(p,q)=1$,否则,特别当 q 是 p 的倍数时,直接对密文模 p 便可以恢复出明文. $f(f\in L_f)$ 是一个随机选取的部分私钥多项式,需要保证多项式 f 在模 p 与 q 时有逆多项式 f_p,f_q. $g(g\in L_g)$ 是为了产生 h 而随机选取的一个秘密多项式,需要在首次使用后丢弃. $r(r\in L_r)$ 是加密者随机选取的一个随机多项式,若 r 泄露,敌手可以根据 r 计算出消息 m,因此也需要在首次使用后将其丢弃.多项式 r,g,f,m 的系数都应小于 q,以保证解密时 $prg+fm$ 的系数在 $\left[-\frac{q}{2},\frac{q}{2}\right]$,从而在模 q 时所有多项式的系数保持不变,因此可以正确恢复原始的消息 m.

与基于乘法同态加密算法中的方法类似,每个参与者 P_j 根据 P_1 公布的公钥 pk 和系统参数将自己的多项式 x_j 加密为 $E(x_j)$,然后将 $E(x_j)$ 随机地分成非零的 k_j 份 $(k_j\in[1,m])$,即 $E(x_j)_1,\cdots,E(x_j)_{k_j}$,使

$$E(x_j)=\sum_{s=1}^{k_j}E(x_j)_s$$

(具体分割方法见附录 3). P_j 自己保留一个密文份额,将剩下的 k_j-1 个份额分别发送给其他 $m-1$ 个参与者中的 k_j-1 个.即使敌手得到关于 P_j 的 k_j-1 份密文,也无法得到关于 P_j 的密文.因为

$$E(x_j)=E(x_j)_1+\cdots+E(x_j)_{k_j},$$

敌手不知道 $E(x_j)$ 和其中的一份密文如 $E(x_j)_s$,一个方程中有两个未知数,是一个不定方程,所以得不到 $E(x_j)$.即使敌手拥有私钥,对上式解密后,得到的仍然是一个不定方程,无法得到 x_j.若某个敌手想要得到 x_j,需要得到 $E(x_j)$ 的全部份额和私钥.由于 P_j 在分发密文时,自己保留了 $E(x_j)$ 的一个份额,在协议执行过程中,即使敌手和除 P_j 外的其他全部参与者合作,也得不到 $E(x_j)$,只有在协议执行后,敌手和除 P_j 外的其他参与者合作才可以得到 $a_j\boldsymbol{X}_j$,但这和借助可信第三方的理想模型的效果一样.每个参与者 P_j 将 $E(x_j)$ 的份额分发后,将收到的所有密文和自己保留的密文份额相加,得到新的密文 $E(x_j^*)$,并将其发送给 P_1.

参与者 P_1 将收到的所有多项式密文相加,构成 $E(x)$,则

$$E(x)=E(x_1^*)+\cdots+E(x_m^*).$$

因为任何 $E(x_j)$ 被分成 k_j 份并发送给 k_j 个参与者,这 k_j 份之和是密文 $E(x_j)$ 且 k_j 个参与者在 m 个参与者内,同时所有参与者都是严格按照协议执行的半诚实参与者,所以 P_j 的 k_j 份密文包含在这 k_j 个参与者最终发送给 P_1 的密文中,且这 k_j 个参与者不会加入任何额外的信息,因而 P_1 构成的多项式密文 $E(x)$ 是所有参与者多项式密文的和,即

$$E(x)=E(x_1)+\cdots+E(x_m).$$

因为 NTRU 公钥加密算法具有加法同态性,所以,

$$E(x)=E(x_1+\cdots+x_m),$$

从而得到 m 个参与者多项式的和 x,即得到 m 个参与者的向量计算结果

$$\boldsymbol{X}=a_1\boldsymbol{X}_1+\cdots+a_m\boldsymbol{X}_m.$$

在此基础上给出了基于 NTRU 公钥加密算法的向量计算解决方案,具体如下.

协议 3. 基于 NTRU 公钥加密算法的向量安全计算.

输入:$P_1,\cdots,P_m$ 各自的秘密向量 $\boldsymbol{X}_1,\cdots,\boldsymbol{X}_m$

输出:$\boldsymbol{X}=a_1\boldsymbol{X}_1+\cdots+a_m\boldsymbol{X}_m$

1. 参与者 P_1 保留自己的私钥并公布公钥 pk 和系统参数.
2. 每个参与者 $P_j(j=1,\cdots,m)$ 计算如下:
   (a) 将拥有的数乘向量 $a_j\boldsymbol{X}_j$ 用多项式表示为 x_j.
   (b) 用 P_1 的公钥 pk 和系统参数加密 x_j 为 $E(x_j)$.
   (c) 将密文 $E(x_j)$ 随机地分成 $k_j(k_j\in[1,m])$ 份,P_j 自己保留一份,将剩下的 k_j-1 份分别发送给其他 $m-1$ 个参与者中的 k_j-1 个.
   (d) 把所有收到的密文和自己保留的密文相加得到新的密文 $E(x_j^*)$,并发送给 P_1.
3. 参与者 P_1 计算如下:

(a) 将所有收到的密文相加，得到 $E(x)$.

