论文

基于拉格朗日酉算子的量子安全多方求积

张龙威¹, 宋秀丽^{1*}, 李闯², 刘羽¹

1. 重庆邮电大学网络空间安全与信息法学院, 重庆 400065;

2. 重庆邮电大学计算机科学技术学院, 重庆 400065

*联系人, E-mail: songxl@cqupt.edu.cn

收稿日期: 2021-08-22; 接受日期: 2021-11-05; 网络出版日期: 2022-02-18

国家自然科学基金(编号: 61772098)、重庆邮电大学博士启动基金(编号: A2020211)和重庆市研究生科研创新项目(编号: CYS21335)资助

摘要 现有的量子安全多方求积协议大多数利用纠缠态携带秘密信息, 然而制备这些量子态难度较大. 为了解决这个问题, 本文提出一种基于 d 维拉格朗日酉算子的单粒子量子安全多方求积协议. 在该协议中, 半可信的第三方和每个参与者共享一个私有密钥, 参与者首先将共享密钥、秘密整数和随机数模 d 相乘, 接着使用拉格朗日酉算子将计算的结果编码到无偏基粒子上, 第三方测量每一个参与者发送回来的粒子获得编码信息, 然后使用所有随机数的模 d 乘积计算多方参与者秘密整数的求积. 性能分析表明, 该协议与其他相似协议相比, 对实体的量子计算能力要求更低. 在安全性方面, 该协议能够同时抵抗提出的外部攻击和内部攻击. 最后, 协议使用 $Q\#$ 量子编程语言结合Python仿真验证了本协议的正确性.

关键词 量子安全多方乘法, d 维拉格朗日酉算子, 无偏基粒子, $Q\#$ 量子编程语言

PACS: 03.67.Dd, 03.67.Hk, 03.67.Lx

1 引言

经典安全多方计算^[1-3]是研究多个参与者在互不泄露自己私密信息的基础上协同计算的问题, 该研究一经提出就引起了关注^[4,5]. 经典安全多方加法和乘法计算作为更复杂的经典安全多方计算协议的基础, 被广泛应用于多个领域, 如多方匿名投票、隐私拍卖和数据挖掘等. 随着量子算法和量子计算机的发展, 经典安全多方计算协议的安全性受到了巨大的挑战. 量子安全多方计算(Quantum Secure Multiparty Computation, QSMC)作为量子密码的重要组成部分, 弥补了经

典安全多方计算在量子计算平台上的安全缺陷, 于是越来越多的研究者将目光转向QSMC. QSMC应用量子力学的基本原理^[6], 如量子纠缠^[7]、Heisenberg测不准原理^[8]等, 结合量子密码和量子计算的优点, 允许多个参与者安全地计算一些公共函数, 并且保证每一个参与者的隐私输入不被窃取, 为我们提供了一种基于量子通信技术^[9]的安全多方计算方法. 现如今, 对QSMC的研究如火如荼, 量子安全多方求积(Quantum Secure Multiparty Multiplication, QSMM)协议的应用也趋于广泛, 如量子密钥协商、量子隐私比较和量子多方匿名投票等. 可以看出, QSMM在保证人们安全

引用格式: 张龙威, 宋秀丽, 李闯, 等. 基于拉格朗日酉算子的量子安全多方求积. 中国科学: 物理学 力学 天文学, 2022, 52: 260311
Zhang L W, Song X L, Li C, et al. Quantum secure multiparty multiplication based on Lagrange unitary operator (in Chinese). Sci Sin-Phys Mech Astron, 2022, 52: 260311, doi: 10.1360/SSPMA-2021-0234

通信中起到的作用, 越来越不容忽视.

现有的QSMC协议大多是量子安全多方求和(Quantum Secure Multiparty Summation, QSMS)协议, QSMM协议较少. 2007年, Vaccaro等人^[10]在量子匿名投票和调查的方案中提出QSMS协议. 同年, Du等人^[11]提出一种基于非正交态的模 $n+1$ ($n \geq 2$)的QSMS协议. 2010年, Chen等人^[12]提出了一种基于二维多粒子GHZ纠缠态的QSMS协议. 2015年, Zhang等人^[13]利用真正最大纠缠6粒子态提出了一种允许三个参与者共同求和的QSMS协议. 2017年, Zhang等人^[14]提出了一种使用单光子传递加密密钥, 之后通过加密密钥加密计算数据再进行多方求和的QSMS协议. 2020年, Duan^[15]提出了一种在 d 维量子系统中利用qudit移位操作实现求和的QSMS协议, 该协议能计算所有参与者随机输入数的和. 同年, Song等人^[16]提出一种基于 d 维拉格朗日酉算子和Shamir秘密共享的QSMS协议, 其利用拉格朗日酉算子对环形传输的粒子进行酉操作实现加法计算. 近期, 叶天语和胡家莉^[17]利用相互无偏基^[18]实现有限域 $GF(d)$ 上整数加整数模 d 求和的QSMS协议, 并且给出了该协议的应用场景.

对于QSMM协议的相关研究, 直到2016年, Shi等人^[19]提出一种基于量子傅里叶变换(Quantum Fourier Transform, QFT), CNOT门和逆傅里叶变换(Inverse Quantum Fourier Transform, IQFT)的QSMM协议, 在该协议中发起者对多个粒子执行QFT和CNOT操作制备纠缠态, 参与者通过执行Oracle算子将计算数据转换为相应的相位信息实现多方求积. 2019年, Lv等人^[20]设计了一种基于无偏基的QSMM协议, 在该协议中, Server在一组无偏基中选取若干个粒子组成信息粒子序列并在其中插入诱骗粒子, 通过在多个参与者之间环形传输, 参与者对信息粒子执行酉操作并更新诱骗粒子, 最终实现多方求积. 针对以上协议需要所有参与者共同参与求积, Sutradhar和Om^[21]在2020年提出一种基于Shamir秘密共享、QFT和SUM门的 (t, n) 门限多方求积协议, 协议中参与者首先对秘密整数产生的多个秘密份额进行计算, 接着发起者 P_1 对 t 个粒子执行QFT和 $t-1$ 次SUM门操作生成多粒子纠缠态, 然后将 $t-1$ 个粒子分别发送给 $t-1$ 个可信任的参与者, P_1 和 $t-1$ 个参与者通过对手中的纠缠态粒子执行QFT和Pauli变换等操作实现求积. 2021年, 他们对文献^[21]中的QSMM协议进行改进^[22], 改进的协议仍是基于Sha-

mir秘密共享和QFT的 (t, n) 门限多方求积, 不同的是协议中改用环形粒子传输模式传输纠缠态的信息粒子, 参与者通过对纠缠粒子执行Oracle算子, 利用量子纠缠的基本原理实现多方乘法, 相比于文献^[21], 改进的协议在执行QFT的次数上进行了缩减.

通过分析现有的QSMM协议发现, 文献^[19,20]的计算复杂度较大, 文献^[20-22]对参与者的粒子制备能力或操作能力要求较高, 文献^[20]在安全性方面不能抵抗少于 $n-1$ ($n > 3$)个参与者的合谋攻击. 针对现有QSMM协议的上述不足, 本文基于拉格朗日酉算子提出了一种新的QSMM协议, 协议使用无偏基态的单粒子作为信息载体, 每个参与者只需对收到的信息粒子执行一次酉操作. 性能分析表明, 协议在兼顾安全性的基础上, 总体性能表现良好. 协议的最后, 通过Microsoft公司的Q#量子编程语言集成经典编程语言Python仿真实现.

