基于区块链的具有隐私保护的多项式外包计算方案

郭 祯[1] 张 银[1] 安方林[1] 赵科杰[1] 张文杰[1] 叶 俊[1]

[1](海南大学计算机与网络空间安全学院 海口 中国 570228)

摘要 随着区块链的快速发展, 基于区块链的外包计算得到了广泛应用.外包计算允许资源受限的用户将复杂的计算以付费的方式外包给资源强大的外包计算者来计算, 从而可以便捷地获得计算结果.然而外包计算过程中可能会泄露用户的隐私数据, 因此, 在外包计算过程中需要考虑用户数据的隐私性、安全性以及计算结果的可验证性。本文针对高阶多项式的外包计算进行研究, 提出了基于区块链的可验证外包多项式计算方案, 通过区块链智能合约完成外包计算。首先, 提出了一种混淆方法, 能够将原始多项式系数进行盲化, 从而保证多项式的安全性和隐私性。外包计算者将盲化后的两个多项式进行计算, 计算结果上传至星际文件系统(IPFS), 同时挖矿节点仅需计算一个盲化后的多项式; 其次, 设计了一种可快速、简单的验证方法, 智能合约通过用户给出的参数能快速的对外包计算者及挖矿节点返回的计算结果进行验证, 根据验证结果给予相应的报酬。整个方案不需要任何密码学假设, 通过外包计算者和挖矿节点的双重计算, 保证了方案的安全性且效率较高。

关键词 区块链; 外包计算; 多项式; 可验证; 智能合约
中图法分类号 TP309 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2021.01.07

Secure computing outsourcing scheme for polynomial with privacy protection based on blockchain

GUO Zhen[1] , ZHANG Yin[1] , AN Fanglin[1] , ZHAO Kejie[1] , ZHANG Wenjie[1] , YE Jun[1]

[1](School of Computer and Cyberspace Security, Hainan University, Haikou 570228, China)

Abstract With the rapid development of blockchain, blockchain-based outsourcing computing has been widely used. Outsourcing computing allows users with limited resources to outsource complex calculations to outsourced computing with powerful resources for a fee. Conveniently obtain calculation results. However, the user's private data may be leaked during the outsourcing calculation process. Therefore, the privacy, security and verifiability of the calculation results of the user data need to be considered in the outsourcing calculation process. This article is aimed at higher-order polynomials Research on outsourcing calculations, a verifiable outsourcing polynomial calculation scheme based on blockchain is proposed, and outsourcing calculations are completed through blockchain smart contracts. First, an obfuscation method is proposed, which can blind the original polynomial coefficients to ensure the polynomial The security and privacy. The outsourcing calculator calculates the two blinded polynomials, and uploads the calculation results to the Interplanetary File System (IPFS). At the same time, the mining node only needs to calculate one blinded polynomial; secondly, the design A fast and simple verification method. The smart contract can quickly verify the calculation results returned by the outsourced calculator and the mining node through the parameters given by the user, and give corresponding rewards based on the verification results. The entire scheme does not require any password. The academic hypothesis, through the double calculation of outsourced calculators and mining nodes, the security of the scheme and high efficiency are guaranteed.

Key words blockchain; computing outsourcing; polynomial; verify; intelligent contract

受限于本地计算资源, 在大数据分析、机器学习以及人工智能需求不断提高的环境下, 许多用户需求都无法实现。随之出现了一种新的计算范式——外包计算。向外提供服务的计算机通常具备高性能硬件条件, 并按需分成不同的产品类型为用户提供不同的服务, 用户可以用按量付费等方式获取计算

通讯作者: 叶俊, 博士, 研究领域包括密码学, 信息安全, 云计算和隐私保护, Email: yejun@hainanu.edu.cn。
本课题得到海南省自然科学基金(No.619QN193), 海南省教改项目(No. Hnjg 2019-9)支持。
收稿日期: 2020-10-17; 修改日期: 2020-11-30; 定稿日期: 2020-11-30

服务。外包计算模式的出现, 以解决本地资源开销过大的问题。

区块链是现代经济中最新的、有无限前景的技术。在数据处理的信任度、透明度、安全性以及可靠性等方面都能为工业领域提供良好的解决方案。区块链去中心化的特质, 使得系统可以在没有中间人的介入下还能公平公正运行。在区块链挖矿机制的设立下, 参与工作量证明(PoW)的计算节点, 往往都是具有很强计算资源与能力的节点, 这正是那些计算资源匮乏又具有高强度计算要求的本地计算用户所需要的。后续智能合约的发展弥补了区块链不具备图灵完备性的缺点, 它可以定义为一个自我执行的合约, 利用区块链技术以数字方式执行、验证或者促进合约的履行, 并且可以培养合约双方之间的交易可信度, 而无需第三方的参与。我们将外包过程通过智能合约放在区块链上, 其中流程靠智能合约自主严格执行, 区块链存储交易信息。借助区块链和智能合约不可篡改等等优点, 建立一个基于区块链的外包计算模型。

在计算机科学中, 许多科学计算都是基于线性代数的, 但这些计算往往可以简化为多项式的运算, 所以多项式计算对计算机科学起着重要的作用。研究多项式计算不仅可以为计算机这门学科带来进展, 同时在密码学等其他很多学科都会产生意想不到的促进效果。但与此同时, 多项式计算往往伴随到十分复杂的计算过程, 本地计算机无法承担如此庞大的计算资源消耗。外包计算正好是目前解决这个问题的最佳方式。本文将重点研究多项式外包计算的相关问题。

即使区块链中的计算节点可以为多项式计算提供强大的资源助力, 但是还是存在一些亟待解决的问题。第一个问题: 用户数据的保密性问题。用户在外包过程中, 会将数据往外传送, 放在一个专门存储数据的过渡场所, 然后再由计算节点通过区块链领取计算任务, 获得智能合约, 执行智能合约, 完成外包计算, 这就造成了计算节点或者是攻击者搜集、利用用户数据可能性增高, 进而造成用户的直接或间接损失。在用户使用外包计算服务的过程中, 用户不可避免地需要考虑输入隐私安全问题。用户一方面需要从外部获得高效便捷的服务, 另一方面希望自己的隐私数据不会泄漏给不完全可信的计算节点。在以往的外包计算方案中, 为了保护用户数据的安全和隐私, 常用的办法是对数据进行加密预处理, 然后再外包给云服务提供商(CSP)[1]。尽管完全同态加密(FHE)[2]被设计为支持密文上的各种计算, 但是这样也带来了更大的计算资源的消耗[3], 并且使用加密也会使云计算过程变得复杂。

