委托计算下基于区块链的公平支付方案

李沓[1,2]，田有亮[1,2,3]，向康[1,2]，高鸿峰[1,4]

（1. 贵州大学计算机科学与技术学院，贵州 贵阳 550025；2. 贵州大学密码学与数据安全研究所，贵州 贵阳 550025；
3. 贵州省公共大数据重点实验室，贵州 贵阳 550025；4. 贵州大学网络与信息化管理中心，贵州 贵阳 550025）

摘 要：传统委托计算中，由于参与者双方的自利行为，存在委托方抵赖和计算方不诚实计算等问题。在支付阶段，需采用第三方来保证支付的公平性，从而导致额外的开销且存在泄露参与者隐私的风险。利用博弈论分析了支付过程中双方达到纳什均衡解，提出一种基于比特币时间承诺的公平支付协议。首先，利用比特币时间承诺技术保证参与者支付的公平性；其次，利用区块链去中心化的特性来取代第三方服务保护各方隐私且实现责任溯源；最后从安全性和正确性对方案进行分析，证明了参与者在支付过程中诚实选择行为策略。所提方案不仅解决了传统委托计算中公平支付的难题而且保护了参与方的隐私。

关键词：委托计算；时间承诺；博弈论；公平支付
中图分类号：TP393
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2020046

Block-based fair payment scheme under delegation computation

LI Ta[1,2],TIANYouliang[1,2,3], XIANG Kang[1,2], GAO Hongfeng[1,4]

1. College of Computer Science and Technology, Guizhou University, Guiyang 550025, China
2. Institute of Cryptography & Date Security, Guizhou University, Guiyang 550025, China
3. Guizhou Provincial Key Laboratory of Public Big Data, Guiyang 550025, China
4. Network and Information Management Center, Guizhou University, Guiyang 550025, China

Abstract: In the traditional delegation computation, due to the self-interested behavior of both participants, there were some problems such as the denial of the principal and the dishonest calculation by the computing party. In the payment phase, a third party was required to ensure the fairness of the payment, which led to extra costs and risks of disclosure of participants' privacy. Game theory was used to analyze the Nash equilibrium solution between the two parties in the payment process, and a fair payment protocol based on time commitment of bitcoin was proposed. Firstly, the bitcoin time commitment technology was used to ensure the fairness of payment. Secondly, the decentralized feature of block chain was used to replace the third-party service to protect the privacy of all parties and realize the source of responsibility. Finally, the scheme was analyzed from the perspective of security and correctness, which proved that the participants choose behavioral strategies honestly in the payment process. The scheme proposed in this paper not only solved the problem of fair payment in traditional delegation computation, but also protected the privacy of participants.

Key words: delegation computation, time commitment, game theory, fair payment

---

收稿日期：2019−10−17；**修回日期**：2020−02−15
基金项目：国家自然科学基金资助项目（No.61662009，No.61772008, No.U1836205）；贵州省教育厅科技拔尖人才支持基金资助项目（No.[2016] 060）；贵州省科技重大专项计划基金资助项目（No.20183001）；教育部—中国移动科研基金研发基金资助项目（No.MCM20170401）；贵州大学培育基金资助项目（No.[2017]5788）；贵州省科技计划基金资助项目（No.[2019]1098）；贵州省科学技术基金资助项目（No.[2008]2121）

Foundation Items: The National Natural Science Foundation of China (No.61662009, No.61772008, No.U1836205), Guizhou Provincial Department of Education Science and Technology Top Talent Support Project (No.[2016]060), Science and Technology Major Support Program of Guizhou Province (No.20183001), Ministry of Education China Mobile Research Fund Project (No.MCM20170401), Guizhou University Cultivation Project (No.[2017]5788), Science and Technology Program of Guizhou Province (No.[2019]1098), Science and Technology Foundation of Guizhou Province (No.[2008]2121)

1　引言

随着大数据、云计算的迅速发展，数据的处理及存储能力受到越来越多的重视。一些自身能力有限的委托方往往需要将复杂的计算任务委托给计算能力强的计算方。而在实际环境中，无论是委托方还是计算方，都是理性参与者，都希望让自己的利益最大化。

传统委托计算过程中，需要对计算方返回的结果进行验证从而保证结果的准确性。Xu 等[1]提出在不使用昂贵的完全同态加密的情况下，诚实但好奇的第三方可以帮助验证外包计算任务的结果，而不需要学习计算任务或其结果。通过在参数系统模型中结合新颖的承诺协议和加法同态加密来保护计算任务和来自不受信任的第三方验证者的结果。Wang 等[2]为了解决传统的 VC（verifiable computation）问题，提出了一种可验证的外包计算，具有完全委托 FD-VC（verifiable out sourced computation with full delegation），通过将预处理委托给云，大大降低了客户端的计算成本。Zhao 等[3]使用加同态加密技术和混淆电路构造了可验证计算方案。现有的对于委托计算结果的验证方案大多采用第三方或者处理速度较慢的加密技术。虽然保证了结果的正确性，但会泄露参与方的隐私而且极大地增大了委托方的开销。

而在实际环境中，由于参与者的自利性，众多学者利用博弈论研究委托计算中参与者之间的关系，从而保证计算方诚实进行计算从而取缔第三方验证过程。Yin 等[4]基于博弈论研究了委托计算的验证复杂问题，提出了基于比特币和 Micali-Rabin 的随机向量表示技术的公平理性委托计算协议。该协议不仅解决了传统委托计算验证复杂的问题，而且保证了诚实参与者的利益。Li 等[5]利用博弈委托代理理论，构造委托计算博弈模型并结合全同态加密技术构造理性委托计算协议保证了参与者双方的利益。Dong 等[6]利用博弈论和智能合约技术来激励云之间的约束、背叛和不信任，这样理性云就不会串通和欺骗。在没有串通的情况下，通过交叉检查来自 2 个云的结果，可以容易地验证外包云对用户计算任务的正确性。

