高技术通讯 2012 年 第22 卷 第12 期:1225 ~ 1232

doi:10.3772/j.issn.1002-0470.2012.12.002

基于保护隐私同余方程组协议的多秘密共享算法①

孙茂华②*** 李 涵*** 罗守山***** 辛 阳***

(*北京邮电大学信息安全中心 北京 100876)
(**灾备技术国家工程实验室 北京 100876)
(***北京安码科技有限公司 北京 100876)

摘 要 针对 Asmuth-Bloom 秘密共享方案存在的安全、复杂度和存储空间问题,研究了保护隐私的同余方程组的求解问题——安全多方科学计算领域中的新课题。在半诚实模型下利用中国剩余定理、安全多方求和协议和分布式 ElGamal 同态加密协议,设计了保护隐私的同余方程组求解,分析了该协议的正确性、安全性和复杂性,并利用该协议设计了多秘密共享方案,该方案解决了 Asmuth-Bloom 秘密共享方案中存在的问题。

关键词 安全多方计算,中国剩余定理,ElGamal 同态加密,安全多方求和计算,同余方程组,多秘密共享

0 引 言

保护隐私的同余方程组求解是安全多方计算(secure multi-party computation,SMC)领域中的新课题。SMC 是指在分布式网络中,多个用户在不泄露自己秘密的情况下,以自己的秘密作为输入共同计算某个函数[1]。SMC 拓展了传统密码学、分布式计算及信息安全的范围,为网络计算提供了一种新的计算模式。目前已得到研究的 SMC 问题有多种,本文基于对 Asmuth-Bloom 秘密共享方案中存在的问题的分析,研究了保护隐私的同余方程组求解,给出了保护隐私的同余方程组求解协议,并利用此协议设计了多秘密共享方案。

1 相关研究

Yao 在文献[2]中首次提出了安全两方计算协议。后来,Goldreich 对 SMC 做了比较完整的总结,并提出了 SMC 的安全性定义[3]。近几年来,安全多方计算取得了众多研究成果[4-18]。文献[4]构造了一种安全多方计算协议用于分享公共安全网络中的秘密信息。文献[5]利用秘密共享构造了一种安全多方排序协议,该协议可在一定程度上同时抵抗主动和被动敌手攻击。仲红等人在文献[6]中提出了安全多方计算中的一个特殊问题,即保护私有信息的最近点对问题,并基于半诚实模型和认证信道设计了向量差最小值协议,利用同态加密方案构建了一个求解保护隐私的最近点对协议。文献[7]利用安全多方计算技术,针对可嵌套共谋,提出了可信防共谋协议模型。2011 年,Maheshwari 和 Kiyawat 将安全多方计算和云计算结合,提出了安全多方云计算的概念、应用场景和框架结构[8]。

Goldreich, Mic-ali 及 Wigderson 提出了可以计算任意函数的基于密码学安全模型的安全多方计算协议[9]。理论上讲,任何安全多方计算问题都可以使用电路估值方案解决。但是,基于效率和复杂度的考虑,这种方案在实际应用中并不可行。针对不同的应用环境,需要研究不同的解决方案。

安全多方科学计算解决了保护隐私的科学计算问题,是安全多方计算的一个重要研究方向。表 1 对经典安全多方科学计算文献的创新点和不足进行了总结。

目前对安全多方科学计算的研究主要集中在线性代数问题的求解,却忽略了非线性代数领域的科学计算问题。例如,保护隐私的同余方程组的求解

① 863 计划(2009AA01Z430)和国家发改委信息安全专项资助项目。
② 女,1986 年生,博士生;研究方向:信息安全,安全多方计算;联系人,E-mail:starjingxiang@sina.com
(收稿日期:2011-11-28)

表 1 安全多方科学计算研究的创新点和不足

文献	创新点和不足
文献[10]	指出从信息论的角度看,点积协议不是无条件安全的。并在一定假设条件下,提出了安全点击协议。但是,协议的安全性基于存在安全第三方,但是实际上绝对安全的第三方是不存在的。
文献[11]	给出了线性方程组 $Ax = y(A \in F^{m\times n}, m \leqslant n)$ 的安全求解方案。但是方案计算复杂度为 $O(n^5)$,计算复杂度高。
文献[12]	降低了安全求解线性方程组的复杂度,复杂度降低到了 $O(m^4 + n^2m)$。但是,当 m 和 n 相差不大时,计算复杂度仍然很高。
文献[13]	解决了两方安全矩阵乘法问题,但是没有考虑多方时如何实现。
文献[14]	分别给出了安全求解矩阵运算、线性方程组、线性规划、最小平方等线性代数问题的两方求解方案。但是,只能实现安全两方求解,没有考虑多方时如何实现。

问题。同余方程组是密码学协议中常用的数学工具之一。利用同余方程组的求解,Asmuth 和 Bloom 设计了著名的 Asmuth-Bloom 秘密共享方案[15]。绕进平和冯登国利用同余方程组实现了一种提高 RSA 算法速度的方法,将 RSA 算法的计算速度提高了约 4 倍[16]。陈泽文等人利用同余方程组实现了一种群签名技术[17]。