(b) 用私钥 sk 解密 $E(x)$ 得到

$$x=x_1+x_2y+\cdots+x_ny^{n-1}.$$

(c) 进而得到 $\boldsymbol{X}=a_1\boldsymbol{X}_1+\cdots+a_m\boldsymbol{X}_m$ 并公布.

4.2 方案分析

正确性分析. 参与者将拥有的数乘向量用多项式的形式表示，然后使用 NTRU 公钥加密算法加密. 由于 NTRU 公钥加密算法具有加法同态性质，对参与者的密文做加法运算相当于对明文做加法运算，同时多项式相加等于对应项系数分别相加. m 个参与者经过 NTRU 加法同态加密算法运算后，解密后得到

$$\begin{aligned}x&=x_1+x_2+\cdots+x_m\\&=(a_1x_{11}+\cdots+a_mx_{m1})+\\&\quad(a_1x_{12}+\cdots+a_mx_{m2})y+\cdots+\\&\quad(a_1x_{1n}+\cdots+a_mx_{mn})y^{n-1}\\&=x_1+x_2y+x_ny^{n-1}.\end{aligned}$$

又因为

$$(x_1,\cdots,x_n)=a_1\boldsymbol{X}_1+\cdots+a_m\boldsymbol{X}_m=\boldsymbol{X},$$

所以 m 个参与者经协议 3 计算后得到的结果是他们的向量计算结果.

安全性分析. 协议 3 的安全性具体如下推论.

推论 1. 基于 NTRU 公钥加密算法的向量安全计算协议 3 是安全的.

类似定理 1 的证明，推论 1 很容易证明，在此省略.

由协议 2 的安全性分析可知，协议 2 安全主要是因为基于困难性假设的语义安全乘法同态加密算法是安全的，多项式时间模拟器 S 的构造表明半诚实的参与者不能从协议获得比理想安全多方计算协议更多的信息，即参与者获得的 $view$ 和模拟得到的 $view$ 是计算不可区分的，所以是安全的. 因为 NTRU 基于的困难性问题与格上最短向量问题强相关，被认为是可能抵抗量子攻击的公钥密码算法，即使攻击者具有量子计算能力，也不能攻破 NTRU 加密算法，因此 NTRU 公钥加密算法是可能抵抗量子攻击的. NTRU 公钥加密算法虽然不是语义安全的加密算法[35]，但在协议 3 中，参与者对密文的拆分是随机的，敌手不能根据某个参与者的部分密文得到该参与者明文的任何信息；同时密文分发的过程也是随机的，由于参与者自己保留了一个密文份额，敌手在协议执行过程中不能得到关于某个参与者的全部密文份额，因此即使有解密密钥，也不能得到该参与者的数乘向量，只有在协议执行后，敌手可以推算出不参与合谋的参与者集合的数乘向量之和，而不能推算出某个不合谋参与者的数乘向量，这与借助可信第三方的理想模型相同. 因此参与者在协议 3 执行过程中得到的 $view$ 和模拟过程中得到的 $view$ 是计算不可区分的，所以协议 3 是可能具有量子安全的.

效率分析. 当有 m 个参与者，每个参与者的向量有 n 个分量时，每个参与者使用具有加法同态性的 NTRU 公钥加密算法将自己的数乘向量构成的多项式加密，借助于快速傅里叶变换需要 $N\log N$ 次模 q 运算，而解密需要先计算模 q 然后计算模 p，需要 $2N\log N$ 次模运算. 在协议 3 中，m 个参与者共需要做 m 次加密运算和 1 次解密运算，则共需要计算 $(m+2)N\log N$ 次模 q 运算. NTRU 公钥加密算法的每个密文最多包含 N 项系数，每个系数最多为 $|q|$ 位，每个密文的长度最多为 $N\cdot|q|$. 由于每个参与者将其密文拆分成 k(参与者密文拆分份数的平均值)份，自己保留一个份额，将剩下的 $k-1$ 份分别发送给 $k-1$ 个参与者，同时仍需要将收到的密文与自己保留的密文之和发送给 P_1，因此参与者之间的通信量是 $km\cdot N|q|$.