2 预备知识

针对本协议研究的内容, 本节将给出协议中需要使用到的基础知识, 包括两个 d 维相互无偏基和 d 维拉格朗日酉算子的定义.

2.1 d 维相互无偏基

在 d 维量子系统中, 两个标准正交基 $\{|\phi_l\rangle\}_{l=0}^{d-1}$ 和

$\{|\psi_j\rangle\}_{j=0}^{d-1}$, 如果满足

$$|\langle \phi_l | \psi_j \rangle| = \frac{1}{\sqrt{d}}, \quad l, j = 0, 1, \dots, d-1, \quad (1)$$

则称这两个基是相互无偏的^[23].

两个无偏基可以写作如下形式^[24]:

$$\begin{aligned} |\phi_l\rangle &= |l\rangle, \\ |\psi_j\rangle &= \frac{1}{\sqrt{d}} \sum_{u=0}^{d-1} e^{\frac{2\pi i}{d} u j} |u\rangle, \end{aligned} \quad (2)$$

其中, $l, j = 0, 1, \dots, d-1$.

2.2 d 维拉格朗日酉算子

假设 q^0 是一个 $n \times n$ 的单位矩阵, 集合 $\{q^0, q, q^2, q^3, \dots, q^{p-1}\}$ 是一个有限循环矩阵群, 其中整数 p 是该矩

阵群的阶, 此集合的子集合 $\{q, q^2, q^3, \dots, q^{p-1}\}$ 中任何一个矩阵都不等于单位矩阵, 通过拉格朗日插值多项式^[25], 构造如下高维的拉格朗日酉算子:

$$m(\theta) = \sum_{j,k=0}^{p-1} \frac{\prod_{k \neq j} (e^{i\theta} - \omega^k)}{\prod_{k \neq j} (\omega^j - \omega^k)} q^j, \quad (3)$$

其中, $\omega = e^{2\pi i/p}$. 特别地, 当 $p=4$ 时, 可以得到 4 维的拉格朗日酉算子为

$$m(\theta) = \frac{1}{4}[(e^{2i\theta} + 1)(e^{i\theta} + 1)q^0 + (1 - e^{2i\theta})(ie^{i\theta} - 1)q^1 + (e^{2i\theta} + 1)(1 - e^{i\theta})q^2 + (1 - e^{2i\theta})(ie^{i\theta} + 1)q^3]. \quad (4)$$

此时循环矩阵集合 $\{q^0, q, q^2, q^3\}$ 为如下形式:

$$q = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}, q^2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad (5)$$

$$q^3 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, q^0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = q^4.$$

将上面矩阵集合式(5)代入式(4)中, 可得到如下 4 维的拉格朗日酉算子:

$$m(\theta) = -\frac{1}{4} \begin{pmatrix} \eta\varepsilon v & v\tau\eta & \eta\varepsilon\tau & -v\tau\varepsilon \\ -v\tau\varepsilon & \eta\varepsilon v & v\tau\eta & \eta\varepsilon\tau \\ \eta\varepsilon\tau & -v\tau\varepsilon & \eta\varepsilon v & v\tau\eta \\ v\tau\eta & \eta\varepsilon\tau & -v\tau\varepsilon & \eta\varepsilon v \end{pmatrix}, \quad (6)$$

其中, $v=(1+e^{i\theta})$, $\tau=(1-e^{i\theta})$, $\varepsilon=(ie^{i\theta}+1)$, $\eta=(ie^{i\theta}-1)$. 在式(6)中, 当 $\theta=0$ 时, 酉算子 $m(0)$ 是单位矩阵.

3 提出的 QSMM 协议

本协议包括 6 个阶段: 初始化阶段、制备粒子和发送粒子阶段、参与者计算阶段、安全验证阶段、随机数收集阶段与结果输出阶段, 提出的量子安全多方求积协议见图 1.

3.1 初始化阶段

(1) 协议中的实体

协议包括三种实体: 半可信的第三方(Server), n 个参与者 $\{P_1, P_2, \dots, P_n\}$ 和半诚实的参与者代表 P_t ($t \in \{1, 2, \dots, n\}$). 这三种实体的内涵如下.

1) Server 是半可信的, 即它会正确地计算多方求积, 但它也可能在不与任何参与者合谋的情况下, 单方面地获取参与者的秘密整数.

2) 参与者 $\{P_1, P_2, \dots, P_n\}$ 中部分是不诚实的, 不诚实的参与者会与其他不诚实的参与者合谋(总人数少于 $n-1$ 个)进行攻击.

3) 参与者代表 P_t 是半诚实的, 它由 n 个参与者共同推选出来, 它会正确地计算收集到的所有随机数的乘积, 并发送给 Server, 也可能会联合不诚实的参与者(人数少于 $n-2$ 个)获取他人的秘密整数.

(2) 协议初始化

首先, Server 通过安全信道发送给 n 个参与者每人一个随机的私有密钥 k_i ($i \in \{1, 2, \dots, n\}$), 并满足

$$\prod_{i=1}^n k_i (\text{mod } d) = 1, \quad (7)$$

其中, d 是维数.

每一个参与者 P_i ($i=1, 2, \dots, n$) 的秘密整数为 x_i , n 个参与者想要计算

$$\text{product} = \prod_{i=1}^n x_i (\text{mod } d), \quad (8)$$

而不让其他人知道自己的秘密整数 x_i .

3.2 制备粒子和发送粒子阶段

Step 1 Server 从式(2)的两个无偏基子集 $\{|\phi_i\rangle\}_{i=0}^{d-1}$ 和

$\{|\psi_j\rangle\}_{j=1}^{d-1}$ 中, 随机选取 n 个量子态 $\{|\alpha_i\rangle \mid i=1, 2, \dots, n\}$, 并记录对应序号 i 的量子态.

Step 2 Server 将量子态 $|\alpha_i\rangle$ ($i=1, 2, \dots, n$) 发送给参与者 P_i .

3.3 参与者计算阶段

当每一个参与者 P_i ($i=1, 2, \dots, n$) 收到粒子 $|\alpha_i\rangle$ 后, 执行如下操作.

Step 3 P_i 随机生成一个正整数 r_i , 然后使用此随机正整数 r_i 、自己的秘密整数 x_i 和私有密钥 k_i 计算拉格朗日酉算子的角度值

$$\theta_i = \frac{2\pi(r_i \times x_i \times k_i) \text{mod } d}{d}. \quad (9)$$

Step 4 P_i 将 k_i 和 $r_i \times x_i \times k_i (\text{mod } d)$ 进行拼接, 可得到

图 1 量子安全多方求积协议

Figure 1 Quantum secure multiparty multiplication protocol.

$(k_i \| (r_i \times x_i \times k_i \bmod d))$, 然后 P_i 选择一个安全的 Hash 函数, 计算拼接后的哈希值 $h_i = \text{Hash}(k_i \| (r_i \times x_i \times k_i \bmod d))$.