近年来,计算机和互联网技术飞速发展。随着各种攻击手段、攻击工具的出现,敏感信息的保护引起人们的高度关注。为了保护敏感信息,现代密码技术被提出并得到了广泛应用。现代密码学体制基于 Kerchhoff 假设,一个秘密系统的安全性主要取决于密钥的安全性,与所采取的算法无关。因此,密钥的管理极为重要。一种传统的密钥管理方法是对密钥执行多次备份,但是,备份数量的增加会增加密钥泄露的风险;而备份数量少又会增加密钥丢失的风险。

为了解决上述问题,Shamir[18] 和 Blakley[19] 于 1979 年分别提出了秘密共享的概念。秘密共享是指秘密持有人将秘密拆分为多个子秘密,分发给多个秘密保管者;符合要求的秘密保管者可共同恢复出秘密,但是不符合要求的秘密保管者不能获得秘密的任何信息[18]。在著名的 (t, n) 秘密共享方案中,至少 t 个秘密保管者在一起时,才可以恢复出秘密。

1983 年,Asmuth 和 Bloom 基于同余方程组的求解提出了一个经典的 (t, n) 门限秘密共享方案[15]。但是,在 Asmuth-Bloom 秘密共享方案中,存在安全和存储空间占用较大两个问题。考虑如下实例:(1)假设公司 A 和公司 B 都希望竞拍某商业地皮,公司 A 的最高报价作为一个重要商业秘密需要使用 Asmuth-Bloom 秘密共享方案在 n 个产品经理中分享。公司 B 为了成功拍下该地皮,收买了公司 A 的产品经理 P_1。在秘密恢复阶段,产品经理提供虚假的子秘密 (a_1', m_1')。P_1 根据秘密恢复阶段其他产品经理提供的子秘密 (a_i, m_i),利用自己真实的子秘密 (a_1, m_1),计算同余方程组后可恢复出真实的报价。而对于其他参与者,由于其恢复报价时使用了 P_1 提供的虚假子秘密,因此其无法恢复出真实的报价。(2)假设某公司有 m 个商业秘密,每个秘密都包含一个用于商业运营的重要信息。该公司雇佣了 n 个员工,但是它不希望任意一个员工单独获取秘密。该公司规定,每一个秘密都需要根据一个特定的规则在员工中分享。该公司可以使用多个秘密分享方案来实现这些秘密的分享。但是如果这样,每个员工需要保存多个子秘密来参与每个秘密分享过程。也就是说,这种方案下子密钥将占用大量的存储空间。

为了解决上述两个问题,本文首先设计了保护隐私的同余方程组求解协议。该协议执行后,参与方仅可获知同余方程组的解,但无法获知其他参与者的方程参数。利用该协议,本文将 Asmuth-Bloom 秘密共享方案扩展为多秘密共享方案,解决 Asmuth-Bloom 秘密共享中存在的两个问题:(1)在秘密恢复阶段,恶意参与者无法收集到其他参与者的子秘密,因此无法独立恢复秘密;(2)通过保护子秘密的安全性,可实现一次共享多个秘密,节省存储空间。

半诚实模型是密码协议研究的一种重要模型。目前,在安全多方计算协议的研究中,几乎所有的协议都建立在半诚实模型下。同时,半诚实模型下的各种安全多方计算协议也具有很强的应用背景。本文亦在半诚实模型下进行讨论。

2 预备知识

2.1 半诚实模型

在安全多方计算协议的执行过程中,诚实参与者完全按照协议的要求完成协议的各个步骤,同时保密自己的所有输入、输出以及中间结果。半诚实参与者完全按照协议的要求完成协议的各个步骤,

不会中途强行退出或恶意掺入虚假数据,但是,他们可能会保留所有可以收集到的关于其他参与者的信息,以期望在协议结束后推断出对方的输入信息或泄漏给攻击者。如果所有参与者都是半诚实或诚实的,这样的模型称为半诚实模型[3]。

目前,在安全多方计算协议的研究中,几乎所有的协议都是建立在半诚实模型下。同时,半诚实模型下的各种安全多方计算协议也具有很强的应用背景。

2.2 安全性定义

定义 1(半诚实模型下多方计算的安全性) 设 $f:\{0,1\}_1^* \times \{0,1\}_2^* \times \cdots \times \{0,1\}_m^* \to \{0,1\}_1'^* \times \{0,1\}_2'^* \times \cdots \times \{0,1\}_m'^*$,其中,$f_i(x_1,x_2,\cdots,x_m)$ 为 $f(x_1,x_2,\cdots,x_m)$ 的第 i 个元素。定义 $f_I(x_1,x_2,\cdots,x_m) = \{f_{i_1}(x_1,x_2,\cdots,x_m),\cdots,f_{i_j}(x_1,x_2,\cdots,x_m)\}$,其中 $I=\{i_1,i_2,\cdots,i_j\}\subseteq\{1,2,\cdots,m\}$。设Π为计算 f 的 m 方协议,当输入为 $\bar{x}=(x_1,x_2,\cdots,x_m)$ 时,第 i 方在执行Π的过程中得到的信息序列为 $(x_i,r^i,m_1^i,m_2^i,\cdots,m_k^i)$,记为 $view_i^{\Pi}(\bar{x})$,其中 r^i 表示第 i 方独立的掷币输出;m_j^i 表示第 i 方收到第 j 次的信息。当输入为 $\bar{x}=(x_1,x_2,\cdots,x_m)$ 时,执行协议Π后,第 i 方的输出结果记为 $output_i^{\Pi}(\bar{x})$。对于一个函数 f,如果存在概率多项式时间算法(或者称为模拟器) S,使得对于所有的 $I\subseteq[m]$ 有 $\{S(I,(x_{i_1},x_{i_2},\cdots,x_{i_t}),f_1(\bar{x})),f(\bar{x})\}_{\bar{x}\in(\{0,1\}^*)^m} \overset{c}{\equiv} \{view_i^{\Pi}(\bar{x}),output^{\Pi}(\bar{x})\}_{\bar{x}\in(\{0,1\}^*)^m}$,则认为Π秘密地计算了 f,其中,$\overset{c}{\equiv}$ 表示多项式电路计算不可区分。要证明一个安全多方计算协议是安全的,就必须构造满足上述条件的模拟器 S。