在协议 2 中，以 ElGamal 公钥加密算法为例. ElGamal 公钥加密算法加密一次消息需要 $2p_E^3$ 次模 p_E 运算，解密一次消息需要 p_E^3 次模 p_E 运算. 整个计算过程需要加密 m 次和解密 1 次，则共需要 $(2m+1)p_E^3$ 次模 p_E 运算. 每个参与者将自己的密文分成 k(参与者密文拆分份数的平均值)份，自己保留一个份额，将剩下的 $k-1$ 份分别发送给 $k-1$ 个参与者，同时将其收到的密文与自己保留的密文之和发送给 P_1，共需要传递 2 次密文，则在协议 2 中，需要 $2km\cdot|p_E|$ 的通信量，其中 p_E 代表 ElGamal 加密算法的模素数. 协议 2 和协议 3 之间的比较如表 2.

表 2　协议 2 与协议 3 的比较

	协议 2	协议 3
计算开销	$(2m+1)p_E^3$ 模 p_E	$(m+2)N\log N$ 模 q
通信开销	$2km\cdot\|p_E\|$	$km\cdot N\|q\|$
基于的困难性问题	离散对数困难性	格上最短向量困难性

5 应　用

在实际的应用中，我们通常需要对多组数据进行安全计算，并且每组数据之间相互没有联系，利用安全多方向量计算协议可以很容易地解决多组数据同时安全计算的问题，下面我们给出了两个具体问

题的解决方法.

5.1 多项数据的安全统计

假设有 5 个参与者 Alice、Bob、Carol、Dave、Ella 他们想要统计共有的人民币、美元、英镑和欧元各是多少,但出于个人隐私考虑,他们都不想让其他参与者知道自己各种币种的量,这需要他们安全地统计多项数据的和. 该问题可以看作是安全多方向量计算的一个特例. 如果 Alice、Bob、Carol、Dave、Ella 分别拥有的币种是人民币、美元、英镑和欧元,同时也知道各种钱币兑换成人民币的汇率,他们想要安全地计算共同拥有的钱币兑换成人民币之和,这个问题也可以用安全向量计算的方法实现. Alice 拥有 A_1 人民币、A_2 美元、A_3 英镑、A_4 欧元,相当于 Alice 拥有向量

$$\boldsymbol{A}=(A_1,A_2,A_3,A_4),$$

与 Alice 相同,Bob、Carol、Dave、Ella 分别拥有向量

$$\boldsymbol{B}=(B_1,B_2,B_3,B_4),$$
$$\boldsymbol{C}=(C_1,C_2,C_3,C_4),$$
$$\boldsymbol{D}=(D_1,D_2,D_3,D_4),$$
$$\boldsymbol{E}=(E_1,E_2,E_3,E_4),$$

他们共同计算 5 个向量的和 $\boldsymbol{S}$,

$$\boldsymbol{S}=(S_1,S_2,S_3,S_4),$$

其中

$$S_i=A_i+B_i+C_i+D_i+E_i(i=1,2,3,4).$$

现有安全统计多项数据和的方法主要有两种. 一种方法[19]可以表示为 Alice 将每种钱币分别加入随机数,然后用 Bob 的公钥加密发送给 Bob,Bob 解密后加入自己的钱币数,然后 Bob 再用 Carol 的公钥加密发送给 Carol,如此循环,直到 Ella 加入自己的钱币后发送给 Alice,Alice 解密后减去自己加入的随机数,得到共有的各种钱币. 另外一种方法[18,25]是借助于加法同态加密算法,Alice 公布自己的公钥,然后每个参与者加密自己的各种钱币数,Alice 将各种钱币的密文发送给 Bob,Bob 计算 Alice 和自己各种钱币的同态密文和,然后发送给 Carol. 如此循环,直到 Ella 将 Alice、Bob、Carol、Dave 和自己各种钱币的密文发送给 Alice,Alice 将其解密,得到他们共同的各种钱币之和.

以上两种做法都需要每个参与者为各种钱币加密,这样无疑增加了计算复杂性,也增加了通信复杂性. 由于是每一项单独加密,可能会泄露某个参与者具体币种的数量. 借助于协议 3,可以实现高效且抗合谋的多项数据保密统计,具体如下.

协议 4. 多项数据和的安全统计协议.

输入:Alice、Bob、Carol、Dave、Ella 各自拥有钱币的向量 $\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D},\boldsymbol{E}$

输出:$\boldsymbol{S}=\boldsymbol{A}+\boldsymbol{B}+\boldsymbol{C}+\boldsymbol{D}+\boldsymbol{E}$

1. Alice 公布自己 NTRU 公钥加密算法的公钥 pk 及系统参数.
2. Alice 将自己向量的分量作为多项式的系数编码成多项式 $x_{\boldsymbol{A}}$,即

$$x_{\boldsymbol{A}}=A_1+A_2y+A_3y^2+A_4y^3.$$

然后将 $x_{\boldsymbol{A}}$ 用公钥 pk 加密为 $E(x_{\boldsymbol{A}})$,并将 $E(x_{\boldsymbol{A}})$ 随机地分成非零的 4 份(份数可以随机选取)$E(x_{\boldsymbol{A}})_1$,$E(x_{\boldsymbol{A}})_2$,$E(x_{\boldsymbol{A}})_3$,$E(x_{\boldsymbol{A}})_4$ 使

$$E(x_{\boldsymbol{A}})=E(x_{\boldsymbol{A}})_1+E(x_{\boldsymbol{A}})_2+E(x_{\boldsymbol{A}})_3+E(x_{\boldsymbol{A}})_4.$$