Step 5 P_i 将在式(9)中计算得到的 θ_i 代入式(3), 得到拉格朗日酉算子 $m(\theta_i)$, 然后对粒子 $|\alpha_i\rangle$ 执行酉算子, 得到 $m(\theta_i)|\alpha_i\rangle = |\alpha'_i\rangle$.

Step 6 P_i 将粒子 $|\alpha'_i\rangle$ 和 Hash 值 h_i 一同发送给 Server.

3.4 安全验证阶段

Step 7 Server 根据发送给参与者 P_i 的初始态 $|\alpha_i\rangle$, 使用测量基 $\left\{ m\left(\frac{2\pi z}{d}\right) |\alpha_i\rangle \mid z = 0, 1, \dots, d-1 \right\}$ 对 P_i 发送回来的粒子 $|\alpha'_i\rangle$ 执行测量, 测量结果对应基中的 z 值为 z_i .

Step 8 Server 将 k_i 与 z_i 进行拼接 ($k_i \| z_i$), 然后计算 Hash 值 $h'_i = \text{Hash}(k_i \| z_i)$, 比对 h'_i 是否与 P_i 发回的 h_i 相同, 若相同则继续执行协议, 否则终止协议.

3.5 随机数收集阶段

Step 9 $P_i (i \in \{1, 2, \dots, n\})$ 从每一个参与者手中收集随机正整数 $r_i (i = 1, 2, \dots, n)$, 计算所有随机正整数的乘积 $R = \prod_{i=1}^n r_i \bmod d$, 最后 P_i 将 R 发送给 Server.

3.6 结果输出阶段

Step 10 Server 对所有参与者发送回来的粒子执行验证后, 将得到的测量值 $z_1, z_2, \dots, z_n$ 进行求积

$$\text{product}' = \prod_{i=1}^n z_i \bmod d. \quad (10)$$

接着, Server 使用随机正整数的乘积 R , 计算所有秘密整数的乘积 $\text{product} = \text{product}' / R \bmod d$. 最后, Server 将计算结果 product 告知各个参与者.

4 正确性证明

引理 1 每一个参与者将执行完拉格朗日酉算子的量子态发送给 Server, Server 测量收到的量子态可得到唯一对应的测量值 $z_v (v = 1, 2, \dots, n)$, 并可以进一步正确计算出安全多方求积的结果.

证明 Server 从式(2)的两个无偏基子集 $\{|\phi_i\rangle\}_{i=0}^{d-1}$ 和 $\{|\psi_j\rangle\}_{j=1}^{d-1}$ 中随机选取一个量子态, 然后将此量子态 $|\alpha_v\rangle$ 发送给参与者 P_v , P_v 对 $|\alpha_v\rangle$ 执行 d 维拉格朗日酉算子时:

(1) 若 Server 从无偏基中 $\{|\phi_i\rangle\}_{i=0}^{d-1}$ 选取的是第 l

($l \in \{1, 2, \dots, d-1\}$) 个量子态 $|\alpha_v\rangle$, 标记为 $|\alpha_{v,l}\rangle$.

$$\begin{aligned} m(\theta_v)|\alpha_{v,l}\rangle &= \sum_{j,k=0}^{d-1} \frac{\prod_{k \neq j} (e^{i\theta} - \omega^k)}{\prod_{k \neq j} (\omega^j - \omega^k)} q^j |\alpha_{v,l}\rangle \\ &= \prod_{k \neq 0} \frac{e^{i\theta} - \omega^k}{\omega^0 - \omega^k} q^0 |\alpha_{v,l}\rangle + \prod_{k \neq 1} \frac{e^{i\theta} - \omega^k}{\omega^1 - \omega^k} q^1 |\alpha_{v,l}\rangle \\ &\quad + \dots + \prod_{k \neq (d-1)} \frac{e^{i\theta} - \omega^k}{\omega^{d-1} - \omega^k} q^{d-1} |\alpha_{v,l}\rangle, \end{aligned} \quad (11)$$

其中 $\omega = e^{2\pi i/d}$. 当 $\theta_v = (2\pi z_v)/d$ ($z_v \in \{0, 1, \dots, d-1\}$) 时, 式 (11) 可得

$$\begin{aligned} &\prod_{k \neq 0} \frac{e^{\frac{2\pi z_v i}{d}} - \omega^k}{1 - \omega^k} q^0 |\alpha_{v,l}\rangle + \prod_{k \neq 1} \frac{e^{\frac{2\pi z_v i}{d}} - \omega^k}{\omega - \omega^k} q^1 |\alpha_{v,l}\rangle \\ &+ \dots + \prod_{k \neq z_v} \frac{e^{\frac{2\pi z_v i}{d}} - \omega^k}{\omega^{z_v} - \omega^k} q^{z_v} |\alpha_{v,l}\rangle \\ &+ \dots + \prod_{k \neq d-1} \frac{e^{\frac{2\pi z_v i}{d}} - \omega^k}{\omega^{d-1} - \omega^k} q^{d-1} |\alpha_{v,l}\rangle \\ &= \prod_{k \neq z_v} \frac{e^{\frac{2\pi z_v i}{d}} - \omega^k}{\omega^{z_v} - \omega^k} q^{z_v} |\alpha_{v,l}\rangle = q^{z_v} |\alpha_{v,l}\rangle = |\alpha'_{v,l}\rangle. \end{aligned} \quad (12)$$

(2) 若 Server 从无偏基 $\{|\psi_j\rangle\}_{j=1}^{d-1}$ 中选取的是第 j ($j \in \{1, 2, \dots, d-1\}$) 个量子态 $|\alpha_v\rangle$, 标记为 $|\alpha_{v,j}\rangle$. 由式 (11) 和 (12) 得

$$\begin{aligned} m(\theta_v)|\alpha_{v,j}\rangle &= \sum_{t,k=0}^{p-1} \frac{\prod_{k \neq t} (e^{i\theta_v} - \omega^k)}{\prod_{k \neq t} (\omega^t - \omega^k)} q^t |\alpha_{v,j}\rangle \\ &= \sum_{t,k=0}^{p-1} \frac{\prod_{k \neq t} (e^{i\theta_v} - \omega^k)}{\prod_{k \neq t} (\omega^t - \omega^k)} q^t \frac{1}{\sqrt{d}} \sum_{u=0}^{d-1} e^{\frac{2\pi i}{d} u j} |u\rangle \\ &= \frac{1}{\sqrt{d}} \sum_{u=0}^{d-1} e^{\frac{2\pi i}{d} u j} \left(\sum_{t,k=0}^{p-1} \frac{\prod_{k \neq t} (e^{i\theta_v} - \omega^k)}{\prod_{k \neq t} (\omega^t - \omega^k)} q^t |u\rangle \right) \\ &= \frac{1}{\sqrt{d}} \sum_{u=0}^{d-1} e^{\frac{2\pi i}{d} u j} |u'\rangle = |\alpha'_{v,j}\rangle. \end{aligned} \quad (13)$$

通过以上分析可知, 对于每一个由拉格朗日酉算子操作后得到的量子态 $|\alpha'_{v,j}\rangle$, 它对应测量基

$\{m(2\pi z_v/d)|\alpha_v\rangle \mid z_v = 0, 1, \dots, d-1\}$ 中的一个状态. 拉格朗日酉算子作用的效果通过式 (12) 看出, 它与 $\{q^0, q, q^2, q^3, \dots, q^{p-1}\}$ 密切相关.