该定义可以直观地理解为,对于一个半诚实参与者,如果可以直接利用自己的输入与协议的输出通过单独模拟整个协议的执行过程而得到在执行协议过程他所能得到的任何信息,那么协议就能保证输入的私密性。如果一个计算协议能被这样模拟,参与者就不能从协议的执行过程中得到有价值的信息,这样的多方计算就是安全的。

上述安全性定义由国际著名密码学家 Goldreich 在文献[3]中首次提出的,并被广泛应用于安全多方计算协议的安全性证明中。本文也将采用这种方法证明协议的安全性。

2.3 分布式 ElGamal 加密协议

1985 年,El Gamal 基于离散对数难题提出了分布式 ElGamal 加密协议[20],该协议满足乘同态性。

假设系统有 n 个参与者,每个参与者分别拥有秘密 $m_i(i=1,2,\cdots,n)$。$M=m_1m_2\cdots m_n$ 即要求得的明文。

系统参数:选择大素数 p,满足 Z_p 中离散对数问题是难解的。g 是 Z_p^* 的本原元。n 个参与者分别选择一个随机数 $x_i\in Z_p$,$i=1,2,\cdots,n$ 计算 $y_i=g^{x_i} \bmod p$ 并公布。

系统私钥:$x=\sum_{i=1}^{n} x_i \bmod p$。

系统公钥:$y=\prod_{i=1}^{n} y_i \bmod p = g^{\sum_{i=1}^{n} x_i} \bmod p$。

加密:各参与者 P_i 随机选择 $k_i\in Z_p$,计算 $E(m_i)=(g^{k_i} \bmod p, y^{k_i}m_i \bmod p)$ 并广播。各参与者共同计算 $E(M)=\prod_{i=1}^{n} E(m_i)=(\alpha,\beta)=(g^{\sum_{i=1}^{n} k_i} \bmod p, (y^{\sum_{i=1}^{n} k_i}\prod_{i=1}^{n} m_i) \bmod p)$。

解密:n 个参与者分别计算 α^{x_i} 并公布,然后共同计算出 $\prod_{i=1}^{n}\alpha^{x_i}$,最终得到 M:$M=\frac{\beta}{\alpha^x} \bmod p = \frac{\beta}{\alpha^{\sum_{i=1}^{n} x_i}} \bmod p = \frac{\beta}{\prod_{i=1}^{n}\alpha^{x_i}} \bmod p$。

2.4 中国剩余定理

设 $m_1,m_2,\cdots,m_r$ 是 $r\geqslant 2$ 个两两互素的且均大于 1 的整数,令 $M=m_1m_2\cdots m_r$,$M_i=M/m_i$,并且每个同余式 $M_iy\equiv 1(\bmod m_i)$ 有解 $y\equiv b_i(\bmod m_i)$,则同余式组

$$\begin{cases} x\equiv a_1(\bmod m_1) \\ x\equiv a_2(\bmod m_2) \\ \quad\vdots \\ x\equiv a_r(\bmod m_r) \end{cases} \tag{1}$$

的解是 $x\equiv \sum_{i=1}^{r} M_ib_ia_i(\bmod M)$[21]。

2.5 Asmuth-Bloom 秘密共享方案

Asmuth-Bloom 秘密共享方案由秘密分发和秘密恢复两个阶段组成,其执行步骤如下:

秘密分发阶段:

步骤 1:构造 Asmuth-Bloom 序列 $\{m_1,m_2,\cdots,m_n,p\}$,满足如下条件:

(1) $m_1<m_2<m_n<p$;

(2) $gcd(m_i,m_j)=1$,$i\neq j$;

(3) $m_1\times m_2\times m_t > p\times m_{n-t+2}\times\cdots\times m_n$。

步骤 2:构造秘密影子。对于秘密 $m(m<p)$,首先随机选择 r,计算 $M=m+rp$。秘密影子 a_i 满足 $a_i\equiv M(\bmod m_i)$。

步骤 3：秘密发布者将 (a_i, m_i) 通过安全信道发送给参与者 P_i，并广播 p 的值。

秘密恢复阶段：

步骤 1：t 个参与者计算下列方程组的解 M：

$$\begin{cases} M \equiv a_1 (\mathrm{mod} m_1) \\ M \equiv a_2 (\mathrm{mod} m_2) \\ \quad \vdots \\ M \equiv a_t (\mathrm{mod} m_t) \end{cases}$$