但在委托计算的支付过程中，计算方的作假行为和委托方的抵赖行为会使诚实参与者的利益受到损失。因此，既要保证诚实的参与者的利益，又要对恶意的参与者进行惩罚。目前，越来越多的研究者基于区块链技术及比特币系统来研究支付的公平性。Zhang 等[7]提出了一种基于区块链的公平支付框架 BCpay，用于云计算中的外包服务，保证了参与者的利益不受损失。Andrychowicz 等[8-9]利用比特币系统保证双方安全通信协议的公平性，并设计了一种安全的双方协议，以实现从一方到另一方的“强制”财务转移的功能。Bentov 等[10]研究了安全计算中的公平模型，并展示了在比特币网络中如何在双方和多方环境中实现。Wang 等[11]提出了基于智能合约的可审计的公平支付和实物资产交付协议。利用区块链的可追溯性和可审计性为整个运输中的资产和数据共享提供了有效的方法。Yu 等[12-13]为了防止比特币的双花攻击，设计了公平存款以抵御双重支出。公平存款既保证了支付过程中的付款方因双倍花费而受到处罚，又保证了受害者的损失得到赔偿，保证了在支付过程中的公平性。Baza 等[14]提出了一种基于公共区块链的分散式乘车共享服务 B-Ride，利用智能合约和零知识集成员证明引入时间锁定存款协议来实现位置隐私隐藏以及公平付款。Zhao 等[15]提出了一种名为 SPS（secure pub-sub）的新架构，去除了第三方，基于区块链的公平支付和声誉实现数据的机密性和可靠性、订阅者的匿名性及发布者和订阅者之间的支付公平性。Huang 等[16]提出了一种新的基于比特币的外包计算公平支付方案。因为比特币语法的优点，用户可以直接进行交易而不需要银行。Liu 等[17]为实现加密货币支付收据之间的公平交换，引入了公平交换协议的强时效性概念，并提出了 2 个公平的收款协议实例，利用区块链的功能来实现强大的及时性。

当前，基于区块链技术的支付方案中主要利用了比特币系统的去中心化特性，保证了支付过程中参与者的公平性。但多数方案中是针对结果验证后进行奖惩，不仅耗费参与方的代价，而且验证过程的效率较低。博弈论在研究经济体制中有独特的地位，因此利用博弈论分析支付过程的纳什均衡解可以很好地保证双方诚实选择行为策略，从而提高委托计算的效率，降低委托方的成本。

本文结合博弈论与区块链技术提出了基于区块链的公平支付方案，实现了委托计算支付过程下的纳什均衡解。所提方案首先保证了参与方诚实执

行策略，其次保护了参与方的隐私，最后实现了对诚实参与者的奖励以及恶意参与者的惩罚。本文的具体工作如下。

1) 利用博弈论刻画了委托计算中参与方在支付过程中的博弈模型，并给出了唯一的纳什均衡解。

2) 利用区块链技术提出委托计算过程中计算方的信誉值模型，并实现不可篡改。

3) 提出基于比特币时间承诺技术的公平支付协议，实现委托计算下支付的公平性。

4) 对协议的安全性与性能进行分析，证明协议的安全性，并且保证参与者在支付过程中诚实选择行为策略。

2 准备知识

2.1 委托计算

委托计算[18]主要是指委托方由于自身计算能力不足或资源受限而无法计算一个复杂任务，将复杂任务委托给一个计算能力强的计算方，由计算方返回一个正确计算结果而委托方支付相应报酬。但在实际环境中，由于参与者双方的自利行为，支付过程中存在计算方作假以及委托方抵赖等行为。

传统委托计算主要分为两类，分别是基于复杂性理论构造方案和基于密码技术构造方案。传统的委托计算方案主要利用 PCP（probabilistic checking of proof）定理、全同态加密、混淆电路等技术进行方案构造，通常假设委托方是诚实的，对计算方返回结果进行公开验证。而在现实环境中，由于参与者有一定的偏好取向，不能保证参与者都是诚实的，因此出现了理性委托计算。

理性委托计算假设参与者都是理性的，其目的是最大化自身效用，利用博弈论分析参与者的行为策略从而获得纳什均衡解，使参与者双方诚实选择行为策略，保证委托计算结果的正确性及参与者双方的公平性。

2.2 比特币交易

比特币[19]由中本聪于 2008 年提出，比特币没有发行机构，它依据特定算法，通过大量的计算产生，通过整个网络中的所有节点达成共识来确认并记录所有的交易行为。

比特币交易[8]Tx 的最常见形式为

$$((y_1,a_1,\sigma_1),\cdots,(y_n,a_n,\sigma_n),(v_1,\pi_1),\cdots,(v_m,\pi_m),t)$$

Tx 的输入是三元组 $(y_1,a_1,\sigma_1),\cdots,(y_n,a_n,\sigma_n)$，其中 y_i 是某个先前事务 Tx_{y_i} 的哈希值，a_i 是 Tx_{y_i} 输出的索引，σ_i 称为输入脚本。Tx 的输出是一对对的列表 $(v_1,\pi_1),\cdots,(v_m,\pi_m)$，其中 v_i 是 Tx 的第 i 个输出值，π_i 是输出脚本。t 是时间锁定，表示 Tx 仅在时间范围 t 内有效。

2.3 比特币时间承诺

Andrychowicz 等[8-9]基于比特币系统构建了一种定时承诺，其中提交者必须在特定时间范围内揭示其秘密，否则提交者需要支付罚款。

比特币时间承诺方案由 $\mathrm{CS}(S,C,d,t,s)$ 表示，在委托计算中此承诺方案在计算方 S 和委托方 C 之间执行，其中计算方 S 充当提交者，委托方 C 充当接收者，s 表示计算方与委托方各自的秘密值。具体地，S 承诺秘密并且必须在特定时间 t 之内打开承诺以赎回他的存款 d。否则，存款将支付给 C。承诺方案包括 3 个阶段。

1) 承诺阶段 $\mathrm{CS.commit}(S,C,d,t,s)$

2) 打开阶段 $\mathrm{CS.open}(S,C,d,t,s)$

3) 惩罚阶段 $\mathrm{CS.fine}(S,C,d,t,s)$

2.4 博弈论

博弈论[20]是研究一些人或者团体在某种特定的环境下如何进行决策及决策均衡问题的理论，在委托计算的过程中，由于委托方和计算方都有自利行为，因此，可用博弈论来分析双方的行为。