对于第一种情况 $|\alpha_{v,l}\rangle \in \{|\phi_l\rangle\}_{l=0}^{d-1}$, 因为 $|\alpha_{v,l}\rangle$ 属于单态, 当取拉格朗日酉算子中的某一位置的列向量时, $\theta_v \in (0, 2\pi)$ 取不同的角度, 此位置的列是不同的^[25]. 由上述证明可得在确定 $\theta_v = 2\pi z_v/d$ ($z_v \in \{0, 1, \dots, d-1\}$) 的情况下, 计算得到的酉算子与循环矩阵群相关, 而且通过此酉算子和 $|\alpha_{v,l}\rangle$ ($|\alpha_{v,l}\rangle \in \{|\phi_l\rangle\}_{l=0}^{d-1}$) 计算得到的基向量都是互不相同的, 并且相互正交. 对于第二种情况 $|\alpha_{v,j}\rangle \in \{|\psi_j\rangle\}_{j=1}^{d-1}$, 因为 $|\alpha_{v,j}\rangle$ 属于叠加态, $m(\theta_v)$ 会对 $|\alpha_{v,j}\rangle$ 中的每一种量子态进行操作, 最终的结果是叠加态 $|\alpha_{v,j}\rangle$ 中每一种量子态的相位发生改变. 因此, 对量子态 $|\alpha_v\rangle$ ($|\alpha_v\rangle \in \{|\phi_l\rangle\}_{l=0}^{d-1}, \{|\psi_j\rangle\}_{j=1}^{d-1}$) 执行拉格朗日酉算子 $m(\theta_v)$ 后, 可得到一个新的量子态 $|\alpha'_{v,j}\rangle$, 即将 θ_v 代入式 (3) 后计算 $m(\theta_v)|\alpha_v\rangle = |\alpha'_{v,j}\rangle$. 当 Server 测量参与者发送回来的粒子后, 根据测量结果可得 z_v ($z \in \{0, 1, \dots, d-1\}$). 所以 Server 对收到的 n 个量子态测量结束后, 将得到的所有 z 值求积可得

$$\begin{aligned} &\text{product}' \\ &= \prod_{v=1}^n z_v \text{modd} \\ &= (r_1 \times x_1 \times k_1) \times (r_2 \times x_2 \times k_2) \dots \\ &\quad \times (r_n \times x_n \times k_n) \text{modd} \\ &= \left(\prod_{v=1}^n r_v \text{modd} \right) \times \left(\prod_{v=1}^n k_v \text{modd} \right) \\ &= \left(\prod_{v=1}^n r_v \text{modd} \right) \times \left(\prod_{v=1}^n x_v \text{modd} \right). \end{aligned} \quad (14)$$

最后, Server 使用随机正整数的乘积 $R \in \{1, 2, \dots, d-1\}$, 计算秘密整数的求积结果 $\text{product} = \text{product}'/R(\text{modd})$.

进一步, 根据费马小定理: 若 d 是素数, R 是正整数且不能被 d 整除, 则 $(R)^{d-1} \equiv 1(\text{modd})$, 将上式改写为 $(R)^{d-2} \equiv R^{-1}(\text{modd})$. 那么, 安全多方求积的结果也可以写为 $\text{product} = \text{product}' \times R^{d-2}(\text{modd})$.

5 安全性分析

5.1 外部攻击

攻击者Eve的外部攻击可能是截获-测量-重发攻击、纠缠-测量攻击和伪造攻击. 在提出的协议中, 通过使用无偏基态和插入随机整数的方式来防止Eve的外部攻击, 对这三种攻击的分析如下.

(1) 截获-测量-重发攻击

假设粒子在传输过程中存在攻击者Eve, 他具有一定的计算能力并且知道方案的执行流程, 他对传输过程中的粒子进行截获和测量, 此处从两条路线考虑.

1) 如果他截获Server发送给参与者 $P_j (j \in \{1, 2, \dots, n\})$ 的粒子 $|\alpha_j\rangle \in \left\{ \{|\phi_i\rangle\}_{i=0}^{d-1}, \{|\psi_i\rangle\}_{i=1}^{d-1} \right\}$, 这个过程是没有意义的, 因为此时该粒子上并没有携带任何有价值的信息. 类似于BB84协议^[26], Eve如果只是想要获得这个粒子的初始状态, 但由于这个粒子选自相互无偏基态, 那么他有1/2的概率会选择错误的测量基. 若他使用错误的测量基去测量截获的粒子, 只会有1/d的概率得到正确的测量值. 所以通过以上分析可得, Eve的攻击行为会有 $(d-1)/(2d)$ 的概率被Server检测到.

2) 假设Eve截获 P_j 发送给Server的粒子 $|\alpha'_j\rangle$, 然后对粒子进行测量. 同样, 如果Eve使用了错误的测量基去测量 $|\alpha'_j\rangle$, 那么在安全验证阶段也会有 $(d-1)/(2d)$ 的概率被Server发现.

即使Eve以 $(d+1)/(2d)$ 的概率正确地测量出 z_j 值, 由于他无法获得私有密钥 k_j 和随机正整数 r_j , 所以他也就无法推断出 $z_j = (r_j \times x_j \times k_j) \bmod d$ 中的秘密整数 x_j . 进一步, 如果Eve正确测量后得到 z_j , 然后制备一个与截获粒子一样的新粒子, 接着将这个新的粒子连同截获的Hash值 h_j 一同发送给Server, 实现截获-测量-重发攻击, 这个过程并不能使攻击者Eve获得有价值的信息, 对协议的安全性不会构成威胁.

(2) 纠缠-测量攻击

假设Eve截获了 $P_j (j \in \{1, 2, \dots, n\})$ 发送给Server的量子态 $|\alpha'_j\rangle$, 然后他制备了一个辅助粒子 $|\partial\rangle (\partial \in \{0, 1, \dots, d-1\})$. 首先, 他将截获到的粒子 $|\alpha'_j\rangle$ 和辅助粒子 $|\partial\rangle$ 形成一个两粒子复合系统, 然后通过 d 维

CNOT操作将 $|\partial\rangle$ 和 $|\alpha'_j\rangle$ 进行变换, 以 $|\alpha'_j\rangle$ 为控制粒子, $|\partial\rangle$ 为目标粒子, 可得到 $|\varphi\rangle = |\alpha'_j\rangle|\alpha'_j + \partial\rangle$. 对于纠缠交换得到的结果, 他测量自己的辅助粒子可得到 $\alpha'_j + \partial$, 因为知道 ∂ 的值, 于是可推出 α'_j , 但由于缺少必要的数据信息 k_j 和 r_j , 所以Eve仍无法获得参与者 P_j 的秘密整数 x_j , 故该攻击无效.

