Operations Research and Fuzziology 运筹与模糊学, 2023, 13(3), 2429-2440
Published Online June 2023 in Hans. https://www.hanspub.org/journal/orf
https://doi.org/10.12677/orf.2023.133244

Hans汉斯

基于区块链的可控监管匿名交易方案

陶俊宏，陈玉玲

贵州大学公共大数据国家重点实验室计算机科学与技术学院，贵州 贵阳

收稿日期：2023年5月11日；录用日期：2023年6月23日；发布日期：2023年6月30日

摘　要

由于区块链技术的公开透明特性，大量交易以明文形式存储在公共网络上，对区块链参与者的隐私保护提出了重大挑战。然而，现有的研究方案大多过于强调匿名性，忽略了交易参与者之间的信息不对称。这导致恶意犯罪者的成本较低，且缺乏可控性和监督，难以在现实环境中应用。本文主要研究两个问题：如何在区块链网络中实现完全匿名交易，以及如何对交易双方的恶意行为进行跟踪和规范，并进行有效的惩罚。本文提出了一种安全、匿名、可审计的交易方案。在该方案中，交易发起方和响应方通过共同计算产生一个隐私地址，并利用承诺机制隐藏交易金额。此外，引入监管方对每笔交易进行验证和跟踪，既实现了可控监管，又降低了交易双方的计算负担。最后，采用挑战-响应机制，使监管方能够识别欺诈者并进行处罚。实验结果与分析表明，所提方案能够在区块链中有效实现匿名交易，适用于恶意模型。

关键词

区块链，匿名交易，可控监管，隐私地址

Secure Anonymous Regulated Transaction Scheme Based on Blockchain

Junhong Tao, Yuling Chen

State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang Guizhou

Received: May 11th, 2023; accepted: Jun. 23rd, 2023; published: Jun. 30th, 2023

Abstract

Due to the public and transparent properties of blockchain technology, a large number of transactions are stored in plaintext on the public network, presenting significant challenges for the privacy protection of blockchain participants. However, most existing research solutions excessively emphasize anonymity and ignore the information asymmetry between the participants of a trans-

文章引用: 陶俊宏, 陈玉玲. 基于区块链的可控监管匿名交易方案[J]. 运筹与模糊学, 2023, 13(3): 2429-2440.
DOI: 10.12677/orf.2023.133244

action. This leads to lower costs for malicious perpetrators and a lack of controllability and supervision, making it difficult to apply in the real-world environment. This paper primarily investigates two issues: how to achieve fully anonymous transactions in the blockchain network, and how to track and regulate malicious behavior by both parties involved in a transaction and impose effective punishment. In this paper, we propose a secure, anonymous, and auditable transaction scheme in which the initiating and responding parties jointly perform calculations to generate a hidden address and utilize a commitment mechanism to conceal the transaction amount. In addition, a regulatory party is introduced to verify and track each transaction, which not only achieves controllable supervision but also reduces the computational burden for both parties involved in a transaction. Finally, a challenge-response mechanism is used to enable the regulatory party to identify cheaters and impose penalties. The experimental results and analysis demonstrate that the proposed scheme can effectively achieve anonymous transactions in a blockchain and is applicable to malicious models.

Keywords

Blockchain, Anonymous Transactions, Controllable Supervision, Stealth Address

1. 引言

随着网络技术的快速发展，区块链技术[1] [2]因其抗篡改、去中心化、公开透明等特点逐渐受到国内外学者的关注，并被广泛应用于各个领域[3] [4] [5]。例如，基于区块链的加密货币被设计为链式的分布式账本，记录了交易双方的地址和交易金额等私人数据，以及其他交易数据。在区块链系统中，交易参与者通常使用公钥地址进行交易，所有数据都记录在分布式账本中[6]。然而，这种分布式账本在区块链网络中是完全透明的，交易关系密切相关，这意味着任何与每笔交易相关的信息都可以被区块链上的任何人访问。如果恶意对手从大量交易记录中获取有效信息，并分析地址和交易记录，则极有可能推断出参与者的身份信息，从而导致参与者身份隐私的泄露和可能的经济损失[7]。此外，在没有监管机构的情况下，恶意节点可以冒充正常节点加入网络，通过侦听网络信息、制造虚假信息等方式破坏交易协议[8]。

事务身份隐私保护可以进一步细分为事务发起者隐私保护和接受者隐私保护。此外，随着区块链技术在业内的日益普及，隐私保护与身份监管之间的冲突日益明显。基于公开链的加密货币采用匿名无访问认证机制，并通过混淆技术提供强大的交易身份隐私保护[9]。虽然这降低了区块链系统中与公共分类帐簿记相关的隐私风险，但它也为非法交易提供了便利[10]。

目前有各种可用的技术来保护事务处理、事务发起者和接收者地址的机密性。例如，零知识证明[11]，环签名[12]和隐形地址协议[13]，Zerocoin 是一种基于零知识证明的加密货币，Zerocoin 通过零知识证明技术隐藏真实交易金额和参与者身份，保证匿名性和机密性[14]。当用户将零币从一个地址发送到另一个地址时，零币系统将零币转换为等价的“零币支出”。这些花费与任何特定的用户身份无关。然后，零币系统利用零知识证明技术证明“零币支出”等于已知数量的“零币存款”，而不透露任何关于交易发起者或接收者身份或交易金额的信息。该方法保证了交易金额和参与者身份的机密性，支持匿名交易。虽然零币提供了更高的隐私性和安全性，但其复杂的零知识证明在交易处理过程中需要大量的计算能力和存储空间。Zerocash 是作为一种对零币的改进而提出的，使用 Pedersen 承诺和非交互式零知识证明来