定义 1 博弈表达的基本形式[21]由局中人集合 P、策略空间 Y 和效用函数 u 这 3 个要素组成，即 $G=\{P,Y,u\}$，其中 $P=\{P_1,\cdots,P_n\}$，$Y=\{Y_1,\cdots,Y_n\}$，$u=\{u_1,\cdots,u_n\}$。效用函数 $u_i:Y\to R$（R 代表实数空间），它表示第 i 个局中人在不同组合下所得的效益。

3 支付博弈模型

在委托计算的支付过程中，由于参与者双方的自利行为，使实际结果出现偏差。对于委托方而言，只想得到计算的正确结果而不想向计算方支付服务费，这将导致计算方即使提供了正确的结果依然得不到相应的奖励。对于计算方来说，只想得到奖励而不想提供相应的计算服务，这将导致委托方在支付了相应的服务费之后不能得到正确的结果。本文基于博弈论的角度分析了委托计算过程中支付的博弈模型，其支付博弈树如图 1 所示，其中，S 表示计算方，U 表示委托方，S_i 表示计算方的收益，U_i 表示委托方的收益。

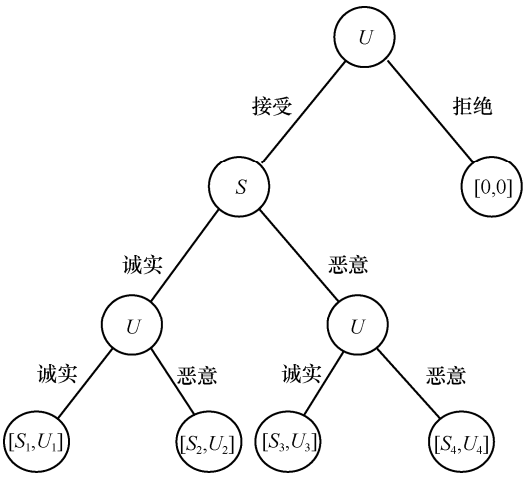


图 1　支付博弈树

假设存在一次委托任务，使委托方得到计算结果后的收益为 P，委托方需要支付的服务费为 T，计算方在收到任务后完成正确结果需要耗费的成本为 S。由于参与者双方都有自利行为，为了防止参与者双方做出欺骗的行为，假定委托方在发布任务的时候需要支付押金 Q，计算方在接受任务时需要支付押金 R，其中，$Q>P>T$，$S<T<R$。基于博弈模型的分析，可以得到如表 1 所示的委托方和计算方的交付效用矩阵，根据参与方各自的行为得到最终的效用函数。

表 1　支付效用矩阵

委托方	计算方	
	诚实	恶意
诚实	$(P+Q-T,T+R-S)$	$(Q+R+T,-R)$
恶意	$(-(Q+T),Q+R+T-S)$	$(-Q,-R)$

由效用矩阵可以看出，委托方和计算方在此次委托任务过程中的唯一纳什均衡解是双方都做出诚实的行为，只有这样双方的利益才不会有损失。无论哪一方做出恶意行为，做出恶意行为的一方将会损失自身的利益。若双方都做出恶意行为，则双方都将受到惩罚，损失自己的利益。

4　系统模型

本节主要介绍系统模型，系统模型如图 2 所示。在此模型中，计算方首先上传自己的信誉值到区块链上，委托方查看区块链根据不同的任务需求选择相应的计算方进行任务发布。然后计算方与委托方共同创建公共存款交易，若参与者双方诚实执行策略，则基于公共存款交易创建支付交易完成委托计算的支付。若参与者存在不诚实行为，则根据公共存款交易创建打开惩罚交易实现对诚实参与者的奖励以及恶意参与者的惩罚。

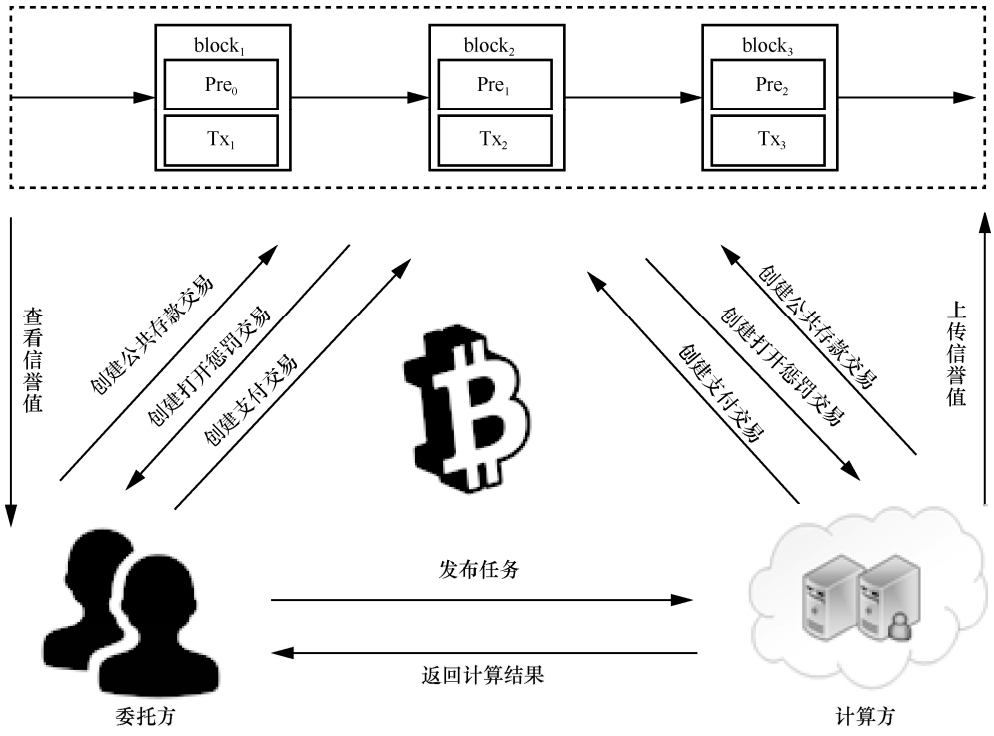


图 2　系统模型

4.1 计算方信誉度模型

每个计算方基于它的交易历史或者工作性质都会形成相对应的信誉值。在本文中，由于计算任务需求的不同往往需要不同的计算方。在委托计算的过程中，计算方信誉度的取值由于在社会网络中的关系很难量化，所以本文基于计算方的交易历史进行信誉值评估。