(3) 伪造攻击

假设Eve假冒为其中一个参与者 $P_j (j \in \{1, 2, \dots, n\})$, 首先他制备一个粒子 $|\alpha'_j\rangle^*$, 并按照协议的流程执行结束后发送给Server. 由于参与者 $\{P_1, P_2, \dots, P_n\}$ 收到的初始态都是随机的, 并且Eve不知道Server和真正参与者 P_j 共享的私有密钥 k_j 值, 所以Server在收到伪造的粒子 $|\alpha'_j\rangle^*$ 之后, 通过安全验证他能够发现攻击者的存在. 因为私有密钥 k_j 对Eve是未知的, Eve在伪造粒子 $|\alpha'_j\rangle^*$ 之后, 粒子的测量结果为 z_j^* , 伪造出来的私有密钥为 k_j^* , 计算 $\text{Hash}(k_j^* \| z_j^*)$ 得到Hash值 h_j^* . 那么在安全验证阶段, Server通过 $\text{Hash}(k_j \| z_j^*)$ 计算出正确的Hash值 h_j , 与Eve发送的Hash值 h_j^* 比对, 很容易发现Eve的伪造行为, 然后终止协议. 因此该攻击对本协议无效.

5.2 内部攻击

内部攻击包括三种情况: 来自半可信Server的攻击、来自多个不诚实参与者的攻击和半诚实参与者代表 P_i 联合其他参与者的攻击.

(1) 来自半可信Server的攻击

提出的协议中, Server是一个半可信的第三方, 他在不与任何参与者合谋的情况下, 想要通过自己已知的信息窃取参与者 $P_j (j \in \{1, 2, \dots, n\})$ 的秘密整数 x_j . 在提出的协议中, Server可以获知的信息包括: 量子态 $|\alpha_j\rangle$ 、私有密钥 k_j 、测量值 z_j 和所有随机正整数的模 d 乘积 R . 因为参与者代表 P_i 是半诚实的, 他发送给Server的数据只是所有随机正整数模 d 的乘积 R , Server通过自己已知的信息无法从乘积 R 推出 P_j 的随机正整数 r_j , 那么Server在获得测量值 z_j 后, 也就无法推断出 P_j 的秘密整数 x_j . 因此来自半可信Server的攻击是无效的.

(2) 来自多个不诚实参与者的攻击

当多个不诚实参与者合谋进行攻击, 他们想要通过已知的信息推出参与者 P_j 的秘密整数, 这个过程存在极大的难度. 由于这些不诚实参与者不知道 P_j 收到粒子的初始态 $|\alpha_j\rangle$ ($j=1, 2, \dots, n$), 以及 P_j 返回给Server的粒子在测量后得到的 z_j 值. 假设这些不诚实参与者通过5.1节中的截获-测量攻击或纠缠-测量攻击, 正确获得了参与者 P_j 的 z_j 值, 但由于参与者 P_j 的 z_j 值中包含了私有密钥 k_j 和随机正整数 r_j , 所以这些不诚实参与者不可能推断出参与者 P_j 的秘密整数 x_j . 以常见的参与者 P_{j-1} 和 P_{j+1} 合谋攻击 P_j 的情况为例.

1) 参与者 P_{j-1} 和 P_{j+1} 想要合谋出 P_j 的 k_j 值.

因为私有密钥 k_j 只有Server与 P_j 才知道, 参与者 P_{j-1} 和 P_{j+1} 并不知道, 他们想要窃取只有通过式(7)推出参与者 P_j 的 k_j , 由于他们的合谋人数少于 $n-1$ 个, 所以他们不能够通过式(7)推出 k_j 的值.

2) 参与者 P_{j-1} 和 P_{j+1} 想要合谋出 P_j 的 r_j 值.

P_j 的随机正整数 r_j 只有 P_j 和半诚实参与者代表 P_i 知道, 攻击者想要窃取只有与 P_i 合谋. 如果他们截获 P_i 发送给Server的乘积 R , 和上述情况1)类似, 由于他们的合谋人数较少, 同样也无法确定 r_j 的值.

通过以上分析可知, 参与者 P_{j-1} 和 P_{j+1} 两人合谋窃取 k_j 和 r_j 是不可能成功的, 因此他们也不可能推断出参与者 P_j 的秘密整数 x_j .

(3) 半诚实参与者代表 P_i 联合其他参与者的攻击

若参与者代表 P_i 联合多个不诚实参与者, 获取其他参与者的秘密整数, P_i 向这些不诚实参与者提供自己已知的所有信息, 包括收集到的所有随机数等. 假设这些不诚实参与者通过5.1节中的截获-测量攻击或纠缠-测量攻击, 正确获得了参与者 P_j 的 z_j 值, 由于 $z_j =$

$r_j \times x_j \times k_j \pmod{d}$, 私有密钥 k_j 只有Server和 P_j 持有, 所以这些不诚实的参与者在不知道 k_j 的情况下, 无法获得参与者 P_j 的秘密整数 x_j . 如果他们想要使用自己手中的私有密钥, 通过式(7)推出参与者 P_j 的私有密钥值, 由于他们的合谋人数少于 $n-1$ 个, 不能够确定私有密钥的值, 所以无法成功.

6 性能分析

本节对提出的协议与其他4个类似协议进行性能比较, 为了便于对比分析, 在此统一各协议的共同性质: 制备粒子的初始态均在 d 维Hilbert空间中, 参与计算的人数为 n 个, 求积的结果在有限域 $GF(d)$ 上, 5个协议的性能比较结果如表1所示.

在表1中, QFT和IQFT分别表示对粒子执行量子傅里叶变换和逆傅里叶变换, U 表示某种单次的变换酉操作, m 代表协议中需要使用到的诱骗粒子个数, M 代表对单个粒子的测量操作. 文献[22]对纠缠态执行的Oracle操作进行分解, 分解为形成纠缠态、 U 操作和解纠缠操作. 计算消耗表示协议中, 需要执行以上酉操作的消耗总和. 资源消耗表示协议需要消耗的粒子总个数. 通信消耗是以单个粒子在量子信道上传输一次为单位, 在协议执行过程中总的粒子传输消耗.

在文献[19]中提出了两种QSM协议, 由于第一种协议无法抵抗少于 $n-1$ ($n>3$)个参与者的合谋攻击, 第二种协议对前者的安全性进行了增强, 为了比较的公平性, 这里使用第二种协议进行比较. 参与者 P_k ($k=1, 2, \dots, n$)将自己的秘密整数 x_k 分割为 n 个随机整数 $x_{k,r}$ 相乘的形式, 其中 $x_{k,r} \in \{0, 1, \dots, d-1\}$, $r=1, 2, \dots, n$, 然后 P_k 作为发起者完成计算 $\prod_{j=1}^n x_{j,k}$ 的任务. 计算过程如

表 1 五个协议的性能比较结果

Table 1 Performance comparison results of five protocols

协议	粒子类型	制备粒子方	量子操作类型	计算消耗	资源消耗	通信消耗	粒子传输方式
文献[19]	多粒子纠缠态	Server	QFT, IQFT, 变换操作 U	$QFT(n+1)+IQFT(n+1)+U(4n)+M(2(n+1))$	$2(n+1)$	$2n(n+1)$	环形
文献[20]	单粒子	Server, 参与者	变换操作 U	$U(n)+M(1+(n+1)m)$	$nm+m+1$	$(n+1)(m+1)$	环形
文献[21]	多粒子纠缠态	Server	QFT, 变换操作 U	$QFT(n+1)+U(2n-1)+M(n)$	n	$n-1$	树形
文献[22]	多粒子纠缠态	Server, 参与者	QFT, IQFT, Oracle操作	$QFT(1)+IQFT(1)+U(3n-1)+M(2)$	$n+1$	n	环形
本协议	单粒子	Server	变换操作 U	$U(n)+M(n)$	n	$2n$	树形