不失一般性,Alice 自己保留 $E(x_{\boldsymbol{A}})_4$,然后将 $E(x_{\boldsymbol{A}})_1$,$E(x_{\boldsymbol{A}})_2$,$E(x_{\boldsymbol{A}})_3$ 分别发送给 Bob、Carol、Dave. 不妨令 Bob、Carol、Dave、Ella 将各自向量的密文 $E(x_{\boldsymbol{B}})$,$E(x_{\boldsymbol{C}})$,$E(x_{\boldsymbol{D}})$,$E(x_{\boldsymbol{E}})$ 也分为 4 份,各自保留一份,并将剩下的 3 份发送给其后面的 3 个参与者.

3. Alice 将收到的所有密文和自己保留的密文之和记为 $E(x_{\boldsymbol{A}}^*)$,则

$$E(x_{\boldsymbol{A}}^*)=E(x_{\boldsymbol{A}})_4+E(x_{\boldsymbol{E}})_1+E(x_{\boldsymbol{D}})_2+E(x_{\boldsymbol{C}})_3.$$

与 Alice 相同,Bob、Carol、Dave、Ella 分别将其收到的所有密文和自己保留的密文和记为 $E(x_{\boldsymbol{B}}^*)$,$E(x_{\boldsymbol{C}}^*)$,$E(x_{\boldsymbol{D}}^*)$,$E(x_{\boldsymbol{E}}^*)$ 并发送给 Alice.

4. Alice 将刚收到的密文与自己上一步收到的密文 $E(x_{\boldsymbol{A}}^*)$ 相加,构成 $E(x)$,则

$$\begin{aligned}E(x)&=E(x_{\boldsymbol{A}}^*)+E(x_{\boldsymbol{B}}^*)+E(x_{\boldsymbol{C}}^*)+E(x_{\boldsymbol{D}}^*)+E(x_{\boldsymbol{E}}^*)\\&=E(x_{\boldsymbol{A}})+E(x_{\boldsymbol{B}})+E(x_{\boldsymbol{C}})+E(x_{\boldsymbol{D}})+E(x_{\boldsymbol{E}})\\&=E(x_{\boldsymbol{A}}+x_{\boldsymbol{B}}+x_{\boldsymbol{C}}+x_{\boldsymbol{D}}+x_{\boldsymbol{E}}),\end{aligned}$$

然后 Alice 用其私钥 sk 将 $E(x)$ 解密得 x,

$$\begin{aligned}x&=(A_1+\cdots+E_1)+(A_2+\cdots+E_2)y+\cdots+\\&\quad(A_4+\cdots+E_4)y^3\\&=S_1+S_2y+\cdots+S_4y^3,\end{aligned}$$

从而得到向量

$$\boldsymbol{S}=(S_1,S_2,S_3,S_4)=\boldsymbol{A}+\boldsymbol{B}+\boldsymbol{C}+\boldsymbol{D}+\boldsymbol{E},$$

即得到共有的人民币、美元、英镑、欧元分别为 S_1,S_2,S_3,S_4.

5. Alice 公布 $\boldsymbol{S}$.

推论 2. 多项数据和的安全统计协议是安全的.

借助于推论 1 的证明,该推论很容易证明,在此省略.

5.2 n 选 k 的安全选举

现有的电子选举方案主要利用混合网、盲签名和同态加密算法来实现安全的选举方案. 混合网和盲签名需要借助于理想的匿名通道,在现实中难以实现. 由于同态加密算法可以很好的应用于云计算

环境中，且不需要特定的运行环境，因此基于同态加密算法的选举算法有更好的应用前景. 但是现有基于 Paillier 加法同态加密算法或改进的 ElGamal 加法同态加密算法的选举方案中存在一些问题. 首先，当有 m 名选民且每名选民可以选 n 名候选人中的 k 名时，每名选民为了保证选举的安全性需要使用加法同态性质的加密算法对每名候选人的选票进行加密，计算量和通信量都和候选人的人数 n 相关，当候选人较多时计算比较困难；其次，算法安全性基于的困难性问题在量子计算模型下不再是困难的；最后，现有的算法为了安全计票，要么需要将密钥在多个参与者之间进行秘密共享，要么需要一个专门的计票中心，来抵抗参与者之间的合谋，这都需要额外的计算量. 为此，在协议 3 的基础上，我们提出了一种基于半诚实模型下可能抵抗量子攻击，同时可以抵抗合谋的高效且安全的 n 选 k 选举协议，具体如下.