步骤 2：利用 $m \equiv M (\mathrm{mod}\ p)$ 且 $m < p$ 求得秘密 m。

3 保护隐私的同余方程组求解协议

3.1 问题描述

假设有 n 个参与者 $P_1, P_2, \cdots, P_n$，$(n > 2)$，分别拥有保密数据 $a_1, a_2, \cdots, a_n$ 和 $m_1, m_2, \cdots, m_n$。其中，$m_1, m_2, \cdots, m_n$ 是两两互素的大于 1 的整数。现在，n 个参与者希望在不泄露自己的保密数据的情况下，共同计算下列同余方程组：

$$\begin{cases} s \equiv a_1 (\mathrm{mod} m_1) \\ s \equiv a_2 (\mathrm{mod} m_2) \\ \quad \vdots \\ s \equiv a_n (\mathrm{mod} m_n) \end{cases} \tag{2}$$

该问题称为保护隐私的同余方程组求解问题。

3.2 实现原理

根据中国剩余定理，同余方程组(2)的解为 $s \equiv \sum_{i=1}^{n} M_i b_i a_i (\mathrm{mod} M)$。其中，$M = m_1 m_2 \cdots m_n$，$M_i = M/m_i$，并且每个同余式 $M_i y \equiv 1 (\mathrm{mod} m_i)$ 有解 $y \equiv b_i (\mathrm{mod} m_i)$。

本文设计的协议首先利用 ElGamal 同态加密协议计算出 M 和 M_i，然后利用安全多方求和计算出 $\sum_{i=1}^{n} M_i b_i a_i$，再进行模 M 运算，即得到方程组的解。

3.3 协议描述

系统参数：大素数 p，满足 Z_p 中离散对数问题是难解的。g 是 Z_p^* 的本原元。系统私钥：$x = \sum_{i=1}^{n} x_i \mathrm{mod} p$。系统公钥：$y = \prod_{i=1}^{n} y_i \mathrm{mod} p = g^{\sum_{i=0}^{n} x_i} \mathrm{mod} p$。

协议输入：参与者 $P_1, P_2, \cdots, P_n$ 分别输入隐私数据 $a_1, a_2, \cdots, a_n$ 和 $m_1, m_2, \cdots, m_n$，以及各自的子私钥 $x_1, x_2, \cdots, x_n$。

中间变量： $k = \sum_{i=1}^{n} k_i$，$k_i = M_i b_i a_i$。

协议输出：一次同余方程组的解 x。

协议执行过程：

步骤 1：利用分布式 ElGamal 同态加密协议计算出 M，进而计算出 M_i：

a) n 个参与者分别选择随机数 $k_i \in Z_p$，利用系统公钥 y 加密隐私数据 m_i，得到 $E(m_i) = (g^{k_i} \mathrm{mod} p, y^{k_i} m_i \mathrm{mod} p)$ 并广播。

b) 各参与者共同计算 $E(M) = \prod_{i=1}^{n} E(m_i) = (\alpha, \beta) = (g^{\sum_{i=1}^{n} k_i} \mathrm{mod} p, (y^{\sum_{i=1}^{n} k_i} \prod_{i=1}^{n} m_i) \mathrm{mod} p)$。

c) n 个参与者分别计算 α^{x_i} 并公布，然后共同计算出 $\prod_{i=1}^{n} \alpha^{x_i}$，从而得到 $M = \frac{\beta}{\alpha^x} \mathrm{mod} p = \frac{\beta}{\alpha^{\sum_{i=1}^{n} x_i}} \mathrm{mod} p = \frac{\beta}{\prod_{i=1}^{n} \alpha^{x_i}} \mathrm{mod} p$。

d) 各参与者分别计算 $M_i = M/m_i$。

步骤 2：对于 $i = 1, 2, \cdots, n$，利用 $M_i y \equiv 1 (\mathrm{mod} m_i)$ 有解 $y \equiv b_i (\mathrm{mod} m_i)$，参与者 P_i 计算出 b_i。

步骤 3：每个参与者计算出中间变量 $k_i = M_i b_i a_i$。

步骤 4：利用安全多方求和计算 k 的值：

a) 每个参与者 P_i 产生 k_i 的 n 个随机份额 $k_{i,j}$，其中，$j = 1, 2, \cdots, n$，$k_i = \sum_{i=1}^{n} k_{i,j}$。

b) 对于 $i = 1, 2, \cdots, n$，$j = 1, 2, \cdots, n$，$i \neq j$，参与者 P_i 将 $k_{i,j}$ 发送给 P_j。

c) 对于 $i = 1, 2, \cdots, n$，参与者 P_i 计算 $k_i' = \sum_{j=1}^{n} k_{j,i}$，并广播计算结果。

d) 各参与者计算 $k = \sum_{i=1}^{n} k_i'$。

步骤 5：各参与者计算 $s \equiv k (\mathrm{mod} M)$，即一次方程组的解。

3.4 性能分析

3.4.1 正确性分析

定理 3.4.1 在半诚实模型下，保护隐私的同余方程组计算协议是正确的。

证明：步骤 1－3 的正确性易知，证明略。

因为 $\sum_{i=1}^{n} k_i' = \sum_{i=1}^{n} \sum_{j=1}^{n} k_{j,i} = \sum_{j=1}^{n} \sum_{i=1}^{n} k_{j,i} = \sum_{j=1}^{n} k_j = k$，又 $k = \sum_{i=1}^{n} k_i = \sum_{i=1}^{n} M_i b_i a_i$，所以 $s \equiv k (\mathrm{mod} M)$，即 $s \equiv \sum_{i=1}^{n} M_i b_i a_i (\mathrm{mod} M)$。证毕。