此外, 用户也非常担心数据计算结果的正确性。用户对计算接待你的非绝对掌控导致了一个新的问题产生——计算节点返回的结果或许因为计算量太过庞大, 为节省计算资源, 从而返回一个虚假的计算结果; 又或者是结果在传输过程中, 收到了来自恶意第三方的攻击、篡改等等, 都会导致用户不能按时按需收到正确的计算结果, 从而导致后续工作无法正常进行, 消耗人力物力财力, 以及一些不可预计的可怕后果。有专门研究这一问题的领域——可验证计算[4]。可验证计算力求运用各种方案, 来解决不可信第三方服务器返回结果验证问题。Ye 等人[5]提出了一种新的高阶多项式安全外包算法, 用户可以轻松地验证返回的结果, 并且不需要用户的私人信息(密钥)等, 来验证返回结果的正确性, 对可验证计算领域的研究, 具有一定的启发意义。

为解决上述的外包计算的隐私泄漏问题以及不可信计算节点返回结果无法准确验证的问题, 本文提出了一种基于区块链的具有隐私保护的多项式外包计算方案, 主要贡献如下:

- 提出了基于区块链和智能合约的外包计算解决方案。
- 提出了一种多项式盲化方法, 可以有效保证多项式自身的安全性。
- 提出了高效的可验证方案, 用户只需付出极小计算代价的情况下, 做到了不可信计算节点返回结果的可验证。

本文的其余部分安排如下。在第二部分中, 我们简要讨论了外包计算安全研究、可验证多项式外包计算以及基于区块链的外包计算领域中的相关工作; 在第三部分中, 我们介绍了本方案需要用到的关键技术; 在第四部分中, 我们提出了基于区块链的具有隐私保护的多项式外包计算方案; 在第五部分中, 我们分析了本方案的安全性能; 第六部分, 与相关工作进行了比较; 最后, 对本文工作进行了总结。

1 相关工作

1.1 外包计算安全协议研究

早在 2005 年, Hohenberger 等人[6]就对外包计算安全性进行了定义, 包括效率和可检查性的概念并且还提供了两种实用的外包安全方案——使用两个不受信任的程序进行外包安全的模块化幂运算, 这对后来的云外包安全计算研究进程具有极大的促进作用。Yang 等人[7]给出了一种高效、安全的 RSA 密

码外包计算算法, 能做到很容易地扩展到多个客户端, 可是在假设中没有考虑服务器不诚实的情况。Abadi 等人[7-8]提出了两个在外包私有数据集上进行委派私钥集交集(PSI)计算的协议, 目的也是为了保护外包数据的安全, 使用此协议无需透露任何有关数据和计算结果的信息, 可是该协议的强弱基于假设, 协议不具有稳定性。Wang 等人[9]为抵御来自云服务器提供商的某些攻击, 提出了加密序列比较算法, 结合外包计算流程, 设计了一种新的可计算加密算法, 保护了用户的数据隐私, 并支持密文的有效计算, 但是由于最终结果是共享的, 所以存在一定的可能性会被其他恶意用户恢复, 造成隐私泄露。在将数据外包给云之前, 控制器应使用本地代理掩盖数据, Domingo 等人[10]分析了三种已验证的数据保护方法(数据拆分, 匿名化和加密)各有优缺点。为了实现适用于资源受限的移动用户的安全的基于属性的数据共享(ABDS), Li 等人[11]引入了一种新的在线/离线基于属性加密(ABE)方案, 该方案通过添加系统公共参数来消除计算任务的繁重性, 同时将加密计算开销移到了服务器上, 有效地控制了用户本地计算资源承受负担。Yu 等人[12]借助于完全同态加密和多项式因子分解算法, 提出了一种可验证的加密数据外包计算方案, 改方案在保护外包处理中的用户数据安全的同时, 并允许以零知识对云服务提供商(CSP)处理的计算结果进行公开验证, 但加密计算过程复杂度还可以进行优化。Li 等人[13]提出了一种在单个不可信服务器模型中用于模块化幂运算的安全外包算法, 以及一种生成转换密钥的新方法, 该方案将用户端的转换密钥成本降低为一个常数, 如此获得安全性以及高效性。Jiang 等人[14]在外包环境下, 提出了一种保护双方隐私的协作 k-means 聚类协议, 由于大部分计算都是在云上进行的, 所以任何计算能力较弱的本地客户端都可以运行该协议来实现隐私保护的目的, 但是在通信效率上还需要改进。Domingo 等人[15]为外包计算设计了几种安全协议, 确保了外包保持分离, 从而保持隐私, 但是对于更多类型的数据并没有考虑到。全韩彧[16]提出了一种双方 SIMD 编码协议, 用户先加密, 再允许云服务器和数据分析者联合解密接触向量, 并证明了云服务器不能得到任何的中间和最终计算结果, 但是在服务器计算优化方面还有待提升。张静等人[17]提出了一种基于云服务器外包的安全二方集计算协议, 该协议结合多项式的点值计算和 Boneh 加密系统, 实现了参与者的隐私保护,但是该协议缺乏在恶意云服务器环境下的测试。

1.2 可验证多项式云外包计算安全性研究

针对多项式云外包计算方面的研究, He 等人[18]提出了一种多矩阵函数外包的可验证计算方案, 该方案可公开授权, 存储复杂度小, 无论在客户端计算和在服务器端计算, 都能节省成本。但该方案未能实现公开可验证, 且未能保护用户输入输出的隐私性。Shen 等人[19]提出了一种基于多项式承诺的可公开验证云计算外包方案, 该方案计算结果可公开验证, 且计算成本较低。Guo 等人[20]提出了一种新的可验证的更新数据库外包计算方案, 该方案客户端通过生成两个多项式和, 将原始数据进行伪装, 发送给云服务器, 且云服务器无法获取原始数据, 保护了数据的安全性, 但该方案仅适用于资源受限的客户端。Wang 等人[21]构建了一种具有完全授权的可验证外包计算, 该方案与多项式的阶数无关, 降低了客户端计算成本, 但该方案在标准模型中, 可验证性基于双线性配对和双线性 Diffie-Hellman 指数问题的硬度假设。罗小双等人[22]基于 BGN 和多线性映射, 提出了一种基于双服务器模型的多元多项式公开可验证的扩展计算方案, 该方案虽然确保了输入输出的安全性与专有性, 但是扩展计算效率有待改进。Zhou 等人[23]在有限域上基于扩展欧几里得算法, 构造了一种大规模多项式的外包计算方案。Premkamal 等人[24]提出了一种密文策略属性基加密(CP-ABE)的可验证外包计算方案, 通过大量的数据计算外包给云服务器, 减少了计算开销, 且可验证计算结果的正确性, 但该方案仅适用于云中的大数据隐私和访问控制。Zhang 等人[25]设计了一种基于矩阵向量乘法的多元多项式分解为两步计算的高次多项式外包方案, 该方案可公开授权和可公开验证, 提高了用户输入和多项式函数的私密性。Sun 等人[26]构建了一种可公开验证的保护系数和输出的多项式外包计算方案, 该方案保护了多项式的隐私性, 结果可验证正确性, 但该方案仅支持特定类型的多项式。Zhang 等人[27]设计了一种高效且公开可验证的矩阵乘法外包计算方案, 该方案在密钥生成和计算阶段节省了大量计算开销, 但该方案的可证明安全性是基于 co-CDH 假设下。Zheng 等人在文献[28]中提出了快速的多项式外包方案, 但是只能针对特定的输入进行计算。Liu 等人[29]设计了一种新的安全外包内产品协议, 以实现安全的轻量级单层神经网络, 同时提出了一种保护隐私的分段多项式计算协议, 但是该方案只是考虑了隐私性, 并没有考虑可验证性。Elkhiyaoui 等人[30]引入了两个用于公开可验证计算的加密协议, 允许轻量级客户安全地将单变量多项