协议 5.　n 选 k 的安全选举协议.

输入：$P_1,\cdots,P_m$ 各自的秘密向量 $\boldsymbol{X}_1,\cdots,\boldsymbol{X}_m$

输出：$\boldsymbol{X}=\boldsymbol{X}_1+\cdots+\boldsymbol{X}_m$

1. 准备阶段. 权威中心公布 NTRU 公钥加密算法的公钥、系统参数和 n 名候选人的编码 $i(i=0,1,\cdots,n-1)$.
2. 选票阶段. 每名选民 $P_j(j=1,\cdots,m)$ 将选票编码成多项式的形式：
$$x_j=a_{j0}+a_{j1}y+\cdots+a_{j(n-1)}y^{n-1},$$
如果 P_j 选的人中有 i，则多项式第 i 项的系数 $a_{ji}=1$，否则，$a_{ji}=0$. P_j 将选票 x_j 加密为 $E(x_j)$，并将 $E(x_j)$ 随机地分成 $k_j(k_j\leq m)$ 份，自己保留一份，剩下的 k_j-1 份分别发送给 $m-1$ 名选民中的 k_j-1 名（其他选民不知道 P_j 具体将这 k_j-1 份密文发送给了 $m-1$ 名选民中的哪 k_j-1 名）. 每名选民 P_j 将 $E(x_j)$ 都分发后，将其收到的所有密文和自己保留的密文相加，记为 $E(x_j^*)$，并发送给权威中心.
3. 计票阶段. 权威中心将所有收到的密文相加，得到密文 $E(x)$，然后根据保存的 NTRU 公钥加密算法的私钥解密 $E(x)$，得到多项式
$$x=a_0+a_1y+\cdots+a_{n-1}y^{n-1},$$
从而得到 n 名候选人中每人获得的选票：
$$(a_0,a_1,\cdots,a_{n-1})=\boldsymbol{X}_1+\cdots+\boldsymbol{X}_m=\boldsymbol{X}.$$
4. 公布阶段. 权威中心公布 $\boldsymbol{X}$.

推论 3.　n 选 k 的安全选举协议是安全的.

根据推论 1 的安全性证明，该推论的安全性很容易证明，在此省略其证明过程.

6　结　语

向量计算是许多安全多方计算的基本模块，在保密统计、隐私保护的数据挖掘、保密选举等方面有重要的应用. 现有的解决方案只是针对向量计算的特例向量求和与数据求和，但它们的计算量与向量的分量个数相关，计算效率比较低. 本文首先给出一个高效的向量计算方案. 考虑到该方案不能适用于量子计算模型的安全要求，本文又给出了一种可能抵抗量子攻击的高效向量计算方案，并用该方案解决了多项数据和的安全统计问题和多选多的安全选举问题. 本文中方案的计算复杂性不再与向量中分量的个数直接相关，但方案中使用的 NTRU 公钥加密算法，目前可以找到的文献建议的小模数 p 基本上是 3 和 5，这样做主要是为了保证 NTRU 加密算法的计算效率. 这意味着按照现有的参数建议，利用 NTRU 算法构造的协议只有当各个向量与它们线性组合成的向量的各个分量值都小于 5 时是有效的，超出可能会出错. 但如果有量子计算机的话，计算复杂性不成为问题，而抗量子攻击则成为协议的一个明显的优势. 因此如何在保证效率的同时增大 p 的值是我们进一步研究的问题.

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. Santa Barbara, USA, 1997: 1-6

[2] Yao A C. Protocols for secure computations//Proceedings of the 23th IEEE Symposiumon Foundations of Computer Science. Chicago, USA, 1982: 160-164

[3] Ben-Or M, Goldwasser S, Wigderson A. Completeness theorems for non-cryptographic fault-tolerant distributed computation//Proceedings of the 20th Annual ACM Symposium on Theory of Computing. Chicago, USA, 1988: 1-10

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

[5] Goldreich O. The Fundamental of Cryptography: Basic Applications. London, England: Cambridge University Press, 2004

[6] Brassard G, Chaum D, Crépeau C. Minimum disclosure proofs of knowledge. Journal of Computer and System Sciences, 1988, 37(2): 156-189

[7] Goldwasser S, Micali S, Rackoff C. The knowledge complexity of interactive proof systems. SIAM Journal on Computing, 1989, 18(1): 186-208

[8] Ioannidis I, Grama A. An efficient protocol for Yao's Millionaires' problem//Proceedings of the 36th Hawaii International Conference on System Science. Honolulu, USA, 2003: 1-6

[9] Li Shun-Dong, Wang Dao-Shun. Efficient secure multiparty computation based on homomorphic encryption. Chinese Journal of Electronics, 2013, 41(4): 798-803(in Chinese)
(李顺东, 王道顺. 基于同态加密的高效多方保密计算. 电子学报, 2013, 41(4): 798-803)