4.1.1 本地信誉值

计算方在一次交易后有 2 种状态，分别是诚实评价和不诚实评价。每一次交易完成后计算方都将计算自己的信誉值。由于委托的任务有相应的要求，故不同的任务对应不同的难度系数。在一次交易中，任务要求越高，难度越大，其复杂度系数越大，反之则越小。由此，计算方的本地信誉值计算模型为

$$\text{Lcred}=\frac{H(t)}{T(t)}D,\quad 0<D\leqslant 1$$

其中，Lcred 表示计算方的本地信誉值，$T(t)$ 表示计算方总的交易次数，$H(t)$ 表示计算方总交易次数中诚实交易的次数，D 为一次交易中任务的复杂度系数。

4.1.2 全局信誉值

在一次委托计算任务完成后，计算方计算自己的本地信誉值并与上一次的全局信誉值进行运算，然后得到此次委托计算的全局信誉值并进行签名上传到区块链上，此全局信誉值计算模型可以表示为

$$\text{Gcred}_i=\text{Sig}_s(\text{Lcred}\,\|\,\text{Gcred}_{i-1})$$

其中，Gcred 表示计算方的全局信誉值。

信誉值是计算方可以得到计算任务的基础，委托方根据不同的任务需求选择相应的计算方进行委托。计算方的信誉值越高，其被选择的概率越大。全局信誉值是将计算方的信誉值形成可信链，在每一次任务完成后计算方都需要更新自己的全局信誉值并重新上传。

4.2 任务发布

委托方由信誉值链可查看计算方的全局信誉值，然后根据任务难度选择相应的计算方发送任务进行委托。任务公告如图 3 所示。

假设委托方 U 具有密钥对 $(\text{pk}_u,\text{sk}_u)$，则 $\text{Sig}_u(m)$ 表示与 sk_u 相关联的消息 m 上的 ECDSA 签名，$\text{Ver}_u(m,\beta)$ 表示消息 m 上关于 pk_u 的 ECDSA 签名 β 的验证结果。因此，$\text{Sig}_u(\text{task-claim})$ 表示委托方 U 对任务公告的签名；name 表示任务的名称；requirement 表示任务的要求；complexity factor 表示任务的复杂系数；K 表示任务完成后计算方应获得的报酬；T 表示任务完成的截止时间。

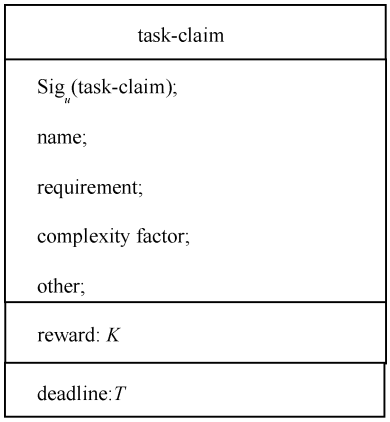


图 3 任务公告

委托方发送任务成功后，创建委托方存款交易 TxU，值为 d。此交易包含委托方的签名及委托方的秘密用于创建公共承诺交易，如图 4 所示。

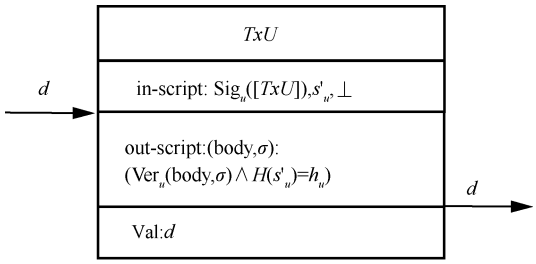


图 4 委托方存款交易

4.3 计算方接收任务

计算方接收来自委托方发出的任务公告，根据公告内容计算此次委托计算的代价。若计算代价大于所得报酬，则计算方不予计算；若计算代价与所得报酬相等，则计算方可以选择接受或者拒绝；若计算代价小于所得报酬，则计算方接收此任务。如果计算方没有在规定时间内提交任务，需要支付一定的赔偿。因此当计算方接受任务时需要向区块链创建一个存款交易 TxS，此存款交易如图 5 所示。计算方计算代价模型如下。

设 price 表示计算方的总开销。有

$$\text{price}<A\rightarrow\text{accept}$$
$$\text{price}=A\rightarrow\text{accept / refuse}$$
$$\text{price}>A\rightarrow\text{refuse}$$

其中，A 表示此次委托计算的服务费；accept 表示

计算方接受此次任务； refuse 表示计算方拒绝此次任务。

计算方接受任务并创建一个存款交易 TxS，此交易包含计算方的签名及计算方的秘密用于创建公共承诺交易。

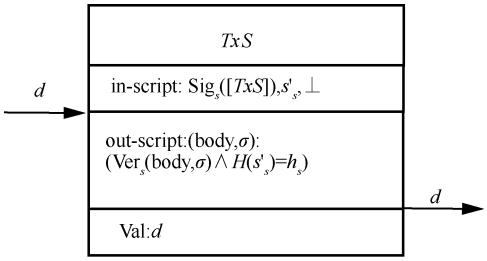


图 5　计算方存款交易

4.4　创建公共承诺交易

委托方与计算方创建公告承诺交易 commit，并最终发布在区块链上。commit 由 (U,S,d,t,s) 表示，其中，U 和 S 是执行协议的双方，分别代表委托方和计算方；d 是 commit 中存款的价值；t 为时间范围，表示参与者双方应该在时间 t 之前履行承诺，打开交易获取存款。承诺阶段由 commit(U,S,d,t,s) 表示，打开阶段由 open(U,S,d,t,s) 表示，惩罚阶段由 punish(U,S,d,t,s) 表示，支付阶段由 pay(U,S,d,t,s) 表示。