3.4.2 安全性分析

定理 3.4.2 在半诚实模型下,保护隐私的同余方程组求解协议是安全的。

证明:下面使用 Goldreich 在文献[3]中提出的安全多方计算的安全性定义,以 3 个参与者的情况为例进行证明。

假设有 3 个参与者 P_1,P_2,P_3,分别拥有保密数据 a_1,a_2,a_3 和 m_1,m_2,m_3。现在这 3 个参与者通过保护隐私的同余方程组求解协议共同计算下列同余方程组:

$$\begin{cases} s \equiv a_1 (\mathrm{mod} m_1) \\ s \equiv a_2 (\mathrm{mod} m_2) \\ s \equiv a_3 (\mathrm{mod} m_3) \end{cases}$$

定义协议的输出结果为 X。由定理 3.4.1 知

$$\begin{aligned} & f_1(a_1,a_2,a_3,m_1,m_2,m_3) \\ &= f_2(a_1,a_2,a_3,m_1,m_2,m_3) \\ &= f_3(a_1,a_2,a_3,m_1,m_2,m_3) \\ &= output^{\Pi}(a_1,a_2,a_3,m_1,m_2,m_3) \\ &= X \end{aligned}$$

构造一个模拟器 S 模拟 P_2 的 $view_2^{\Pi}(a_1,a_2,a_3,m_1,m_2,m_3)$。模拟过程如下:

(1) S 接受 $((a_2,m_2), f_2(a_1,a_2,a_3,m_1,m_2,m_3))$ 作为 P_2 的输入,根据 $f_2(a_1,a_2,a_3,m_1,m_2,m_3)$ 的值确定 (a_1', a_3', m_1', m_3'),使得式 $f_2(a_1,a_2,a_3,m_1,m_2,m_3) = f_2(a_1', a_2, a_3', m_1', m_2, m_3')$ 成立。

(2) 模拟器 S 模拟,$M' = m_1'm_2m_3'$,$M_1' = M'/m_1'$,$M_3' = M'/m_3'$,求得 b_1',b_2'。根据 $k_{si} = M_i'b_i'a_i'$,得到 k_{s1} 和 k_{s3}。

(3) 模拟器 S 根据 $k_s = k_2 + k_{s1} + k_{s3}$ 得到 k_s,验证 $X \equiv k_s (\mathrm{mod} M')$。如果不满足验证条件,则调整 (a_1', a_3', m_1', m_3') 的值。模拟器 S 总是可以达到 (a_1', a_3', m_1', m_3') 的值,使得验证条件成立。模拟器可根据 k_{s1}, k_2, k_{s3} 的值构造 $k_{1,2}', k_{3,2}'$,使得 $k_s = k_{1,2}' + k_{3,2}' + k_{2,2}$。

由上面分析可得到,模拟器 S_1 的输出为 $S((a_1, a_2, a_3, m_1, m_2, m_3), f_1(a_1, a_2, a_3, m_1, m_2, m_3)) = \{M', k_s, k_{1,2}', k_{3,2}', k_{s1}, k_{s3}, E_{pk}(m_3')\}$。

又因为 $view_2^{\Pi}(a_1,a_2,a_3,m_1,m_2,m_3) = \{M,k, k_{1,2},k_{2,3},k_1,k_3,E_{pk}(m_3)\}$,由模拟器的模拟过程知:$S((a_1,a_2,a_3,m_1,m_2,m_3), f_1(a_1,a_2,a_3,m_1,m_2,m_3)) \equiv view_2^{\Pi}(a_1,a_2,a_3,m_1,m_2,m_3)$,那么有 $\{S(1,2,3), (a_1,a_2,a_3,m_1,m_2,m_3), f_1(a_1,a_2,a_3,m_1,m_2,m_3), f(a_1,a_2,a_3,m_1,m_2,m_3)\} \equiv \{view_2^{\Pi}(a_1,a_2,a_3,m_1,m_2,m_3), output^{\Pi}(a_1,a_2,a_3,m_1,m_2,m_3)\}$。

类似地,还可以针对参与者 P_1, P_2 构造模拟器。

根据半诚实者行为的安全性的定义,保护隐私的同余方程组求解协议是安全的。

证毕。

3.4.3 复杂性分析

计算复杂度:在协议的第 1 步中,n 方利用 ElGamal 加密协议计算 M, m_i 时,进行了 $3n$ 次模指运算、$(4n+1)$ 次模乘运算和 n 次数乘运算。在协议的第 2 步至第 5 步中,进行了 n 次模乘运算、$2n$ 次数乘运算和 $2n(n-1)+n-1$ 次加法运算。使用 T_{me} 代表模指运算,T_{mm} 代表模乘运算,T_{dm} 代表数乘运算,T_{da} 代表加法运算,则该协议的计算复杂度为 $3nT_{me} + (5n+1)T_{mm} + 3nT_{dm} + (2n+1)(n-1)T_{da}$。