[10] Li S D, Wu C Y, Wang D S, Dai Y Q. Secure multiparty computation of solid geometric problems and their applications. Information Sciences, 2014, 282: 401-413

[11] Fagin R, Naor M, Winklery P. Comparing information without leaking it. Communications of the ACM, 1996, 39(5): 77-85

[12] Du W L, Atallah M J. Privacy-preserving cooperative statistical analysis//Proceedings of the 17th Annual Conference of Computer Security Applications. New Orleans, USA, 2001: 102-110

[13] Du W L, Atallah M J. Protocols for secure remote database access with approximate matching//Anup K G ed. Advance of E-Commerce and Privacy. Heidelberg, Germany: Springer, 2001: 87-111

[14] Cachin C. Efficient private bidding and auctions with a noblivious third party//Proceedings of the 6th ACM Conferenceon Computer and Communications Security. Singapore, 1999: 120-127

[15] Koh H C, Tan G. Data mining applications in healthcare. Journal of Healthcare Information Management, 2011, 19(2): 64-72

[16] Smyth B, Ryan M, Kremer S, et al. Towards automatic analysis of election verifiability properties//Armando A, Lowe G eds. Automated Reasoning for Security Protocol Analysis and Issues in the Theory of Security. Heidelberg, Germany: Springer, 2011: 146-163

[17] Drosatos G, Efraimidis P S. Privacy-preserving statistical analysis on ubiquitous health data//Furnell S, Lambrinoudakis C, Pernul G eds. Trust, Privacy and Security in Digital Business. Heidelberg, Germany: Springer, 2011: 24-36

[18] Huszti A. A homomorphic encryption-based secure electronic voting scheme. Publicationes Mathematicae Debrecen, 2011, 79(3): 479-496

[19] Wang Ke, Dai Yi-Qi. Secure multiparty computation of statistical distribution. Journal of Computer Research and Development, 2010, 47(2): 201-206(in Chinese)
(王克, 戴一奇. 统计分布的多方保密计算. 计算机研究与发展, 2010, 47(2): 201-206)

[20] Kantardzic M. Data Mining: Concepts, Models, Methods, and Algorithms. Hoboken, USA: John Wiley & Sons, 2011

[21] Dong R. Secure Multiparty Computation[M. S. dissertation]. Bowling Green State University, Bowling Green, USA, 2009

[22] Clifton C, Kantarcioglu M, Vaidya J, et al. Tools for privacy preserving distributed data mining. ACM Sigkdd Explorations Newsletter, 2002, 4(2): 28-34

[23] Shukla S, Sadashivappa G, Mishra D K. Simulation of collision resistant secure sum protocol. arXiv preprint arXiv: 1411.7756, 2014

[24] Sun Mao-Hua. Research on Secure Multi-Partycomptation and Its Application[Ph. D. dissertation]. Beijing University of Posts and Telecommunications, Beijing, 2013(in Chinese)
(孙茂华. 安全多方计算及其应用研究[博士学位论文]. 北京邮电大学, 北京, 2013)

[25] Damgard I, Jurik M. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system //Kim K ed. Public Key Cryptography. Heidelberg, Germany: Springer, 2001: 119-136

[26] ElGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms//Blakely G R, Chaum D eds. Advances in Cryptology. Heidelberg, Germany: Springer, 1984: 10-18

[27] Okamoto T, Takashima K. Adaptively attribute-hiding (hierarchical) inner product encryption. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2016, 99(1): 92-117

[28] Cao N, Wang C, Li M, et al. Privacy-preserving multi-keyword ranked search over encrypted cloud data. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(1): 222-333

[29] Hoffstein J, Pipher J, Silverman J H. NTRU: A ring-based public key cryptosystem//Buhler J P ed. Algorithmic Number Theory. Heidelberg, Germany: Springer, 1998: 267-288

[30] Hermans J, Vercauteren F, Preneel B. Speed records for NTRU//Pieprzyk J ed. Topics in Cryptology-CT-RSA 2010. Heidelberg, Germany: Springer, 2010: 73-88

[31] Perlner R A, Cooper D A. Quantum resistant public key cryptography: A survey//Proceedings of the 8th Symposium on Identity and Trust on the Internet. Gaithersburg, USA, 2009: 85-93

[32] Paillier P. Public-key cryptosystems based on composite degree residuosity classes//Stern J ed. Advances in Cryptology-EUROCRYPT'99. Prague, Czech Republic, 1999: 223-238

[33] Chen Gong-Liang. The Mathematical Fundamental of Information Security. Vol. 2. Beijing: Tsinghua University Press, 2014(in Chinese)
(陈恭亮. 信息安全数学基础. 第 2 版. 北京: 清华大学出版社, 2014)

[34] Shor P W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing, 1997, 26(5): 1484-1509

[35] Nguyen P Q, Pointcheval D. Analysis and improvements of NTRU encryption paddings//Yung Moti eds. Advances in Cryptology-CRYPTO 2002. Santa Barbara, USA, 2002: 210-225

附录 1.