式的计算和大型矩阵的乘法外包给云服务器, 但是该方案没有考虑用户输入的隐私性。

1.3 基于区块链的外包计算

Zhang 等人[31]基于区块链设计了一种高效可靠的用于云服务的外包计算解决方案, 该方案不依赖任何第三方的情况下, 实现了可验证计算结果, 且计算成本较低, 但外包计算的应用受到了限制。Huang 等人[32]基于区块链和承诺抽样提出了一种实现公平支付的外包计算方案, 但该方案的外包计算需要通过第三方(比特币脚本)来解决信任问题, 使得外包数据的安全性和隐私性得不到保障。Fan 等人[33]提出了一种基于区块链技术中的智能合约实现安全可靠的外包计算方案, 但该方案中任何用户都能获取计算结果。Krol 等人[34]在可信执行环境(TEE)中构建了基于区块链智能合约的安全外包计算方案, 虽然该方案高效且提高了用户隐私性, 但是 TEE 的开发成本较高。Lin 等人[35]基于区块链技术提出了双线性配对的外包计算(SOBP)方案, 该方案改进了原有 SOBP 局限性, 并有效解决了限制, 且提高了其安全性, 但需要大量的计算成本。Benil 等人[36]使用区块链对电子医疗数据进行外包计算的解决方案, 该方案实现了医疗数据的安全性, 提高了患者敏感数据的隐私性, 但该方案的计算负担较高。表 1 所示为区块链解决外包计算的方案的现状梳理。

表 1 区块链解决外包计算的方案的现状梳理

Table 1 Presents the status quo of outsourcing computing solutions for block chain

研究方向	研究团队	技术方案	存在问题
区块链与云外包	Zhang 等人	不依赖第三方的可验证计算	不便于应用
区块链与承诺抽样	Huang 等人	实现公平支付的外包计算	安全性和隐私性得不到保障
区块链与智能合约	Fan 等人	智能合约实现安全可靠的外包计算	任何用户都能获取计算结果
区块链与智能合约	Krol 等人	可信执行环境(TEE)中构建了基于区块链智能合约的安全外包计算	开发成本较高
区块链	Lin 等人	双线性配对的外包计算(SOBP)方案	计算成本过高
区块链	Benil 等人	区块链对电子医疗数据进行外包计算	计算负担较高

2 预备知识

本节将会介绍一些后续方案会使用到的解决方法和技术要点。

2.1 可忽略函数

在本方案存在一个攻击者消耗大量时间也不能准确定位的函数 $g(x)$, 找出此函数破解该方案的几率非常小, 概率小到“可忽略”, 所以我们引入可忽略函数的定义:

对于一个函数 $negl(x)$, 如果对于任意一个正多项式 $poly(x)$, 存在一个 $N_c>0$, 使得对于所有的 $x>N_c$ 有

$$negl(x)<\frac{1}{poly(x)}$$

我们就称 $negl(x)$ 是可忽略的。

2.2 秦九韶算法

我国南宋数学家秦九韶将增乘开方术进行了推广, 以求解任意高次方的多项式的值。该算法看似简单, 其最大的意义在于把求 n 次多项式的值转化成求 n 个一元多项式的值。其中心思想就是简化多项式的计算, 以减少中央处理器的运算时间。

设有 $n+1$ 项的 n 次函数

$$f(x)=a_nx^n+a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\cdots+a_2x^2+a_1x+a_0$$

将前 n 项提取公因子 x, 得

$$f(x)=(a_nx^{n-1}+a_{n-1}x^{n-2}+a_{n-2}x^{n-3}+\cdots+a_2x+a_1)x+a_0$$

再将括号内的前 $n-1$ 项提取公因子 x, 得

$$f(x)=((a_nx^{n-2}+a_{n-1}x^{n-3}+a_{n-2}x^{n-4}+\cdots+a_2)x+a_1)x+a_0$$

如此反复提取公因子 x, 最后将函数化为

$$f(x)=(((a_nx+a_{n-1})x+a_{n-2})x+\cdots+a_1)x+a_0$$

令

$$\begin{aligned}f_1&=a_nx+a_{n-1}\\f_2&=f_1x+a_{n-2}\\f_3&=f_2x+a_{n-3}\\&\cdots\cdots\\f_n&=f_{n-1}x+a_0\end{aligned}$$

最后 f_n 为所求。

用平常的方式去计算 n 次多项式的值, 需要进行 $\frac{(n^2+n)}{2}$ 次乘法, 如果用秦九韶算法中迭代的方式计算幂则需要 $2n+1$ 次乘法。消耗的存储空间上,

前者需要 x 占用的字节的 $2n$ 倍空间, 后者需要 x 占用的字节的 n 倍空间。

2.3 区块链与智能合约

区块链从本质上说是一种新型的数据库, 并且当新的区块连接到区块链上时, 几乎不可能更改或是删除它。区块链技术是结合分布式网络, 透明、可靠的链型结构, 不可改变、稳定的技术。工作量证明(PoW)是安全的算法, 挖矿是解决 PoW 协议所带来的计算挑战的过程。参与挖矿的节点必须使用 PoW 协议才能将新的区块添加到区块链中。区块中存储着区块链节点的交易信息, 并存储着前一个区块块头的哈希值, 所以能做到供所有节点审查的情况下, 也能做到区块数据防篡改。

智能合约是在存储在区块链上的具有图灵完备性的协议。它由唯一的地址、一组可执行函数和状态变量组成。智能合约将在链上的每个节点按已建立的顺序自动独立、严格地执行。

2.4 星际文件系统

星际文件系统(InterPlanetary File System, 缩写 IPFS)是一个旨在创建持久且分布式存储和共享文件的网络传输协议[37], 是一种内容可寻址的对等超媒体分发协议。IPFS 是一种分布式文件系统, 并提供了一个高吞吐量的块存储模块, 结合了分布式散列表, 并没有单点故障, 不需要节点相互信任。而且支持 FUSE 与 HTTP 的访问方式。