通信复杂度:在协议的第 1 步中通信 $2n$ 次,第 2 步至第 5 步中,通信 $(n-1)n+n$ 次。该协议共通信 n^2+2n 次。

3.5 举例

假设有 3 个参与者 P_1,P_2,P_3,分别拥有保密数据 $a_1 = 2$, $a_2 = 3$, $a_3 = 2$ 和 $m_1 = 3$, $m_2 = 5$, $m_3 = 7$。现在,这 3 个参与者出于某种需要,比如为了完成某个密码学协议,希望在不泄露自己的保密数据的情况下,共同计算下列同余方程组:

$$\begin{cases} s \equiv 2 (\mathrm{mod} 3) \\ s \equiv 3 (\mathrm{mod} 5) \\ s \equiv 2 (\mathrm{mod} 7) \end{cases} \tag{3}$$

协议计算过程如下:

步骤 1:利用分布式 ElGamal 同态加密方案计算 M,进而计算出 M_i:

a) 3 个参与者分别选择随机数 $k_i \in Z_p$,利用系统公钥 y 加密隐私数据 m_i,得到 $E(m_1)$、$E(m_2)$ 及 $E(m_3)$ 并广播。$E(m_1) = (g^{k_1} \mathrm{mod} p, 3y^{k_1} \mathrm{mod} p)$, $E(m_2) = (g^{k_1} \mathrm{mod} p, 5y^{k_1} \mathrm{mod} p)$, $E(m_3) = (g^{k_1} \mathrm{mod} p, 7y^{k_1} \mathrm{mod} p)$。

b) 3 个参与者共同计算 $E(M) = \prod_{i=1}^{3} E(m_i) = (\alpha, \beta) = (g^{\sum_{i=1}^{3} k_i} \mathrm{mod} p, 105y^{\sum_{i=1}^{3} k_i} \mathrm{mod} p)$。

c) 3 个参与者分别计算 α^{x_i} 并公布,然后共同计算出 $\prod_{i=1}^{n} \alpha^{x_i}$, $M = \frac{\beta}{\alpha^x} \mathrm{mod} p = \frac{\beta}{\prod_{i=1}^{n} \alpha^{x_i}} \mathrm{mod} p = 105$。

d) 参与者 P_1 计算 $M_1 = \frac{M}{m_1} = 35$;参与者 P_2 计

算 $M_2 = \frac{M}{m_2} = 21$；参与者 P_3 计算 $M_3 = \frac{M}{m_3} = 15$。

步骤 2：利用 $M_i y \equiv 1(\mathrm{mod}m_i)$ 有解 $y \equiv b_i(\mathrm{mod}m_i)$，参与者 P_1 计算出 $b_1 = 2$，参与者 P_2 计算出 $b_2 = 1$，参与者 P_3 计算出 $b_3 = 1$。

步骤 3：P_1 计算 $k_1 = M_1 b_1 a_1 = 35 \times 2 \times 2 = 140$，$P_2$ 计算 $k_2 = M_2 b_2 a_2 = 21 \times 1 \times 3 = 63$，$P_3$ 计算 $k_3 = M_3 b_3 a_3 = 15 \times 1 \times 2 = 30$。

步骤 4：利用安全多方求和计算 k 的值：

a) P_1 产生随机份额 $k_{1,1} = 100$，$k_{1,2} = 20$，$k_{1,3} = 20$，并将 $k_{1,2}$ 发送给 P_2，$k_{1,3}$ 发送给 P_3；P_2 产生随机份额 $k_{2,1} = 3$，$k_{2,2} = 30$，$k_{2,3} = 30$，并将 $k_{2,1}$ 发送给 P_1，$k_{2,3}$ 发送给 P_3；P_3 产生随机份额 $k_{3,1} = 10$，$k_{3,2} = 10$，$k_{3,3} = 10$，并将 $k_{3,1}$ 发送给 P_1，$k_{3,2}$ 发送给 P_2。

b) P_1 计算 $k_1' = \sum_{j=1}^{n} k_{j,1} = 100 + 3 + 10 = 113$，并广播计算结果；$P_2$ 计算 $k_2' = \sum_{j=1}^{n} k_{j,2} = 20 + 30 + 10 = 60$，并广播计算结果；$P_3$ 计算 $k_3' = \sum_{j=1}^{n} k_{j,3} = 20 + 30 + 10 = 60$，并广播计算结果。

d) 各参与者计算 $k = \sum_{i=1}^{3} k_i' = 113 + 60 + 60 = 233$。

步骤 5：各参与者计算得一次同余方程组的解为 $s \equiv 233(\mathrm{mod}105)$。由于 $s < M$，得 $s = 23$。

4 基于保护隐私同余方程组求解协议的多秘密共享方案

本节将保护隐私同余方程组求解协议应用到 Asmuth-Bloom 秘密共享方案中，实现了 $t-n$ 多秘密共享方案。

4.1 问题描述

假设某公司有 k 个秘密 $\{s_1, s_2, \cdots, s_k\}$，需要在 n 个员工中共享，该公司要求通过一次秘密共享过程完成上述需求。该问题被称为多秘密共享问题。