4.4.1　准备阶段

U 和 S 各自生成密钥对 u 和 s。S_u 和 S_s 是 U，S 各自的秘密，并且区块链上包含未兑换的交易 TxU 和 TxS，两者值都为 d，分别可以用 S_u 和 S_s 进行兑换。h_u 和 h_s 分别表示对 U 和 S 各自秘密的哈希值，其中，$h_u = H(S_u \| \rho_u)$，$h_s = H(S_s \| \rho_s)$，$\rho_u \leftarrow \{0,1\}^\alpha$ 和 $\rho_s \leftarrow \{0,1\}^\alpha$，$\alpha$ 是安全参数。

4.4.2　承诺阶段

委托方和计算方分别利用 TxU 和 TxS 作为输入计算交易 commit，如图 6 所示，委托方对 commit 进行签名后发送给计算方，如果在广播的最大时延之前计算方都没有收到 commit，则计算方收回自己的押金并退出。若计算方收到了 commit，则计算方对其签名后进行广播。委托方等到 commit 出现在链上，如果在广播的最大时延之前 commit 没有出现在链上，则委托方收回自己的服务费并退出。若 commit 出现在链上，则承诺完成，计算方开始完成相应任务。创建 commit 交易的流程如算法 1 所示。

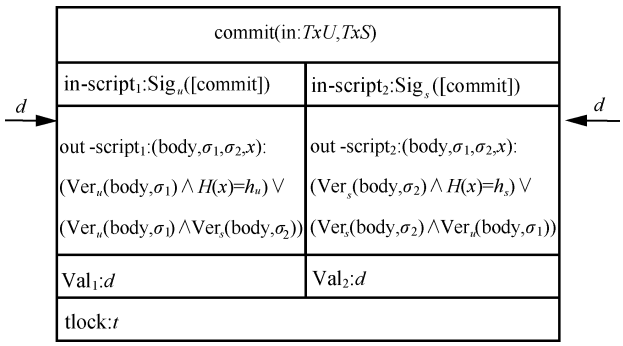


图 6　承诺交易

算法 1　commit

输入　Sig$_u$[TxU]，Sig$_s$[TxS]，S'_s，S'_u，D_u，D_s

输出　commit

$H(x)=H(S'_s||\rho_s)$; $H(x')=H(S'_u||\rho_u)$;

if((Ver$_u$[TxU,β]&&Ver$_s$[TxS,β])==true&&$H(x)==h_s$&&$H(x')==h_u$)

{

goto commit;

}

else return false;

end if

4.4.3　打开阶段

委托方和计算方在 commit 交易情况下承诺对自己的行为进行负责。若出现不诚实的行为，则委托方和计算方依然可以得到各自的服务费或押金。原因如下。

委托方在支付阶段如果不履行诺言，不用自己的秘密打开支付交易。则计算方基于承诺交易 commit 创建计算方打开交易 open$S$ 赎回自己的押金。

计算方在不能提供正确的服务证明情况下却不向委托方支付赔偿，则委托方基于承诺交易 commit 创建委托方打开交易 open$U$ 赎回自己的押金。open$S$ 与 open$U$ 如图 7 所示，创建 open$U$ 与 open$S$ 交易的流程如算法 2 和算法 3 所示。

算法 2　open$U$

输入　Sig$_u$[commit]，Sig$_s$[commit]，S'_s，S'_u，Proof，T

输出　open$U$

$H(x)=H(S'_s||\rho_s)$;$H(x')=H(S'_u||\rho_u)$;

if((Ver$_u$[commit, β] && Ver$_s$ [commit, β])==true&&$H(x)==h_s$&&$H(x')==h_u$)

{

if(Proof==false||T>tlock)

goto open$U$;
end if
}
else return false;
end if

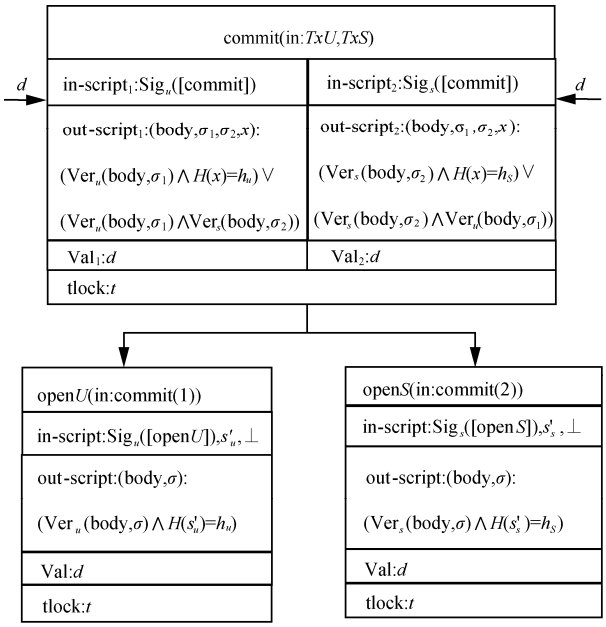


图 7 打开交易

算法 3 open$S$

输入 Sig$_u$[commit]，Sig$_s$[commit]，S'_s，S'_u，Proof，T_1，T_2

输出 open$S$

$H(x)=H(S'_s||\rho_s)$; $H(x')=H(S'_u||\rho_u)$;

if((Ver$_u$[commit,β]&&Ver$_s$[commit,β])==true&&$H(x)==h_s$)
{
if(Proof==true&&T_1<=tlock)
{
if($H(x')\neq h_u$||T_2>tlock)
{goto open$S$;}
end if
}
end if
}
else return false;
end if

4.4.4 惩罚阶段

若在任务截止时间 t 之前，计算方不能给出相应的工作证明，则委托方可以基于 commit 创建委托方惩罚交易 punish$S$，此交易值为计算方所交的押金，在全网公告后，所有链上节点达成共识，则委托方可以基于他的签名和秘密来兑换此交易获得对计算方的罚金。

若计算方的工作证明得到证实，但是委托方在打开阶段抵赖不承认计算方的工作并且不参与兑换打开交易。则计算方可在服务证明证实之后，基于 commit 创建计算方惩罚交易 punish$U$，此交易的值为委托方的押金，待链上节点达成共识后可基于计算方的签名和秘密来兑换此交易。惩罚交易如图 8 所示，创建委托方和计算方惩罚交易的流程如算法 4 和算法 5 所示。