因为区块链上的区块存储容量一般只有 3~4M, 不能存储大容量数据, 而 IPFS 相比区块链更适合存储和处理大型文件, 且 IPFS 在存储文件后可抛出该文件对应的 IPFS 地址, 区块链上只需存储该地址, 即可追溯该文件。如此大大扩展了区块链的应用范围, 将本地文件添加到该文件系统后, 便可向全世界的用户提供文件访问功能。

2.5 可验证外包计算

通过可验证计算方案, 用户可以将经过处理后的函数 $f(x)$ 的计算交给外包计算者, 在客户端可以验证返回结果的正确与否。

关于可验证计算的现有实现方案, 可用加密算法配套的外包计算协议, 正式定义分基本是由四个算法组成: 密钥生成(*KeyGen*)、问题生成(*ProGen*)、计算(*Compute*)以及验证(*Verify*)。可验证的计算方案 由以下算法定义:

KeyGen$(g,r)\rightarrow(PK_g,SK_g,EK_g)$ 。密钥生成算法根据函数 g 和随机参数 r 的生成一对密钥对——公钥 PK_g 和私钥 SK_g, 还有一个经过混淆后的加密函数, EK_g 以及公钥 PK_g 提供给外包计算者, 私钥 SK_g 则由用户自己保存在客户端。

ProGen$_{SK_g}(x)\rightarrow(\sigma_x,VK_x)$ 。问题运行算法将会由用户客户端运行, 使用私钥 SK_g 对输入参数 x 进行加密, 以生成 σ_x, 用于提供给外包计算者计算, 并且得到验证密钥 VK_x, 用户持有。

- 用户将加密后的函数 EK_g 以及输入 σ_x 发送给外包计算者, 服务器计算后返回 σ_y 。

Verify$_{SK_g}(VK_x,\sigma_y)\rightarrow(y\cup\perp)$ 。使用密钥 SK_g, 验证密钥 VK_x 以及计算输出 σ_y, 验证 σ_y 是否正确, 正确输出 y, 否则输出 $\perp$ 。

3 基于区块链的可验证多项式外包计算方案

3.1 外包计算模型

对于多项式函数:

$$f(x)=a_0+a_1x+a_2x^2+\cdots+a_nx^n \bmod P$$

其中 P 是一个大素数(2048bit)。已知系数 $(a_0,a_1,\cdots,a_n)$ 及自变量 x, 客户需要通过在区块链上具有强大计算能力的外包计算者去求出函数值 $f(x)$, 并对最终结果进行验证。

由于需要计算的多项式系数和自变量数据集的庞大, 且计算量很高, 客户端计算机的数据处理能力达不到计算要求, 所以需要向区块链上计算功能强大的外包计算者寻求帮助, 由外包计算者进行计算出值, 其计算过程如下:

外包计算者根据秦九韶算法:

$$\begin{aligned}f(x)&=a_0+a_1x+a_2x^2+\cdots+a_nx^n\\&=a_0+x\left(a_1+x\left(a_2+\cdots+x\left(a_{n-1}+a_nx\right)\right)\right)\end{aligned}$$

迭代计算出每一步的值:

$$\begin{aligned}f_1(x)&=a_{n-1}+a_nx\\f_2(x)&=a_{n-2}+a_{n-1}f_1(x)\\f_3(x)&=a_{n-3}+a_{n-2}f_2(x)\\&\vdots\\f_n(x)&=a_0+a_1f_{n-1}(x)\end{aligned}$$

最后将计算结果 $[f_1(x),f_2(x),\cdots,f_n(x)]$ 上传至星际文件系统(IPFS), 并且把存储的数据地址放进智能合约里, 挖矿节点根据每一步的迭代计算结果进行验证, 验证通过外包计算者和挖矿节点才可以得

到报酬。

由于考虑整个区块链部分节点是不可信的, 为了用户的安全和隐私考虑, 不能将所要计算的多项式系数泄漏出去。本文采取的方案是:

在客户端将预处理数据进行盲化, 使得盲化后的多项式不能被区块链上其他的节点获得原多项式函数系数;

然后再通过智能合约送至区块链, 外包计算者节点在领到计算任务时, 会根据智能合约内容进行计算, 将计算结果发布出来;

这时挖矿节点和客户端节点分别用各自的方式进行验证, 为防止外包计算者作弊而进行不诚实的计算, 或是挖矿节点与外包计算者节点共谋欺骗用户, 在智能合约里, 客户端对计算结果验证后, 如果没有通过验证, 客户端具有一票否决权, 要求重新更换挖矿节点进行验证, 新的挖矿节点如果发现计算结果有问题, 就能发现外包计算者节点具有欺诈性, 就不会给与其报酬。下面讲上述的过程细化成 12 个步骤, 具体的外包计算模型如图 1 所示。

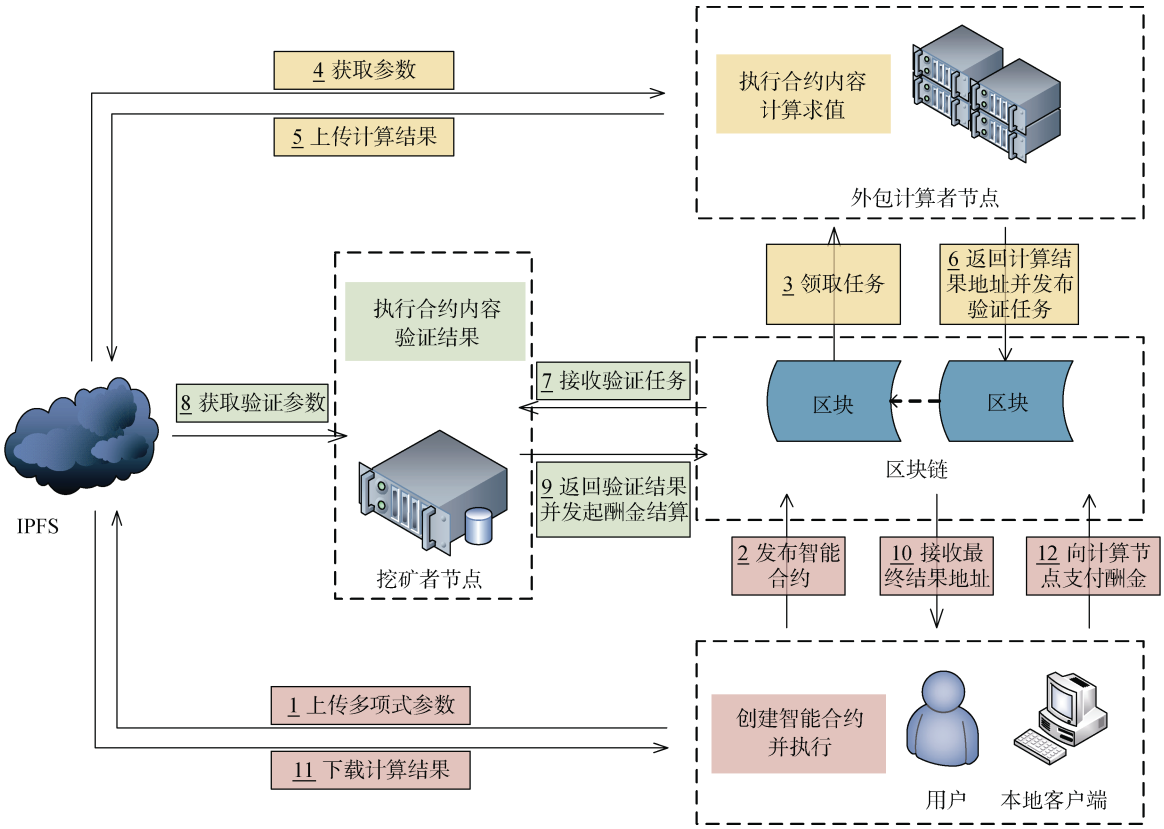

图 1　外包计算模型

Figure 1　Outsourcing calculation model.