下: P_k 制备了一个信息粒子和一个窃听检查粒子, 并对信息粒子执行了一次 QFT, 又以该粒子为控制粒子, 窃听检查粒子为目标粒子执行一次 CNOT 操作生成纠缠态, 将信息粒子以环形传输模式在 n 个参与者间传输, 每一个参与者 P_w 都会对信息粒子执行一次酉操作, 其中 $w \neq k$, $w \in \{1, 2, \dots, n\}$, 将自己秘密整数的分解值 $x_{w,k}$ 编码到信息粒子上. 最终参与者 P_{k-1} 将信息粒子返还到 P_k 手中. P_k 在对信息粒子执行 IQFT 等操作后得到求积结果 $\prod_{j=1}^n x_{j,k}$. 以上过程执行 n 次后, n 个参与者共同随机指定一个可信任的参与者 P_i , 由 P_i 再执行一次上述过程, 计算得到 $\prod_{a=1}^n \prod_{j=1}^n x_{a,j}$ 的结果. 因此, 该协议的计算消耗为 $\text{QFT}(n+1) + \text{IQFT}(n+1) + U(4n) + M(2(n+1))$, 资源消耗为 $2(n+1)$, 通信消耗为 $2n(n+1)$.

在文献[20]中, 乘法协议由一次 QSMS 协议和一次 QSMM 协议组成, 为了比较的公平性, 假定 Server 使用一个信息粒子只进行 QSMM 协议的执行. Server 从两组无偏基中随机选取一组基制备一个信息粒子, 并将其随机插入 m 个诱骗粒子之中, 随后发送给第一个参与者, 第一个参与者对 m 个诱骗粒子进行窃听检查后, 对信息粒子执行一次酉操作, 之后同样将此信息粒子随机插入 m 个诱骗粒子之中, 然后将它们发送给下一个参与者, 在重复 $n-1$ 次后, 将粒子序列返还给 Server, Server 进行测量后得出求积结果. 因此, 此协议的计算消耗为 $U(n) + M(1 + (n+1)m)$, 资源消耗为 $nm + m + 1$, 通信消耗为 $(n+1)(m+1)$.

在文献[21]中, 发起者 P_1 选中 n 个参与者执行 QSMM, 首先 P_1 制备 n 个信息粒子, 然后对第一个粒子执行 QFT 操作, 并以第一个粒子为控制粒子, 剩余粒子为目标粒子执行 $n-1$ 次 SUM 门操作生成纠缠态, 然后将 $n-1$ 个粒子分别发送给其余 $n-1$ 个参与者, n 个参与者同时对自己手中的粒子执行 QFT 等操作后得到计算结果. 因此, 该协议的计算消耗为 $\text{QFT}(n+1) + U(2n-1) + M(n)$, 资源消耗为 n , 通信消耗为 $n-1$.

在文献[22]中, 发起者 P_1 制备一个信息粒子 $|s_i\rangle$ 并对其执行一次 QFT 操作, 再制备一个辅助粒子, 通过对这两个粒子执行 CNOT 操作生成纠缠态, 之后通过环形传输的方式在各参与者间传输辅助粒子. 每一个参与者根据自己的秘密值 s_i ($i \in \{2, 3, \dots, n\}$) 制备一个信息粒子 $|s_i\rangle$, 通过对辅助粒子和信息粒子 $|s_i\rangle$ 执行 *Ora-*

cle 操作, 实现两粒子的纠缠、秘密值的嵌入以及解纠缠操作, 最后 P_1 对信息粒子和辅助粒子执行一次 CNOT 操作后, 再对信息粒子执行一次 IQFT 操作获得求积结果. 因此, 该协议的计算消耗为 $\text{QFT}(1) + \text{IQFT}(1) + U(3n-1) + M(2)$, 资源消耗为 $n+1$, 通信消耗为 n .

在本协议中, Server 制备了 n 个单粒子, 然后通过树形粒子传输模式发送给 n 个参与者. 每个参与者对收到的粒子执行 1 次酉操作, 之后将粒子返回给 Server. Server 收到粒子后, 对 n 个粒子执行测量操作. 因此, 本协议的计算消耗为 $U(n) + M(n)$, 资源消耗为 n , 通信消耗为 $2n$.

从表 1 可以看出, 文献[19,21,22]都采用多粒子纠缠态作为信息粒子, 文献[20]中信息粒子选自无偏基态, 但协议要求 Server 和参与者都需要制备无偏基态, 这使得该协议对 Server 和参与者的量子制备能力要求较高. 本协议采用单粒子作为信息载体, 不需要参与者制备粒子, 并且在计算消耗方面低于其他 4 个协议. 文献[21,22]和本协议的各类消耗综合为 $O(n)$ 的复杂度, 文献[19,20]的各类消耗综合为 $O(n^2)$ 的复杂度.

7 仿真

我们通过一个仿真实验来模拟该协议的执行, 根据量子力学基本理论, 在不涉及量子硬件的基础上使用经典计算机对协议进行验证. 仿真实验使用 Microsoft 开源量子编程语言 Q# 模拟量子态的制备, 在计算拉格朗日酉算子、执行酉操作、安全验证、随机数收集和结果输出部分使用经典编程语言 Python 进行计算和输出. 程序的开发环境为 VS Code 1.58.2, QDK v0.18.2107153439, NET Core SDK 3.1.104 以及 Python 3.7.9.

假设 3 个参与者 P_1 , P_2 和 P_3 在 7 维的量子空间中进行多方乘法计算, 他们的秘密整数分别为有限域 GF(7) 上的 $\{x_1=4, x_2=3, x_3=5\}$, 想要计算的是 $4 \times 3 \times 5 \pmod{7}$ 的结果. Server 分发给参与者 P_1 , P_2 和 P_3 的私有密钥分别为 $k_1=5$, $k_2=19$, $k_3=2$. 实验中随机选用的初始态分别为: $|\alpha_1\rangle$ 选自无偏基 $\{|\phi_l\rangle\}_{l=0}^{d-1}$ 中的第 $l=1$ 个量子态 $|1\rangle$, $|\alpha_2\rangle$ 选自无偏基 $\{|\phi_l\rangle\}_{l=0}^{d-1}$ 中的第 $l=4$ 个量子态 $|4\rangle$, $|\alpha_3\rangle$ 选自无偏基 $\{|\psi_j\rangle\}_{j=1}^{d-1}$ 中的第 $j=6$ 个量子态

$$0.3780|0\rangle + (0.2357 - 0.2955j)|1\rangle - (0.0841 - 0.3684j)|2\rangle \\ - (0.3405 - 0.1640j)|3\rangle - (0.3405 + 0.1639j)|4\rangle \\ - (0.0841 + 0.3684j)|5\rangle + (0.2357 + 0.2955j)|6\rangle,$$

其中对 $|\alpha_3\rangle$ 的相位值四舍五入取小数点后4位. 参与者选择的随机正整数分别为 $r_1=31$, $r_2=23$, $r_3=58$.