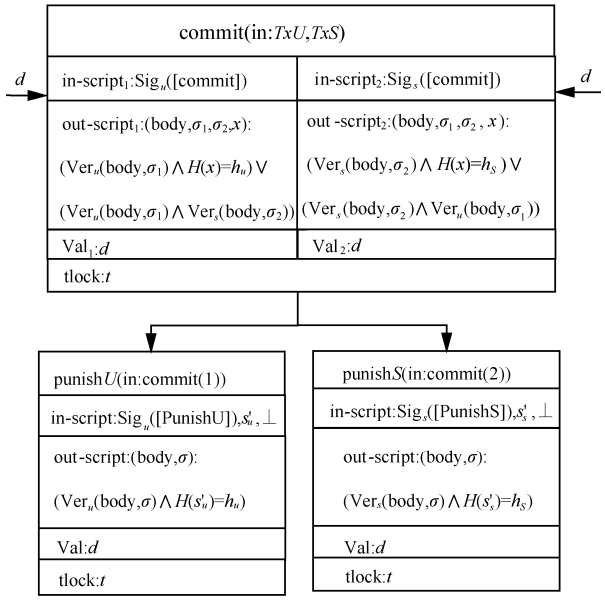


图 8 惩罚交易

算法 4 punish$U$

输入 Sig$_u$[commit]，Sig$_s$[commit]，S'_s，S'_u，Proof，T

输出 punish$U$

$H(x)=H(S'_s||\rho_s)$;$H(x')=H(S'_u||\rho_u)$;

if((Ver$_u$[commit,β]&&Ver$_s$[commit,β])==true&&$H(x)==h_s$&&$H(x')==h_u$)
{
if(Proof==false||T>tlock)
goto punish$U$;
end if
}
else return false;
end if

算法 5　Punish S

输入　Sig_u[commit]，Sig_s[commit]，S'_s，S'_u，Proof，T_1，T_2

输出　punish$S$

$H(x)=H(S'_s||\rho_s)$; $H(x')=H(S'_u||\rho_u)$;

if((Ver_u[commit,β]&&Ver_s[commit,β])==true&&$H(x)==h_s$)

{

if(Proof==true&&T_1<=tlock)

{

if($H(x')\neq h_u||T_2$>tlock)

{goto punish$S$;}

end if

}

end if

}

else return false;

end if

4.4.5　支付阶段

在计算方提供正确的服务证明之后，委托方和计算方基于 commit 创建支付交易 pay，交易值包括委托方的服务费 d 以及计算方预存的押金 d，委托方和计算方用各自的秘密共同打开此交易，然后计算方获得最后金额。支付交易如图 9 所示，创建支付交易的流程如算法 6 所示。

算法 6　pay

输入　Sig_u[commit]，Sig_s[commit]，S'_s，S'_u，Proof

输出　pay

$H(x)=H(S'_s||\rho_s)$; $H(x')=H(S'_u||\rho_u)$;

if((Ver_u[commit,β]&&Ver_s[commit,β])==true&&$H(x)==h_s$&&$H(x')==h_u$)

{

goto pay;

}

else return false;

end if

4.5　更新信誉值

待一次交易完成后，计算方根据此次交易记录更新自己的信誉值并重新上传到区块链上，等待下一次交易的启动。委托方在发布任务的时候可以从区块链上查看计算方的信誉值。每一次交易过程中委托方都能知道计算方在上一次交易中的状态，这样不仅帮助委托方按需选取相应的计算方，同时使计算方遵守协议规定，做出诚实的行为。

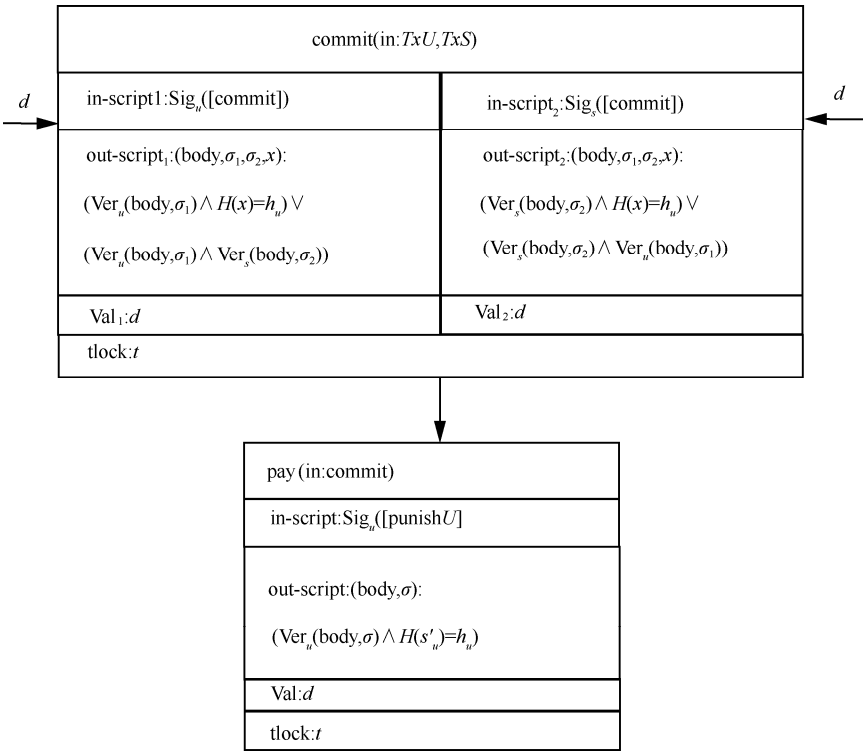


图 9　支付交易

5 协议分析

5.1 安全性分析

本文的安全性主要从以下2个方面来考虑。

定理 1 如果基于ECDSA的签名是不可伪造的且选取的哈希函数是抗碰撞的，那么委托方和计算方的支付是安全的。

证明 委托方和计算方在支付时需要验证各自的签名以及各自秘密的哈希值是否正确。只有签名和秘密的哈希值都正确的情况下才能创建交易获取相关比特币押金。

假设存在敌手 EvE 在一次委托计算中伪造委托方签名$\mathrm{Sig}_{u'}$，伪造计算方签名$\mathrm{Sig}_{s'}$，伪造委托方秘密值为X'，伪造计算方秘密值为Y'。敌手为了获取比特币押金，利用伪造签名以及秘密值创建打开与惩罚交易，EvE的具体操作如下。