3.2　计算方案

为了对多项式

$$f(x)=a_0+a_1x+a_2x^2+\cdots+a_nx^n \mod p$$

进行保护, 我们构造以下多项式:

$$g(x)=bx+b^2x^2+\cdots+b^{n-1}x^{n-1}$$

其中 $b\in_R Z_p$。

$g(x)$ 是一个是首项为 bx, 公比为 bx 的等比数列, 所以在已知 x 的情况下, 用户可以轻松求出 $g(x)$ 的值:

$$g(x)=\frac{bx\left(1-b^{n-1}x^{n-1}\right)}{1-bx}$$

在得到 $g(x)$ 最终值的情况下, 就可以将 $f(x)$ 分别盲化为如下:

$$\begin{aligned}F_1(x)&=f(x)+g(x)+mx^n+r_1\\&=c_0+c_1x+c_2x^2+\cdots+c_nx^n\end{aligned}$$

和

$$\begin{aligned}F_2(x)&=f(x)+(x-k)g(x)+r_2\\&=d_0+d_1x+d_2x^2+\cdots+d_nx^n\end{aligned}$$

其中 $m,k,r_1,r_2\in_R Z_p$。

用户将盲化后的多项式 $F_1(x)$ 和 $F_2(x)$ 的系数和自变量上传至 IPFS 里, 并将其在文件系统里的地址整装进智能合约里。智能合约的伪代码如图 2 所示: 当用户将智能合约发布到区块链中, 这时会有闲置的外包计算节点收到任务来进行完成如图 3 所示:

1：发布合约
2：输入：获取 x 及 $F_1(x)$ 和 $F_2(x)$ 的参数的文件地址 x(addr), c(addr), d(addr)
3：通过市场特定的报酬函数求出的佣金 w

图 2　发布合约
Figure 2　Release contract

1：接收合约
2：输入：通过文件地址 x(addr), c(addr), d(addr) 获取 x 及 $F_1(x)$ 和 $F_2(x)$ 的参数 c_i 和 d_i
3：定义两个 n 维数组 F_1 和 F_2
4：$F_1(1)=c_{n-1}+c_n x$
5：$F_2(1)=d_{n-1}+d_n x$　//初始化首变量
6：for　i=2，3,…，n　　//秦九韶算法迭代求值
7：$F_1(i)=c_{n-i}+c_{n-i+1}F_1(i-1)$
8：$F_2(i)=d_{n-i}+d_{n-i+1}F_2(i-1)$
9：end
10：将 F_1 和 F_2 放进 IPFS 文件系统里，并将存储地址 F_1(addr)和 F_2(addr)放在合约里上传至区块链上

图 3　接收合约
Figure 3　Receive contract

当计算结果已经放在区块链上时, 区块链上的挖矿节点和客户端分别进行独立验证, 由于两个多项式均是由同一个多项式盲化而来, 所以本方案采用只随机选取其中一个多项式进行验证, 这里以验证 F1 为例, 如图 4 所示。

这里我们假设需要计算 $x=x_0$ 时候的值, 用户将 $x=x_0$ 通过智能合约发送到区块链上, 最后通过挖矿节点验证将得到的数据 $F_1(x_0)$ 和 $F_2(x_0)$, 分别对其进行去盲化计算:

$$f_1(x_0)=F_1(x_0)-g(x_0)-mx_0^n-r_1$$

和

$$f_2(x_0)=F_2(x_0)-(x_0-k)g(x_0)-r_2$$

最终用户只需要比较去盲化后的结果, 如果

$$f_1(x_0)\neq f_2(x_0)$$

则说明计算结果是错误的, 就可以判断出外包计算者和挖矿节点是否诚实。如果验证通过, 客户端就可以接受最终的计算结果, 即 $f(x_0)=f_1(x_0)=f_2(x_0)$。

这里会有一个用户节点与挖矿节点博弈的过程, 如图 5 所示。

1：　挖矿节点接收合约对结果进行验证
2：　输入：通过文件地址 x(addr), c(addr)获取 x　和 c
3：　$F_1(x)$ 的参数 c_i，通过 F_1(addr)获取 F_1
4：　If
5：　$F_1(1)$!= $c\_(n-i)+c\_(n-i+1)x$
6：　返回验证失败
7：　else
8：　随机选取 s(s<n)个点
$e_1,e_2,\cdots,e_s\in\{1,2,\cdots,n\}$
9：　for　$i=e_1,e_2,\cdots,e_s$　//依次迭代求值判断
10：If　$F_1(i)$!= $c_{n-i}+c_{n-i+1}F_1(i-1)$
11：返回验证失败
12：　end
13：返回验证成功

图 4　结果验证
Figure 4　Results verification

1：if　挖矿节点验证通过
2：　if　客户端验证通过
3：　　向外包计算者和挖矿者发放计算报酬 w
4：　else
5：　　重新选取新的挖矿节点进行验证
6：　　if　新的挖矿节点验证通过
7：　　向外包计算者和新的挖矿者发放报酬 w
8：　　else 只向挖矿者发放报酬 v
9：else　　//挖矿节点验证失败
10：　if　客户端验证通过
11：　　重新选取新的挖矿节点进行验证
12：　　if　新的挖矿节点验证通过
13：　　向外包计算者和挖矿者发放报酬 w
14：　　else 只向挖矿者发放报酬 v
15：　else 只向挖矿者发放报酬 v

图 5　双验证的博弈
Figure　Double-authentication game

多项式可验证外包计算模型如图 6 所示。

4　安全性证明

4.1　私钥的隐私性

定理 1. 区块链上的节点能够随机破解出用户的私钥 b，m 和 k 的概率均是 $\frac{1}{p}$，能够随机破解出 $r_i(i=1,2)$ 的概率均是 $\frac{1}{p}$。