首先, Server制备上述三个初始态, 在记录初始态的信息后, Server将这三个初始态分别发送给三个参与者. 每一个参与者 P_i ($i=1, 2, 3$)根据自己的秘密整数 x_i 、拥有的私有密钥 k_i 和随机正整数 r_i 计算角度值 θ_i . 参与者的计算结果输出见表2.

将表2中的 θ_1 , θ_2 和 θ_3 代入式(3)中, 计算7维拉格朗日酉算子, 并使用此酉算子对相应量子态执行酉操作.

实验中, 使用SHA-256作为安全验证阶段的Hash函数, 每一个参与者根据自己的秘密整数 x_i 、私有密钥 k_i 和随机正整数 r_i 计算拼接结果: $(k_i || (x_i \times k_i \times r_i \pmod{7}))$, 然后使用此拼接结果计算SHA-256值, Hash结果见表3.

Server接收到参与者 P_1 , P_2 和 P_3 发送回来的粒子和Hash验证值后, 对粒子进行测量操作. 首先, Server根据发送给各参与者的初始态, 计算出各组测量基:

$$\{m(2\pi z/7)|\alpha_z\rangle \mid z = 0, 1, \dots, 6\},$$

$$\{m(2\pi z/7)|\alpha_z\rangle \mid z = 0, 1, \dots, 6\},$$

$$\{m(2\pi z/7)|\alpha_z\rangle \mid z = 0, 1, \dots, 6\},$$

测量基的计算结果见表4和5.

使用表4中的第一组测量基对参与者 P_1 发送回来的粒子进行测量, 使用第二组测量基对参与者 P_2 发送回来的粒子进行测量, 使用表5中的测量基对参与者 P_3 发送回来的粒子进行测量. 测量结果对应的 z 值分别为 $z_1=4$, $z_2=2$, $z_3=6$.

接着, Server进行安全验证操作. 首先将参与者 P_1 , P_2 和 P_3 共享的私有密钥 k_i ($i=1, 2, 3$)和测量得到的测量结果 z_i ($i=1, 2, 3$)进行拼接, 按照 $(k_i || z_i)$ 的形式, 可分别得到拼接后的结果54, 192和26, Server对每一个拼接结果计算Hash值后, 将计算的Hash值与对应参与者发送过来的Hash值比对. 上述拼接结果与表3中的拼接结果相同, 并且计算出的Hash值也相同, 说明安全验证通过.

在三个参与者中随机选择一位参与者代表, 由他来收集其他参与者的随机正整数, 并计算所有参与者的随机数乘积 R , 计算结果为 $R=31 \times 23 \times 58 \pmod{7}=5$, 计算完成后, 参与者代表将结果 R 发送给Server. Server收到参与者代表发送过来的 R 后, 计算测量值的求积

表 2 参与者的计算结果输出^{a)}

Table 2 Calculation results output of participants^{a)}

P_i	x_i	k_i	r_i	θ_i	输出粒子
P_1	4	5	31	$\frac{2\pi \times 4}{7}$	$ 4\rangle$
P_2	3	19	23	$\frac{2\pi \times 2}{7}$	$ 2\rangle$
P_3	5	2	58	$\frac{2\pi \times 6}{7}$	$ \alpha'_3\rangle$

a) $|\alpha'_3\rangle = (0.2357 + 0.2955j)|0\rangle + 0.3780|1\rangle + (0.2357 - 0.2955j)|2\rangle - (0.0841 - 0.3684j)|3\rangle - (0.3405 - 0.1640j)|4\rangle - (0.3405 + 0.1639j)|5\rangle - (0.0841 + 0.3684j)|6\rangle$.

表 3 Hash结果

Table 3 Hash results

P_i	$(k_i (r_i \times x_i \times k_i \pmod{7}))$	Hash值 h_i
P_1	54	2fca346db656187102ce806ac732e06a62df0dbb2829e511a770556d398e1a6e
P_2	192	eb3be230bbd2844b1f5d8f2e4fab9ffba8ab22cfeeb69c4c1361993ba4f377b9
P_3	26	5f9c4ab08cac7457e9111a30e4664920607ea2c115a1433d7be98e97e64244ca

表 4 测量基(1)

Table 4 Measurement bases (1)

组别	$z=0$	$z=1$	$z=2$	$z=3$	$z=4$	$z=5$	$z=6$
第一组 测量基	$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$
第二组 测量基	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$

表 5 测量基(2)

Table 5 Measurement bases (2)

z 值	测量基
$z=0$	$0.3780 0\rangle + (0.2357 - 0.2955j) 1\rangle - (0.0841 - 0.3684j) 2\rangle - (0.3405 - 0.1640j) 3\rangle - (0.3405 + 0.1639j) 4\rangle - (0.0841 + 0.3684j) 5\rangle + (0.2357 + 0.2955j) 6\rangle$
$z=1$	$(0.2357 - 0.2955j) 0\rangle - (0.0841 - 0.3684j) 1\rangle - (0.3405 - 0.1640j) 2\rangle - (0.3405 + 0.1639j) 3\rangle - (0.0841 + 0.3684j) 4\rangle + (0.2357 + 0.2955j) 5\rangle + 0.3780 6\rangle$
$z=2$	$-(0.0841 - 0.3684j) 0\rangle - (0.3405 - 0.1640j) 1\rangle - (0.3405 + 0.1639j) 2\rangle - (0.0841 + 0.3684j) 3\rangle + (0.2357 + 0.2955j) 4\rangle + 0.3780 5\rangle + (0.2357 - 0.2955j) 6\rangle$
$z=3$	$-(0.3405 - 0.1640j) 0\rangle - (0.3405 + 0.1639j) 1\rangle - (0.0841 + 0.3684j) 2\rangle + (0.2357 + 0.2955j) 3\rangle + 0.3780 4\rangle + (0.2357 - 0.2955j) 5\rangle - (0.0841 - 0.3684j) 6\rangle$
$z=4$	$-(0.3405 + 0.1639j) 0\rangle - (0.0841 + 0.3684j) 1\rangle + (0.2357 + 0.2955j) 2\rangle + 0.3780 3\rangle + (0.2357 - 0.2955j) 4\rangle - (0.0841 - 0.3684j) 5\rangle - (0.3405 - 0.1640j) 6\rangle$
$z=5$	$-(0.0841 + 0.3684j) 0\rangle + (0.2357 + 0.2955j) 1\rangle + 0.3780 2\rangle + (0.2357 - 0.2955j) 3\rangle - (0.0841 - 0.3684j) 4\rangle - (0.3405 - 0.1640j) 5\rangle - (0.3405 + 0.1639j) 6\rangle$
$z=6$	$(0.2357 + 0.2955j) 0\rangle + 0.3780 1\rangle + (0.2357 - 0.2955j) 2\rangle - (0.0841 - 0.3684j) 3\rangle - (0.3405 - 0.1640j) 4\rangle - (0.3405 + 0.1639j) 5\rangle - (0.0841 + 0.3684j) 6\rangle$

结果: $product'=4 \times 2 \times 6 \pmod{7}=6$, 因此, 在有限域 $GF(7)$ 上的安全多方求积结果为: $product=product'/R \pmod{7}=6/5 \pmod{7}=6 \times 5^5 \pmod{7}=4$. 最后, Server 将计算结果告知参与者 P_1, P_2 和 P_3 .