参与者 P_j 将其密文向量 $E(a_j\boldsymbol{X}_j)=(E(a_jx_{j1}),\cdots,E(a_jx_{jn}))$ 随机地分成 $k_j(k_j\leq m)$ 份的分割方法. 在 Paillier 公钥加密算法中，

$$E(a_jx_{ji})=g^{a_jx_{ji}}r^N \bmod N^2,$$

$$E(a_jx_{ji})\cdot E(a_jx_{j(i+1)})=g^{a_jx_{ji}}r_i^N\cdot g^{a_jx_{j(i+1)}}r_{i+1}^N \bmod N^2$$
$$=g^{a_jx_{ji}+a_jx_{j(i+1)}}(r_i\cdot r_{i+1})^N \bmod N^2,$$

因此 Paillier 加密算法具有加法同态性，为了根据拆分后的密文得到明文，需要拆分后的 k_j 份密文满足

$$E(a_jx_{ji})=\prod_{s=1}^{k_j}E(a_jx_{ji})_s.$$

但是 $E(a_jx_{ji})$ 的因子有可能少于 k_j 个，这样将不能直接把 $E(a_jx_{ji})$ 分成 k_j 份；又由于 Paillier 公钥加密算法的密文属于 $Z_{N^2}^*$，密文空间很大，通过对 $E(a_jx_{ji})$ 直接进行因子分解来实现将密文分解成 k_j 份是困难的. 因为对于每个密文分量 $E(a_jx_{ji})$ 的 k_j 份密文只需要满足等式

$$E(a_jx_{ji})_1\cdots E(a_jx_{ji})_{k_j}=E(a_jx_{ji})$$

成立，因此 P_j 只需要先把 a_jx_{ji} 加密 $E(a_jx_{ji})$，然后随机地选取 k_j 个随机数 $r_{ji}^1,\cdots,r_{ji}^{k_j}$，同时保证 $r_{ji}^1\cdots r_{ji}^{k_j}=1 \bmod N^2$，随机地这从 k_j 个随机数中选择一个，不妨为 r_{ji}^1，计算 $r_{ji}^1\cdot E(a_jx_{ji})$，并将其作为密文份额 $E(a_jx_{ji})_1$，令 $r_{ji}^{k_j},\cdots,r_{ji}^2$ 分别为 $E(a_jx_{ji})_2,\cdots,E(a_jx_{ji})_{k_j}$，此时

$$E(a_jx_{ji})_1\cdot E(a_jx_{ji})_2\cdots E(a_jx_{ji})_{k_j}=$$
$$r_{ji}^1E(a_jx_{ji})_1\cdot r_{ji}^{k_j}\cdots r_{ji}^2 \bmod N^2=E(a_jx_{ji}),$$

从而将 $E(a_jx_{ji})$ 份成 k_j 份. P_j 每次随机地从每个分量中选择一份不同的密文，构成一个密文向量 $E(a_j\boldsymbol{X}_j)_s(s=1,\cdots,k_j)$，从而实现将 $E(a_j\boldsymbol{X}_j)$ 分成 k_j 份.

附录 2.

参与者 P_j 将其密文 $E(x_j^*)$ 分成 $k_j(k_j\leq m)$ 份的分割方法. 在乘法同态加密算法中，$E(x)\cdot E(y)=E(x\cdot y)$. 为了根据分解后的密文得到明文，分成的 k_j 份密文需要满足

$$E(x_j^*)=\prod_{s=1}^{k_j}E(x_j^*)_s.$$

但是由于乘法同态加密算法的密文属于 Z_p^*，密文空间很大，很难将 $E(x_j^*)$ 直接进行因子分解来实现将密文分解成 k_j 份；同时 $E(x_j^*)$ 的因子有可能少于 k_j 个，这样将不能直接把 $E(x_j^*)$ 分成 k_j 份. 由于 $E(x_j^*)$ 只需要满足等式

$$E(x_j^*)_1\cdots E(x_j^*)_{k_j}=E(x_j^*)$$

成立，因此 P_j 只需要先对 x_j^* 进行加密，得到密文 $E(x_j^*)$，然后随机地选取 k_j 个随机数 $r_{j1},\cdots,r_{jk_j}$，使其满足 $r_{j1}\cdots r_{jk_j}=1 \bmod p$，在 k_j 个随机数中随机的选择一个，不妨为 r_{j1}，计算 $r_{j1}\cdot E(x_j^*)$ 并将其作为密文份额 $E(x_j^*)_1$，令 $r_{j2},\cdots,r_{jk_j}$ 分别为 $E(x_j^*)_2,\cdots,E(x_j^*)_{k_j}$，此时

$$E(x_j^*)_1\cdot E(x_j^*)_2\cdots E(x_j^*)_{k_j}=$$
$$E(x_j^*)\cdot r_{j1}\cdots r_{jk_j} \bmod p=E(x_j^*),$$

从而实现将 P_j 的密文分成 k_j 份.