证明. 由于区块链上的节点获得的多项式是经过盲化后的 $F_1(x)$ 和 $F_2(x)$ 形式如下:

$$F_1(x)=c_0+c_1x+c_2x^2+\cdots+c_nx^n$$

$$F_2(x)=d_0+d_1x+d_2x^2+\cdots+d_nx^n$$

对于所要计算的每一项 c_i 和 d_i，区块链上的节点在得知下面的规律时，即下述这种形式:

$$F_1(x)=(a_0+r_1)+\sum_{i=1}^{n-1}(a_i+b^i)x^i+(a_n+m)x^n$$

$$F_2(x)=(a_0+r_2)+\sum_{i=1}^{n}(a_i+b^{i-1}-kb^i)x^i$$

而不知道具体各个参数

$$a_i(i=0,1,2,\cdots),b,m,k,r_1,r_2$$

的具体值，所以区块链上的节点可以随机性猜测，而导致破解出各个参数的概率为:

对于 $r_i(i=1,2)$，被区块链上的节点随机猜测出真实值的概率是:

$$\Pr\{A(r_i)=1\}=\frac{1}{p}(i=1,2)$$

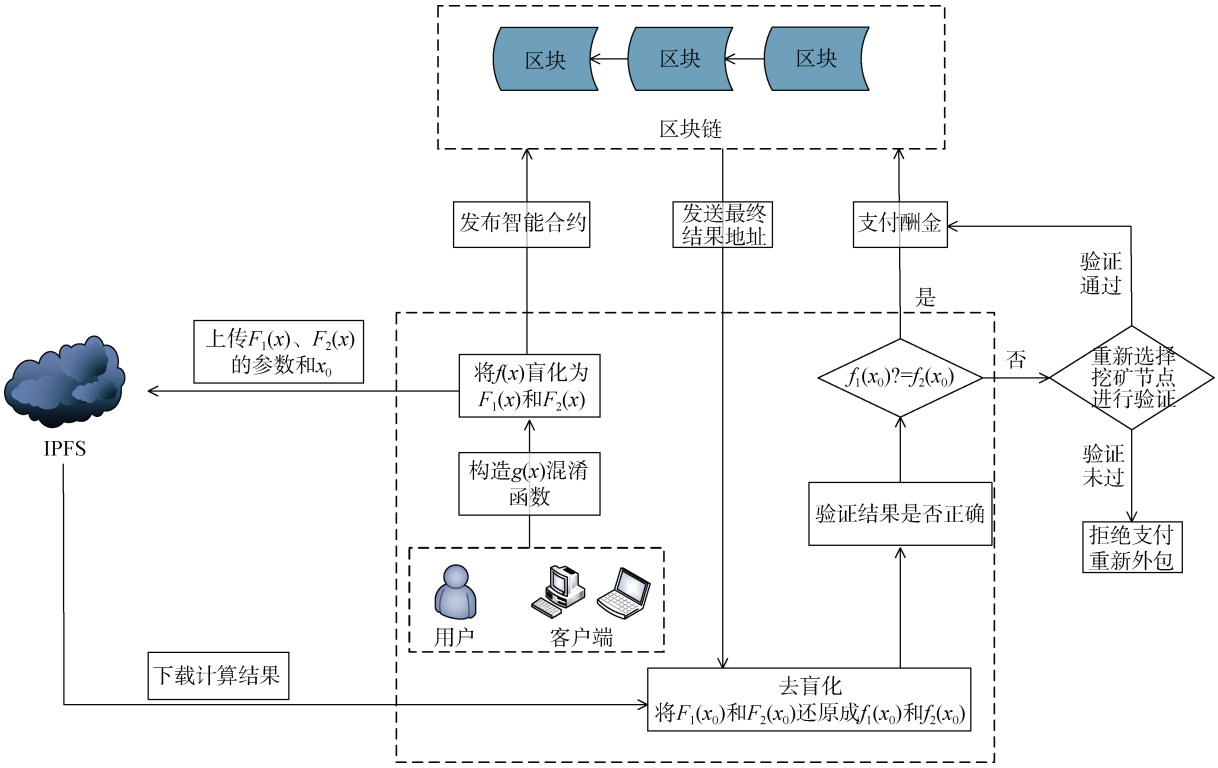

图 6　多项式可验证外包计算模型

Figure 6　Polynomial verifiable outsourcing computing model.

4.2　输入、输出的隐私性

定理 2. 区块链上的节点能得到 $f(x)$ 的概率是 $\frac{1}{p^3}$，区块链上的外包计算者节点伪造能通过客户端验证的结果 R 的概率是 $\Pr\{A(R)=1\}=\frac{1}{p}$。

证明. 根据盲化方法，得到密钥 $[b,m,r_1]$ 可通过 $F_1(x)$ 计算出 $f(x)$，或者是在得到密钥 $[k,b,r_2]$ 可通过 $F_2(x)$ 计算出 $f(x)$，所以其总的概率为:

$$\Pr\{A(f(x))=1\}$$

$$=\max(\Pr\{A(b,m,r_1)=1\},\Pr\{A(k,b,r_2)=1\})$$

$$=\max\left(\frac{1}{p^3},\frac{1}{p^3}\right)=\frac{1}{p^3}$$

假如外包计算者节点通过伪装给出两个值想要通过客户端验证，根据去盲化规则:

$$f_1(x_0)=F_1(x_0)-g(x_0)-mx_0^n-r_1$$

$$f_2(x_0)=F_2(x_0)-(x_0-k)g(x_0)-r_2$$

外包计算者只需要使得:

$$F_1(x_0)-F_2(x_0)$$

$$=g(x_0)-mx_0^n-r_1f_2(x_0)-(x_0-k)g(x_0)-r_2$$

便可通过客户端的验证，而在这一过程中外包计算者节点需要破解出所有的用户私钥 $[b,m,k,r_1,r_2]$，其中对于 r_1 和 r_2，外包计算者节点其只需要根据已知客户端提供的 c_0 和 d_0 的值及其构造规则:

外包计算者节点即可通过 r_1 或 r_2 的一个通过求解 a_0 反推出其中另一个的值。所以由以上可知外包计算者节点通过伪装假解并通过客户端验证的概率是

$$\Pr\{A(b,m,k,r_i)=1\}=\frac{1}{p^4}$$

另一方面, 如果外包计算者节点能直接猜测到 $F_1(x_0)-F_2(x_0)$ 的值, 也可以伪造能通过验证的结果。猜中这个值的概率为 $\frac{1}{p}$。