4.2 多秘密共享协议

4.2.1 秘密分发过程

步骤 1：构造 Asmuth-Bloom 序列：

构造 Asmuth-Bloom 序列 $\{m_1, m_2, \cdots, m_n, p\}$，满足如下条件：

(1) $max(s_1, s_2, \cdots, s_k) < p$；

(2) $m_1 < m_2 < \cdots < m_n < p$；

(3) $gcd(m_i, m_j) = 1$，$i \neq j$；

(4) $m_1 \times m_2 \times \cdots \times m_t > p \times m_{n-t+2} \times m_{n-t+3} \times \cdots \times m_n$。

步骤 2：构造秘密影子：

秘密发布者选择随机数 r。对于每个秘密 P_i，秘密发布者计算 $S_i = s_i + rp$，然后计算参与者的秘密影子 $a_{i,j} \equiv S_i(\mathrm{mod}m_i)$，其中 $1 \leqslant i \leqslant k$，$1 \leqslant j \leqslant n$。

步骤 3：秘密发布者通过秘密信道将 $((a_{1,i}, a_{2,i}, \cdots, a_{k,i}), m_i)$ 发送给参与者 P_i，其中 $1 \leqslant i \leqslant k$。并公布 p 的值。

4.2.2 秘密恢复过程

当需要恢复第 i 个秘密时，至少需要 t 个参与者使用它们的子秘密 $(a_{i,j}, m_i)$ 按照如下步骤完成秘密恢复过程：(假设此时有 t 个参与者，则 $1 \leqslant j \leqslant t$)

步骤 1：t 个参与者利用保护隐私的同余方程组求解协议计算下列方程组的解：

$$\begin{cases} M \equiv a_{i,1}(\mathrm{mod}m_1) \\ M \equiv a_{i,2}(\mathrm{mod}m_2) \\ \quad\vdots \\ M \equiv a_{i,t}(\mathrm{mod}m_t) \end{cases}$$

步骤 2：利用 $s_i \equiv M(\mathrm{mod}\ p)$，$s_i < p$，即可恢复出秘密 s_i。

4.3 性能分析

(1) 该方案的正确性依赖于 Asmuth-Bloom 秘密共享方案和保护隐私的同余方程组求解协议的正确性。

(2) 该方案的安全性依赖于 Asmuth-Bloom 秘密共享方案和保护隐私的同余方程组求解协议的安全性。且每次恢复出一个秘密后，由于每个参与者 P_i 的信息 m_i 是保密的，其他参与者无法获知，因此可以保证其他秘密的安全性。即使参与者 P_i 提供了虚假子秘密，他也无法恢复出真实的秘密。

(3) 在复杂性方面，该方案调用 Asmuth-Bloom 秘密共享一次，每恢复一次秘密调用保护隐私的同余方程组求解协议一次。因此，该方案的通信复杂度为 $O(kn^2)$。使用 Asmuth-Bloom 方案共享 k 个秘密时，通信复杂度为 $O(kn^2)$。因此，本方案和 Asmuth-Bloom 秘密共享方案的通信复杂度一样。

(4) 由构造 Asmuth-Bloom 序列时的条件(4)，可保证少于 t 个参与者不能恢复出秘密，其原理和 Asmuth-Bloom 秘密共享方案相同。

(5) 为了实现共享 k 个秘密，可以使用 k 次 As-

muth-Bloom 秘密共享方案。此时,秘密分发者需要构造 $k \times n$ 对子秘密 $(a_{i,j}, m_{i,j})$,每个参与者 P_i 需要保存 k 对子秘密 $(a_{t,i}, m_{t,i})$。如果使用本文提出的多秘密共享方案,秘密分发者仅需要根据 n 个 m_i 值,构造 $k \times n$ 个 $a_{i,j}$ 值;每个参与者 P_i 仅需要保存 k 个 $a_{t,i}$ 值和 1 个 m_i 值。由此可见,当需要共享的秘密较多时,本文提出的多秘密共享方案可有效节省秘密分发者和参与者的子秘密存储空间。

假设 $a_{i,j}$, $m_{i,j}$ 分别占用一个存储单元,表 2 对比了 Asmuth-Bloom 方案和本文提出的方案所使用的存储空间大小。由该表可知,本文提出的方案比 Asmuth-Bloom 方案节省约一半的存储空间。

表 2 存储空间比较

共享秘密个数/参与者个数	Asmuth-Bloom 方案		本文的方案	
	分发者存储空间	参与者存储空间	分发者存储空间	参与者存储空间
100/100	20000	20000	10000	10100
1000/100	200000	200000	100000	100100
1000/1000	2000000	2000000	1000000	1001000