8 结论

本文基于 d 维拉格朗日酉算子提出了一种新的 QSMM 协议, 协议允许所有参与者共同计算有限域 $GF(d)$ 上的模 d 乘法. 通过正确性证明, 诠释了方案的

正确性. 安全性分析表明, 该协议能够同时抵抗提出的外部攻击和内部攻击, 并且根据海森堡测不准原理, 半可信的第三方 Server 可以很大概率发现攻击者的窃听行为. 相对于现有的 QSMM 协议, 该协议对各参与者的粒子制备能力和操作能力要求更低, 并且在计算消耗

方面表现出良好的效率. 最后的仿真实验结果进一步证明了该协议的正确性. 另外, 协议过程中需要所有参与者(其中可能包含不诚实的参与者), 共同推选一位半诚实的参与者代表, 然而此推选方法还有待进一步的研究.

参考文献

- 1 Yao A C. Protocols for secure computations. In: Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science. Chicago, 1982. 160–164
- 2 Crépeau C, Gottesman D, Smith A. Secure multi-party quantum computation. In: Proceedings of the Thirty-fourth Annual ACM Symposium on Theory of Computing. Montreal, 2002. 643–652
- 3 Yoshida M, Obana S. On the (in)efficiency of non-interactive secure multiparty computation. *Des Codes Cryptogr*, 2018, 86: 1793–1805
- 4 Micali S, Goldreich O, Wigderson A. How to play any mental game. In: Proceedings of the Nineteenth ACM Symposium on Theory of Computing. New York, 1987. 218–229
- 5 Abe M. Non-interactive and optimally resilient distributed multiplication. *IEICE Trans Fundam Electron Commun Comput Sci*, 2000, 83: 598–605
- 6 Ji Z X, Zhang H G, Wang H Z, et al. Quantum protocols for secure multi-party summation. *Quantum Inf Process*, 2019, 18: 168
- 7 Li Z G, Fei S M, Wang Z D, et al. Evolution equation of entanglement for bipartite systems. *Phys Rev A*, 2009, 79: 024303, arXiv: 0806.4228
- 8 Busch P, Heinonen T, Lahti P. Heisenberg's uncertainty principle. *Phys Rep*, 2007, 452: 155–176, arXiv: quant-ph/0609185
- 9 Bennett C H, Brassard G. Quantum cryptography: Public key distribution and coin tossing. *Theor Comput Sci*, 2014, 560: 7–11
- 10 Vaccaro J A, Spring J, Chefles A. Quantum protocols for anonymous voting and surveying. *Phys Rev A*, 2007, 75: 012333, arXiv: quant-ph/0504161
- 11 Du J Z, Chen X B, Wen Q Y, et al. Secure multiparty quantum summation. *Acta Phys Sin*, 2007, 56: 6214–6219
- 12 Chen X B, Xu G, Yang Y X, et al. An efficient protocol for the secure multi-party quantum summation. *Int J Theor Phys*, 2010, 49: 2793–2804
- 13 Zhang C, Sun Z W, Huang X, et al. Three-party quantum summation without a trusted third party. *Int J Quantum Inform*, 2015, 13: 1550011
- 14 Zhang C, Situ H, Huang Q, et al. Multi-party quantum summation without a trusted third party based on single particles. *Int J Quantum Inform*, 2017, 15: 1750010
- 15 Duan M Y. Multi-party quantum summation within a d -level quantum system. *Int J Theor Phys*, 2020, 59: 1638–1643
- 16 Song X L, Gou R, Wen A. Secure multiparty quantum computation based on Lagrange unitary operator. *Sci Rep*, 2020, 10: 7921
- 17 Ye T Y, Hu J L. Quantum secure multiparty summation based on the mutually unbiased bases of d -level quantum systems and its application (in Chinese). *Sci Sin-Phys Mech Astron*, 2021, 51: 020301 [叶天语, 胡家莉. 基于 d 级量子系统相互无偏基的量子安全多方求和及其应用. 中国科学: 物理学 力学 天文学, 2021, 51: 020301]
- 18 Ivonovic I D. Geometrical description of quantal state determination. *J Phys A-Math Gen*, 1981, 14: 3241–3245
- 19 Shi R H, Mu Y, Zhong H, et al. Secure multiparty quantum computation for summation and multiplication. *Sci Rep*, 2016, 6: 19655
- 20 Lv S X, Jiao X F, Zhou P. Multiparty quantum computation for summation and multiplication with mutually unbiased bases. *Int J Theor Phys*, 2019, 58: 2872–2882
- 21 Sutradhar K, Om H. Hybrid quantum protocols for secure multiparty summation and multiplication. *Sci Rep*, 2020, 10: 1–9
- 22 Sutradhar K, Om H. Secret sharing based multiparty quantum computation for multiplication. *Int J Theor Phys*, 2021, 60: 3417–3425
- 23 Wootters W K, Fields B D. Optimal state-determination by mutually unbiased measurements. *Ann Phys*, 1989, 191: 363–381
- 24 Cerf N J, Bourennane M, Karlsson A, et al. Security of quantum key distribution using d -level systems. *Phys Rev Lett*, 2002, 88: 127902, arXiv: quant-ph/0107130
- 25 De Vos A, De Baerdemacker S. From reversible computation to quantum computation by Lagrange interpolation. arXiv: 1502.00819
- 26 Bennett C H, Brassard G. Quantum cryptography: Public key distribution and coin tossing. arXiv: 2003.06557

Quantum secure multiparty multiplication based on Lagrange unitary operator

ZHANG LongWei¹, SONG XiuLi^{1*}, LI Chuang² & LIU Yu¹

¹ School of Cyber Security and Information Law, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;

² School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Most of the existing quantum secure multiparty multiplications use entangled states to carry secret information. However, the preparation of these quantum states is difficult. In order to solve this problem, a single-particle secure multiparty product protocol based on a d -level Lagrange unitary operator is proposed. In this protocol, a semi-honest third party (Server) and each participant share a private key. The participant first uses the private key and random number to multiply his own secret integer with module d , and then he uses Lagrange unitary operator to encode the calculation result into an unbiased basis particle. The server obtains the encoded information by measuring the particle sent back by each participant. Finally, the server uses the module d product of all the random numbers sent by the participant's representative to calculate the product corresponding to the secret integers of multiple participants. Performance analysis shows that, compared with similar protocols, the proposed protocol requires less of the entity's quantum computing capabilities. In terms of security, this protocol can resist both external and internal attacks. Finally, the proposed protocol uses the $Q\#$ quantum programming language combined with Python language to simulate and verify its correctness.

quantum secure multiparty multiplication, d -level Lagrange unitary operator, unbiased basis particle, $Q\#$ quantum programming language

PACS: 03.67.Dd, 03.67.Hk, 03.67.Lx

doi: [10.1360/SSPMA-2021-0234](https://doi.org/10.1360/SSPMA-2021-0234)