因此,

$$\Pr\{A(R)=1\}=\max\left(\frac{1}{p^4},\frac{1}{p}\right)=\frac{1}{p}。$$

4.3 计算结果的可验证性

定理 3. 计算结果的正确性是可验证的。

证明. 假设整个区块链的现有节点共有 S 个, 其中与外包计算者可共谋的挖矿节点个数为 N, 那么外包计算者节点与挖矿节点共谋 C 的概率为:

$$\Pr\{A(C)=1\}=\frac{N}{S}$$

当外包计算者节点通过不共谋的挖矿节点而使用假解通过验证 NC 时, 这种情况下, 只有能够通过计算其中一个多项式的真实解, 这时的概率为:

$$\Pr\{A(NC)=1\}=\frac{1}{2}$$

所以使用假解通过挖矿节点 V 的概率为:

$$\Pr\{A(V)=1\}=\frac{N}{S}+\frac{1}{2}$$

外包计算者节点根据盲化后的式子按照程序进行去计算, 结果出来以后, 挖矿节点不仅要进行验证, 客户端还要进一步验证最终的计算结果 $f_1(x_0)=f_2(x_0)$ 是否成立, 这样外包计算者节点通过双重验证而得到报酬 W 的概率为:

$$\begin{aligned}\Pr\{A(W)=1\}&=\Pr\{A(R)=1\}\times\Pr\{A(V)=1\}\\&=\frac{1}{p}\left(\frac{N}{S}+\frac{1}{2}\right)\end{aligned}$$

由于素数 p 是个很大的数, 使得上述概率变得极低, 这就可以完全杜绝外包计算者节点的欺诈性, 使得计算结果具有很强的验证性。如果挖矿节点和客户端都验证成功, 即可判断外包计算者节点是诚实的, 进而确保得用户到了真实的值。

5 方案比较

由于文献[28-30]是解决外包计算所使用的在安全性, 隐私性或是计算成本来说最优良的算法, 下面我们将本文方案与文献[28-30]的方案进行比较如表 2 所示。

文献[28]的方案更加注重结果的可验证性, 而没有对用户数据进行隐私保护, 这导致其在客户端的计算复杂度为 $O(n)$,云服务器的计算量复杂度也近似为 $O(n)$。所设方案没有能够达到隐私性和安全性。

文献[29]的方案更加所构造的盲化函数是一个普通的函数, 所以虽然其隐私保护做得很好, 但是为后期用户去盲化计算增添了麻烦, 使得客户端的计算量达到了达到了 $O(n)$, 而云服务器的计算代价也是 $O(n)$。为了增加隐私性而并没有注重结果的可验证性, 没有考虑到云服务器的欺骗性。

文献[30]方案使用欧几里得算法, 将原多项式除以一个二次的多项式, 从而得到两个多项式, 最后利用双线性对计算结果进行验证, 实现了可验证性, 但是没有做到对隐私的保护, 客户端的计算复杂度为 $O(1)$, 由于云服务器需要进行累乘幂的积, 并且还要计算多项式的值, 所以其云服务器的计算复杂度为 $O(3n)$。

本文方案使用特殊方法盲化多项式, 不仅能够很好的保证了用户隐私的安全性和防欺诈性, 也使得在计算量方面大大的减少, 客户端计算量主要在于求出 $g(x)$ 的值, 由于加法不计入计算量内, 计算的复杂度在于求 $b^n x^n$, 采用二分法进行递归计算可以使得计算复杂度变为 $O(\log_2 n)$, 挖矿节点验证复杂度为 $O(s)$(s 为该方案验证次数, 由安全系数决定), 而外包计算者是针对两个多项式的计算复杂度为 $O(2n)$, 由于是基于双重验证, 相比上述三个方案安全性方面做到了很强的优化。

表 2 方案比较

Table 2 Scheme Comparison

	Computation cost for verify	*Computation cost for outsourcing calculator*	*Privacy protection*	*verifiability*
Scheme[28]	$O(n)$	$O(n)$	Yes	No
Scheme[29]	$O(1)$	$O(2n)$	Yes	No
Scheme[30]	$O(1)$	$O(3n)$	No	Yes
Our Scheme	$O(\log_2 n)$	$O(2n)$	Yes	Yes

6 总结

外包计算具有广泛的应用前景, 然而外包计算中的数据隐私性和计算的可验证性是亟待解决的问题。本文基于区块链智能合约提出了一种支持可验证的多项式外包计算方案。用户需要将盲化后的多项式函数和函数输入上传至 IPFS, 然后在区块链上创建一个智能合约, 外包计算者从智能合约中获取计算任务, 计算完结果并上传, 提供给用户地址, 用户找到地址的计算结果进行验证; 同时挖矿节点通过执行智能合约, 保证不与用户和外包计算者交互的前提下执行验证过程, 将验证结果反馈给用户。用户接收到外包计算者和挖矿节点的计算结果进行解密验证, 将去盲化后的多项式与之进行结果对比, 若对比结果两者相同, 则用户接受计算结果, 给予相应的报酬; 反之, 用户拒绝计算结果, 不给予相应的报酬。上述返回的计算结果都是公开可验证。通过安全性和隐私性的分析, 与已有的方案对比, 所提方案的安全性和隐私性较高, 更具有高效性。用户可验证计算结果的正确性, 且用户的计算量负担远比直接计算多项式函数小。

参考文献

[1] Wang C, Ren K, Wang J. Secure and practical outsourcing of linear programming in cloud computing[C].*2011 Proceedings Ieee Infocom. IEEE*, 2011: 820-828.

[2] Gentry C, Boneh D. A fully homomorphic encryption scheme[M]. *Stanford: Stanford university*, 2009.

[3] Van Dijk M, Gentry C, Halevi S, et al. Fully Homomorphic Encryption over the Integers[C]. Advances in Cryptology – EUROCRYPT 2010. 2010: 24-43.

[4] Chen X F, Li J, Weng J, et al. Verifiable Computation over Large Database with Incremental Updates[J]. *IEEE Transactions on Computers*, 2016, 65(10): 3184-3195.

[5] Ye J, Zhou X L, Xu Z, et al. Verifiable Outsourcing of High-Degree Polynomials and Its Application in Keyword Search[J]. *Intelligent Automation and Soft Computing*, 2018, 24(1): 41-46.

[6] Hohenberger S, Lysyanskaya A. How to Securely Outsource Cryptographic Computations[M]. Theory of Cryptography. 2005: 264-282.

[7] Yang X Y, Luo X S, Wang X, et al. Improved Outsourced Private Set Intersection Protocol Based on Polynomial Interpolation[J]. *Concurrency and Computation: Practice and Experience*, 2018, 30(1): e4329. DOI:10.1002/cpe.4329.

[8] Abadi A, Terzis S, Metere R, et al. Efficient Delegated Private Set Intersection on Outsourced Private Datasets[J]. *IEEE Transactions on Dependable and Secure Computing*, 2019, 16(4): 608-624.

[9] Wang X F, Zhang Y Q. E-SC: Collusion-Resistant Secure Outsourcing of Sequence Comparison Algorithm[J]. *IEEE Access*, 2018, 6: 3358-3375.

[10] Domingo-Ferrer J, Farràs O, Ribes-González J, et al. Privacy-preserving Cloud Computing on Sensitive Data: A Survey of Methods, Products and Challenges[J]. *Computer Communications*, 2019, 140/141: 38-60.