5 结 论

本文分析了 Asmuth-Bloom 秘密共享方案中存在的问题,提出了保护隐私的同余方程组求解问题,将非线性方程组的求解引入到安全多方科学计算中。在半诚实模型下给出了保护隐私的同余方程组求解协议,并详细分析了协议的正确性、安全性和复杂性,通过具体实例说明了该方案的实用性,最后利用该协议设计了多秘密共享方案,解决了 Asmuth-Bloom 秘密共享方案中存在的安全、存储空间问题。本文提出的协议中使用了 ElGamal 同态加密和安全多方求和两种安全多方计算的基础协议,协议过程中交互次数较多,效率不高。在以后的工作中,我们希望使用一种基础协议完成整个协议交互过程,例如结合 Gentry 在 2009 年提出的全同态加密方案[22],并进一步提高协议的效率。另外,要利用本文提出的保护隐私的同余方程组求解协议,推广现有的使用同余方程组的密码学协议。通过加入验证算法将本协议推广到恶意模型,也是今后值得研究的方向。

参考文献

[1] Goldwasser S. Multi-party computations: past and present. In: Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing, Santa Barbarba, USA, 1997. 21-24

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

[3] Goldreich O. Secure Multi-party Computation. http://ww- w. wisdom. weizman. ac. il/ ~oded/pp. html, 1998

[4] Santos M A S, Margi C B. A secure multi-party protocol for sharing valuable information in public safety networks. In: Proceedings of the IEEE 8th International Conference on Mobile Adhoc and Sensor Systems, Valencia. 2011. 935-940

[5] Tang C M, Shi G H, Yao Z A. Secure multi-party computation protocol for sequencing problem. *Science China Information Sciences*, 2011, 54(8):1654-1662

[6] 仲红,孙彦飞,燕飞飞等. 保护私有信息的空间最近点对协议. 计算机工程与应用. 2011,48(4):87-89

[7] 程柏良,曾国荪,揭安全. 基于安全多方计算的可信防共谋协议模型. 通信学报. 2011,32(8):23-30

[8] Maheshwari N, Kiyawat K. Structural framing of protocol for secure multiparty cloud computation. In: Proceedings of the 5th Asia Modelling Symposium, Kuala Lumpur, Malaysia, 2011. 187-192

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

[10] Thomas B P, Erkay S. Impossibility of unconditionally secure scalar products. *Knowledge Engineering*, 2009, 68(10):1059-1070

[11] Cramer R, Damgard I. Secure distributed linear algebra in a constant number of rounds. In: Proceedings of Advances in Cryptology-CRYPTO 2001. LNCS (2139), Springer -Verlag, 2001. 119-136

[12] Cramer R, Kiltz E, Padró C. A note on secure computation of the Moore-Penrose pseudoinverse and its application to secure linear algebra. In: Proceedings of Advances in Cryptology-CRYPTO 2007. LNCS (4622), Springer-Verlag, 2007. 613-630

[13] Mohassel P , Weinreb E. Efficient secure linear algebra in the presence of covert or computationally unbounded adversaries. In: Proceedings of Advances in Cryptology - C-RYPTO 2008. LNCS(5157), Springer-Verlag. 2008. 481-496

[14] Du W L. A Study of Several Specific Secure Two-Party Computation Problems: [Ph. D dissertation]. USA: Purdue University, 2001

[15] Asmuth C, Bloom J. A modular approach to key safe-

guarding. *IEEE Transactions on Information Theory*, 1983, 29(2):208-210

[16] 绕进平,冯登国. 一种高效率的 RSA 模幂算法的研究.计算机工程与应用, 2003, 39(9): 76-77

[17] 陈泽文,张龙军,王育民等. 一种基于中国剩余定理的群签名方案,电子学报, 2004, 32(7): 1062-1065

[18] Shamir A. How to share a secret. *Communication of the association for computing machinery*, 1979, 22(11): 612-613

[19] Blakley G R. Safe guarding cryptographic keys. In: Proceedings of the National Computer Conference, New York, USA, 1979. 313-317

[20] El Gamal T. A public key cryptosysterm and a signature scheme based on discrete logarithms. *IEEE Transanction on Information Theory*, 1985,31(4):469-472

[21] 阮传概,孙伟. 近世代数及其应用.北京:北京邮电大学出版社,2001.258-268

[22] Gentry C. Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41th annual ACM symposium on Theory of Computing, Bephesda, USA, 2009. 169-178

A multi-secret sharing scheme based on a protocol for computation of simultaneous congruences for privacy-preserving

Sun Maohua[*,**], Li Han[*,**], Luo Shoushan[*,**,***], Xin Yang[*,**]

([*] Information Security Center, Beijing University of Posts and Telecommunications, Beijing 100876)

([**] National Engineering Laboratory for Disaster Backup and Recovery, Beijing 100876)

([***] Beijing Safe-Code Technology Co., Ltd., Beijing 100876)

Abstract

Aiming at the problems of security, communication complexity and storage space in the Asmuth-Bloom secret sharing scheme, the problem of solving simultaneous congruences for privacy-preserving was studied. This problem is a new topic in the field of secure multi-party computation (SMC). A protocol for computation of simultaneous congruences for privacy-preserving was designed under the semi-honest model based on the Chinese remainder theorem, the ElGamal homomorphic encryption protocol and the secure multi-party sum computation. The correctness, security and complexity of the protocol were analyzed, and then a multi-secret sharing scheme was proposed based on the new protocol. It was proved that the new multi-secret sharing scheme can solve the above mentioned problems in the Asmuth-Bloom secret sharing scheme.

Key words: secure multi-party computation, Chinese remainder theorem, ElGamal homomorphic encryption, secure multi-party sum computation, simultaneous congruences, multi-secret sharing