1) 获取公共存款交易commit。

2) 使用伪造的委托方签名$\mathrm{Sig}_{u'}$、计算方签名$\mathrm{Sig}_{s'}$及各自秘密值X'和Y'，时间承诺T和计算结果证明Proof。

3) 计算秘密哈希值，$h_{u'}=H(X'\|\rho_u)$，$h_{s'}=H(Y'\|\rho_s)$。

4) 验证签名和秘密值的哈希值是否正确。

$$\mathrm{Ver}_{u'}[\mathrm{commit},\mathrm{Sig}_{u'}]==\mathrm{true}\ \&\ \&h_{u'}=h_u$$

$$\mathrm{Ver}_{s'}[\mathrm{commit},\mathrm{Sig}_{s'}]==\mathrm{true}\ \&\ \&h_{s'}=h_s$$

5) 创建打开与惩罚交易转移比特币押金。

若敌手最终获取了比特币押金则说明敌手伪造的签名可以验证通过，且敌手伪造的秘密值能够通过哈希运算与最终的哈希值相同。这与假设相矛盾，故敌手不可能创建打开或惩罚交易，获取比特币押金。

同理可得，在支付交易验证中，敌手的验证也不可能通过。这意味着敌手创建的支付交易不能在链上被广播，因此交易不被承认，故敌手不能获取服务费。

在链上的所有交易当且仅当签名和秘密值验证通过，才能获得比特币押金。而基于ECDSA的签名是不可伪造的且选取的哈希函数是抗碰撞的，所以对于委托方和计算方而言支付是安全的。

证毕。

定理 2 若基于ECDSA的签名不可伪造，则计算方的全局信誉值是不可篡改的。

证明 由计算方的全局信誉值计算模型可知$\mathrm{Gcred}_i=\mathrm{Sig}_s(\mathrm{Lcred}\|\mathrm{Gcred}_{i-1})$，假设存在敌手伪造计算方的签名$\mathrm{Sig}_{s'}$，则敌手具体操作如下：

1) 敌手通过伪造的签名对计算方的信誉值进行签名，同时上传至区块链。若信誉值在链上广播并最终记录，则说明敌手伪造的签名验证通过。这与假设相矛盾，故计算方的信誉值不可能被敌手上传至区块链。

2) 恶意计算方伪造本地信誉值参与计算全局信誉值。因为计算方的交易历史在区块链上公开可见，若存在恶意计算方伪造信誉值进行计算，则节点验证不能通过，故不能达成共识，全局信誉值不会被广播记录。

证毕。

5.2 正确性分析

定理 3 基于比特币时间承诺的公平支付协议具有正确性，而且可以保证双方达到唯一纳什均衡解。

证明 在承诺阶段若委托方U和计算方S诚实执行该策略，则双方共同创建交易commit，交易commit包含了委托方的押金Q和计算方的押金R，计算方诚实进行计算最终委托方获得收益P，计算方花费成本S，而委托方需要支付的服务费为T。

若委托方U选择诚实策略，计算方S选择恶意策略，则委托方获得效用$Q+T+R$，计算方获得效用$-R$。

若委托方U选择恶意策略，计算方S选择诚实策略，则委托方获得效用$-(Q+T)$，计算方获得效用$Q+R+T-S$。

若委托方U和计算方S均选择恶意策略，则委托方获得效用$-Q$，计算方获得效用$-R$。

若委托方U和计算方S均选择诚实策略，则委托方获得效用$Q+P-T$，计算方获得效用$R+T-S$。

在博弈模型中，由于支付效用有如下关系$Q>P>T$，$S<T<R$，只有当参与方都选择执行诚实策略，委托方U和计算方S才能获得最大效用，此时双方达到唯一纳什均衡解。

由协议分析可知，因为参与者双方皆为理性，故为了使自己的利益最大化，双方都会诚实执行策略，故协议具有正确性。

证毕。

6　性能分析

下面对本文所提方案的时间开销进行评估。本文所提方案时间开销主要包括创建公共承诺交易时间以及创建打开、惩罚交易时间。而比特币系统平均 10 min 产出一个块，因此将创建公共承诺交易时间设为 10 min。即创建公共承诺交易时间如图 10 中曲线 y_1 所示。本文主要考虑委托任务量与耗费时间的关系，在传统委托计算方案中，需要对计算方返回结果进行验证，因此会耗费大量的验证时间，随着任务量的增加，验证时间也随之增加，如图 10 中曲线 y_2 所示。而本文所提方案中理性委托方不需要对返回结果进行验证，仅在创建公共承诺交易时间的基础上增加创建打开交易与惩罚交易的时间，如图 10 中曲线 y_3 所示。对比传统委计算方案，本文所提方案提高了效率。

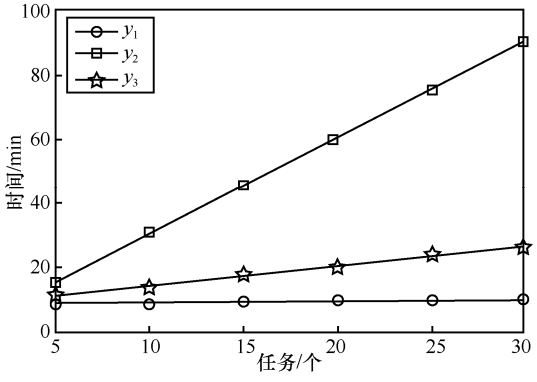


图 10　方案时间开销

下面将本文提出的委托计算公平支付协议与现有的委托计算协议进行比较。表 2 将从委托计算的参与方隐私性、委托方公平性以及计算方公平性等方面与其他方案进行对比，其中“√”代表满足该性能，“×”代表不能满足该性能。

表 2　　本方案与其他方案性能对比

参与方	隐私性	委托方公平性	计算方公平性
Xu 等[1]协议	×	√	×
Yin 等[4]协议	×	√	√
本文协议	√	√	√

Xu 等[1]提出的协议中采用诚实但好奇的第三方帮助验证委托计算任务的结果。通过在参数系统模型中结合承诺协议和加法同态加密来保护计算任务和来自不受信任的第三方验证者的结果，保证了计算方诚实计算结果，即保证了委托方的公平性。但由于方案中采用了第三方来帮助验证计算结果，因此容易泄露参与者的隐私，且无法保证最终计算方可以得到相应的服务费，不能保证计算方的公平性。