[11] Li J, Zhang Y H, Chen X F, et al. Secure Attribute-based Data Sharing for Resource-limited Users in Cloud Computing[J]. *Computers & Security*, 2018, 72: 1-12.

[12] Yu X X, Yan Z, Zhang R. Verifiable Outsourced Computation over Encrypted Data[J]. *Information Sciences*, 2019, 479: 372-385.

[13] Li Z D, Li W M, Jin Z P, et al. An Efficient ABE Scheme with Verifiable Outsourced Encryption and Decryption[J]. *IEEE Access*, 2019, 7: 29023-29037.

[14] Jiang Z L, Guo N, Jin Y B, et al. Efficient Two-Party Privacy-Preserving Collaborative k-means Clustering Protocol Supporting both Storage and Computation Outsourcing[J]. *Information Sciences*, 2020, 518: 168-180.

[15] Domingo-Ferrer J, Sánchez D, Ricci S, et al. Outsourcing Analyses on Privacy-protected Multivariate Categorical Data Stored in Untrusted Clouds[J]. *Knowledge and Information Systems*, 2020, 62(6): 2301-2326.

[16] Quan H Y. Research on Cloud Outsourcing Ciphertext Query and Calculation [D]. *XiDian Univercity*,2019 .
(全韩彧. 云外包密文查询和计算研究[D].西安电子科技大学,2019).

[17] Zhang J, Luo S S, Yang Y X, et al. Private Sets Intersection Protocols Based on Cloud Computing[J]. *Journal of Beijing University of Posts and Telecommunications*, 2019, 42(2): 13-18.
(张静, 罗守山, 杨义先, 等. 安全两方集合交集云外包计算协议[J]. *北京邮电大学学报*, 2019, 42(2): 13-18.)

[18] He Y, Zhang L F. Multi-matrix verifiable computation[J].*Cluster Computing*,2020.

[19] Shen J, Liu D Z, Chen X F, et al. Secure Publicly Verifiable Computation with Polynomial Commitment in Cloud Computing[M]. Information Security and Privacy. 2018: 417-430.

[20] Guo Z, Li H, Cao C J, et al. Verifiable Algorithm for Outsourced Database with Updating[J]. *Cluster Computing*, 2019, 22(3): 5185-5193.

[21] Wang Q, Zhou F, Peng S, et al. Verifiable Outsourced Computation with Full Delegation[C].*International Conference on Algorithms and Architectures for Parallel Processing. Springer, Cham*, 2018: 270-287.

[22] Luo X S, Yang X Y, Li C, et al. Publicly Verifiable Outsourced Computation Scheme for Multivariate Polynomial Based on Two-server Model[J]. *Journal of Computer Applications*, 2018, 38(2): 321-326.
(罗小双, 杨晓元, 李聪, 等. 基于双服务器模型的可公开验证多元多项式外包计算方案[J]. *计算机应用*, 2018, 38(2): 321-326.)

[23] Zhou Q, Tian C L, Zhang H L, et al. How to Securely Outsource the Extended Euclidean Algorithm for Large-scale Polynomials over Finite Fields[J]. *Information Sciences*, 2020, 512: 641-660.

[24] Premkamal P K, Pasupuleti S K, Alphonse P J A. A New Verifiable Outsourced Ciphertext-policy Attribute Based Encryption for Big Data Privacy and Access Control in Cloud[J]. *Journal of Ambient Intelligence and Humanized Computing*, 2019, 10(7): 2693-2707.

[25] Zhang L F, Safavi-Naini R. Protecting Data Privacy in Publicly Verifiable Delegation of Matrix and Polynomial Functions[J]. *Designs, Codes and Cryptography*, 2020, 88(4): 677-709.

[26] Sun J M, Zhu B R, Qin J, et al. Confidentiality-Preserving Publicly Verifiable Computation Schemes for Polynomial Evaluation and Matrix-Vector Multiplication[J]. *Security and Communication Networks*, 2018, 2018: 1-15.

[27] Zhang X Y, Jiang T, Li K C, et al. New Publicly Verifiable Computation for Batch Matrix Multiplication[J]. *Information Sciences*, 2019, 479: 664-678.

[28] Zheng F F, Tang C M. Verifiable Outsourcing of Polynomial Evaluation[J]. *China Sciencepaper*, 2018, 13(5): 537-541.
(郑芳芳, 唐春明. 可验证的多项式外包计算[J]. *中国科技论文*, 2018, 13(5): 537-541.)

[29] Liu X M, Deng R H, Yang Y, et al. Hybrid Privacy-preserving Clinical Decision Support System in Fog-cloud Computing[J]. *Future Generation Computer Systems*, 2018, 78: 825-837.

[30] Elkhiyaoui K, Önen M, Azraoui M, et al. Efficient Techniques for Publicly Verifiable Delegation of Computation[C]. *The 11th ACM on Asia Conference on Computer and Communications Security - ASIA CCS '16*, 2016: 119-128.

[31] Zhang Y H, Deng R H, Liu X M, et al. Blockchain Based Efficient and Robust Fair Payment for Outsourcing Services in Cloud Computing[J]. *Information Sciences*, 2018, 462: 262-277.

[32] Huang H, Chen X F, Wu Q H, et al. Bitcoin-based Fair Payments for Outsourcing Computations of Fog Devices[J]. *Future Generation Computer Systems*, 2018, 78: 850-858.

[33] Fan K, Bao Z J, Liu M X, et al. Dredas: Decentralized, Reliable and Efficient Remote Outsourced Data Auditing Scheme with Blockchain Smart Contract for Industrial IoT[J]. *Future Generation Computer Systems*, 2020, 110: 665-674.

[34] Krol M, Psaras I. SPOC: Secure Payments for Outsourced Computations[C]. *2018 Workshop on Decentralized IoT Security and Standards,* 2018.

[35] Lin C, He D B, Huang X Y, et al. Blockchain-based System for Secure Outsourcing of Bilinear Pairings[J]. *Information Sciences*, 2020, 527: 590-601.

[36] Benil T, Jasper J. Cloud Based Security on Outsourcing Using Blockchain in E-health Systems[J]. *Computer Networks*, 2020, 178: 107344.

[37] Finley K. The inventors of the internet are trying to build a truly permanent web[EB/OL] https://www.wired.com/2016/06/inventors-internet-trying-build-truly-permanent-web/?mbid=nl_62016_p2

郭祯，1981 年出生。博士，副教授。研究方向包括密码学，信息安全，云计算和隐私保护。

赵科杰，1996 年出生。硕士研究生。主要研究方向包括区块链安全，信息安全，机器学习。

安方林，1996 年出生。硕士研究生。主要研究兴趣是信息安全。

张银，1996 年出生。硕士研究生。主要研究方向包括信息安全和机器学习。

张文杰，1999 年出生。本科。主要研究方向包括云计算，信息安全。

叶俊，1980 年出生。博士。研究领域包括密码学，信息安全，云计算和隐私保护。