附录 3.

参与者 P_j 将其密文 $E(x_j)$ 分成 $k_j(k_j\leq m)$ 份的分割方法. 在 NTRU 公钥加密算法中，

$$E(x_j)=r_jh+x_j(\bmod q).$$

因为 NTRU 公钥加密算法具有加法同态性，为了根据分解后的密文得到明文，所以分成的 k_j 份密文需要满足

$$E(x_j)=\sum_{s=1}^{k_j}E(x_j)_s.$$

P_j 将密文 $E(x_j)$ 分成 k_j 份的方法有两种. 具体如下：

(1) 与附录 2 中乘法同态密文分割的方法类似，P_j 随机地选择 k_j 个随机多项式 $r_{j1},\cdots,r_{jk_j}$，并满足 $r_{j1}+\cdots+r_{jk_j}=0 \bmod q$，在 k_j 个随机多项式中随机地选择一个不妨为 r_{j1}，计算 $r_{j1}+E(x_j)$ 并作为密文份额 $E(x_j)_1$，令 $r_{j2},\cdots,r_{jk_j}$ 分别为 $E(x_j)_2,\cdots,E(x_j)_{k_j}$，此时

$$E(x_j)_1+E(x_j)_2+\cdots+E(x_j)_{k_j}=$$
$$E(x_j)+r_{j1}+\cdots+r_{jk_j}(\bmod q)=E(x_j),$$

从而实现参与者 P_j 对密文的拆分.

(2) 参与者 P_j 直接对密文 $E(x_j)$ 分割. 随机地将密文 $E(x_j)$ 分成 k_j 份 $E(x_j)_1,\cdots,E(x_j)_{k_j}$，使

$$E(x_j)=E(x_j)_1+\cdots+E(x_j)_{k_j},$$

同时，这 k_j 份明文可以包含参与者 P_j 加入的假份额，因为密文多项式在模 q 范围内的，只需要保证加入的假份额之和模 q 为零，从而实现参与者 P_j 对密文的拆分.

参与者 P_j 使用方法(1)或方法(2)得到的 k_j 份密文和都是密文 $E(x_j)$，因此在同一个方案中每个参与者使用哪种密文分解方法对其他参与者没有影响.

ZHOU Su-Fang, born in 1990, Ph.D. candidate. Her main research interests include cryptography and information security.

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

GUO Yi-Min, born in 1992, Ph.D. candidate. Her research main research interests include cryptography and information security.

MAO Qing, born in 1973, Ph.D. candidate, lecturer. His research interests focus on secure multiparty computation.

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

Background

Secure multiparty computation (SMC) is one of the most important research fields in cryptography. Secure multiparty vector computation (SMVC) is an important aspect of SMC and has applications in many fields, such as secure elections, privacy-preserving statistical analysis, electronic voting and privacy-preserving data mining.

Research into SMVC is limited. Existing protocols for SMVC are focused on secure multiparty vector sum computation and secure multiparty data sum computations, which are special cases of general secure multiparty vector computation. These protocols use homomorphic encryption primitives to compute vector components individually, in order to perform secure multiparty vector sum computation. These are not secure multiparty vector sum computations and cannot resist collusion attacks.

In this study, we use Paillier's additively homomorphic encryption scheme to design a SMVC protocol (Protocol 1) resistant to collusion attack. Using Gödel encoding to establish a 1:1 mapping between a vector and a natural number (Gödel numbering) and using the semantic security multiplicatively homomorphic encryption scheme, we devise a protocol for privately computing the linear combination of multi-vectors (Protocol 2), which's components are small, that is computationally efficient and secure against collusion attack.

To protect against quantum attack, we use the NTRU additively homomorphic encryption scheme—which is a lattice-based cryptosystem, unbreakable even using quantum computers—to design a computationally efficient protocol for privately computing linear combinations of multi-vectors (Protocol 3). This scheme is likely secure against both collusion attack and quantum attack. While the based NTRU encryption scheme's littler mode palways token as 3 or 5, the plaintext of the protocol must in $[-2, 2]$. It is the further study problem that why there are no literature suggest greaterpand what will happen using a biggerp.

In order to demonstrate applications of the proposed protocols, we utilize an efficient secure statistical protocol (Protocol 4) and a secure electronic election protocol (Protocol 5). Each protocol is proven to be secure using the semi-honest model. This work is supported by the National Natural Science Foundation of China (Grant No. 61272435) and the Fundamental Research Funds for the Central Universities (Grant No. 2016TS061).