Yin 等[4]提出的理性委托计算协议中引入了可信第三方帮助委托方和计算方取回押金。通过委托方和计算方预存押金的形式保证公平性。一旦一方违反承诺，则另一方可以联合可信第三方取回押金来保证协议的公平性。但是协议中依然存在第三方，故容易泄露参与者的隐私。

本文提出的委托计算公平支付协议利用博弈论构建支付博弈模型，并分析了唯一的纳什均衡解，利用效用函数约束参与者双方诚实执行策略，从而保证了委托方的公平性，并取缔了委托计算中结果的验证过程，提高了效率。利用比特币押金保证了计算方的公平性并且不需要第三方进行验证从而保护参与方的隐私。此外，利用区块链的激励机制与计算方的信誉机制相结合提高了委托任务的通信效率。

7　结束语

本文基于博弈论分析了委托计算中支付的公平性，同时利用比特币时间承诺提出了一种在委托计算中保证公平性的支付方案，保证了参与方能够诚实选择行为策略。利用区块链去除了第三方来保证参与者的隐私而且实现责任溯源。如何减少协议的通信复杂度以及确定时间承诺的极限值将是下一步研究的工作。

参考文献：

[1] XU G, AMARIUCAI G T, GUAN Y. Delegation of computation with verification outsourcing: curious verifiers[J]. IEEE Transactions on Parallel and Distributed Systems, 2017,28(3): 717-730.

[2] WANG Q, ZHOU F C, PENG S, et al. Verifiable outsourced computation with full delegation[C]// International Conference on Algorithms and Architectures for Parallel Processing. Berlin: Springer, 2018: 270-287.

[3] ZHAO Q S, ZENG Q K, LIU X M, et al. Verifiable computation using re-randomizable garbled circuits [J]. Journal of Software, 2019, 30(2):209-225.

[4] YIN X, TIAN Y L, WANG H L. Fair and rational delegation computation protocol[J]. Journal of Software, 2018, 29(7):131-140.

[5] LI Q X, TIAN Y L, WANG Z. Rational delegation computation protocol based on fully homomorphic encryption [J]. ACTA Electronica

Sinica, 2019, 47(2):216-220.

[6] DONG C Y, WANG Y L, ALDWEESH A, et al. A ad van moorsel: Betrayal, distrust, and rationality: smart counter-collusion contracts for verifiable cloud computing[J]. IACR Cryptology ePrint Archive, 2018: 489.

[7] ZHANGY H, DENG R H, LIU X M, et al. Blockchain based efficient and robust fair payment for outsourcing services in cloud computing[J]. Information Science. 2018,462: 262-277.

[8] ANDRYCHOWICZ M, DZIEMBOWSKI S, MALINOWSKI D, et al. Fair two-party computations via bitcoin deposits[C]// International Conference on Financial Cryptography and Data Security. Berlin: Springer, 2014: 105-121.

[9] ANDRYCHOWICZ M, DZIEMBOWSKI S, MALINOWSKI D, et al. Secure multiparty computations on bitcoin[J]. IEEE Symposium on Security and Privacy, 2014: 443-458.

[10] BENTOV I, KUMARESAN R. How to use bitcoin to design fair protocols[R]. Cryptology ePrint Archive, (2014-12-19)[2019-10-17].

[11] WANG S P, TANG X X, ZHANG Y L, et al. Auditable protocols for fair payment and physical asset delivery based on smart contracts[J]. IEEE Access, 2019(7): 109439-109453.

[12] YU X J, SHIWEN M T, LI Y, et al. Fair deposits against double-spending for Bitcoin transactions[C]// IEEE Conference on Dependable and Secure Computing. Piscataway: IEEE Press, 2017:44-51.

[13] YU X J, SHIWEN M T, LI Y J, et al. Collusion attacks and fair time-locked deposits for fast-payment transactions in Bitcoin[J]. Journal of Computer Security, 2019,27(3): 375-403.

[14] BAZA M, LASLA N, MAHMOUND M M, et al. B-ride: ride sharing with privacy-preservation, trust andfair payment atop public blockchain[J]. CoRRabs/1906. 09968, 2019.

[15] ZHAO Y Q, LI Y N, MU Q L, et al. Secure pub-sub: blockchain-based fair payment with reputation for reliable cyber physical systems[J]. IEEE Access, 2018(6): 12295-12303.

[16] HUANG H, CHEN X F, WU Q H, et al. Bitcoin-based fair payments for outsourcing computations of fog devices[J]. Future Generation Computing System. 2018(78): 850-858.

[17] LIU J, LI W T, KARAME G O, et al. Toward fairness of crypto currency payments[J]. IEEE Security & Privacy, 2018, 16(3): 81-89.

[18] GOLDWASSER S, KALAI Y T, ROTHBLUM N G. Delegating computation: interactive proofs for muggles[J]. Journal of the Association for Computing Machinery, 2015, 62(4): 1-64.

[19] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system[R]. Bitcoin, (2008-10-31)[2019-10-17].

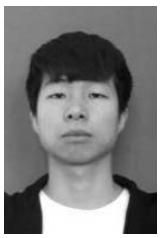
[20] DODIS Y, HALEVI S, RABIN T. A cryptographic solution to a game theoretic problem[C]//Lecture Notes in Computer Science. Berlin: Springer, 2000: 112-131.

[21] OSBORNE M. An introduction to game theory[M]. New York: Oxford University Press, 2004.

[作者简介]

李沓（1998− ），男，贵州盘县人，贵州大学博士生，主要研究方向为密码学与区块链技术。

田有亮（1982− ），男，贵州盘县人，博士，贵州大学教授，主要研究方向为博弈论、密码学与安全协议。



向康（1993− ），男，湖北仙桃人，贵州大学硕士生，主要研究方向为委托计算与机器学习。

高鸿峰（1975− ），男，贵州遵义人，贵州大学副教授，主要研究方向为网络与信息安全。