减少计算需求，尽管它没有解决可控监管的问题[15]。Monero 使用环签名和隐形地址协议来确保交易匿名性、不可追踪性和不可链接性[16]。在群签名算法中，签名者利用自己的私钥和任意 n 个环成员(包括自己)的公钥生成环签名，保护了交易发起者的身份隐私。而隐私地址协议是生成一个一次性的临时交易地址来隐藏真实地址，防止接收者与不同的交易关联，保护交易中的用户身份。有各种可用的技术来保护事务处理、事务发起者和接收者地址的机密性。

最初的隐形地址协议(Stealth Address Protocol, SAP)是由比特币社区的一位匿名成员在2011年提出的，利用承诺方案和交叉签名实现匿名交易，该提议在比特币社区得到了广泛的关注和讨论，并为后来的隐形地址协议的发展奠定了基础[17]。虽然 SAP 可用于保护交易接收者的隐私，但由于需要根据发送者的私钥计算隐形地址，因此缺乏随机性，如果接收者没有及时花掉这笔钱，交易发起者可以收回这笔资金[18]。因此，在 SAP 的基础上提出了双钥隐形地址协议(Dual-Key Stealth Address Protocol, DKSAP)，其中每个用户有两个私钥：一个用于接收交易，另一个用于从接收的交易中提取资金。隐写地址的每次计算都是基于随机数，解决了 SAP 问题，但也带来了较大的计算和存储开销。文献[19]提出了一种无临时密钥泄漏的双密钥不可见地址协议 PDKSAP，通过该协议发送方和接收方维护本地交易记录数据库来记录与其他用户的交易次数。文献[20]提出了一种基于双线性映射的高效双密钥隐藏地址协议 EDKSAP。发送方计算临时事务输出地址，接收方通过双线性映射验证计算结果，提高计算性能。在[21]文献中，作者提出了 DKSAP-IoT，一种基于类似 TLS 会话恢复技术的 DKSAP 改进方案，以提高性能，但是并没有可监管的功能，各参与方的作恶成本较低。

针对现有研究方案存在问题：一是如何在计算开销较低的同时保证完全匿名的安全数据交易，二是在交易中如何识别并追朔作恶行为，并有效的对恶意参与方进行惩罚。本文提出一种基于椭圆曲线密码的可控监督高性能匿名交易方案。在该方案中，我们在 DKSAP 中加入了一个承诺系统，以确保交易金额的机密性。此外，我们还引入了一个监管机构来监控每一笔交易，从而防止交易双方因信息不对称而导致的非法交易。针对恶意行为，设计了挑战－响应机制来跟踪和惩罚违规参与者，所提方案在保持较高交易效率的同时，提供了增强的安全性和隐私保障。

2. 预备知识

2.1. 椭圆曲线密码体制(ECC)

椭圆曲线密码体制(Elliptic Curve Cryptography, ECC)是一种基于椭圆曲线数学的现代广泛使用的公钥加密体制。椭圆曲线的概念最早是在 19 世纪中期由法国数学家 Augustinlouis Cauchy 提出的，他研究了椭圆积分的性质。然而，椭圆曲线在密码学中的应用直到很久以后才被发现。在 20 世纪 70 年代末和 80 年代初[22]，包括 Neal Koblitz 和 Victor Miller 在内的一群数学家独立地提出了将椭圆曲线用于密码学的想法。

椭圆曲线密码(ECC)是一种利用有限域上的椭圆曲线以较小的密钥长度提供强安全性的公钥密码体制。ECC 背后的基本原理是使用椭圆曲线来生成用于加密和解密消息的公钥和私钥对。椭圆曲线密码学的优势是可以在某些情况下使用更小的密钥提供比其他方法更高级的安全性。设有限域 Z_p 有一条椭圆曲线 $E_p(a,b)$，其中 p 为质数，$x,y\in[0,p-1]$：

$$y^2=x^3+ax+b(modp)$$

要求曲线在有限域内处处可导，满足 $4a^3+27b^2\neq 0$。定义 $P+Q=R$ 是椭圆曲线上的加法运算如下描述：任取椭圆曲线上的两点 P、Q，作直线交于椭圆曲线的另一点 R，R 点关于 x 轴的对称点则为 R，如果 P 和 Q 是一个点，那么作 P 点的切线。

同理定义椭圆曲线上的被点运算，以 kP 为例：

$$kP = P + P + \cdots + P$$

kp 代表 $k-1$ 个加法运算。椭圆曲线密码系统的安全性是基于求解椭圆曲线离散对数问题(ECDLP) [6] [7]的难度。这个问题涉及到找到给定一个点 kp 的整数 $k\left(0<k<n\right)$，其中 n 是 $Ep\left(a,b\right)$ 的群阶。

椭圆曲线离散对数问题(ECDLP)：在上述条件中在有限域 F_z 中取一条 E_p，取两点 $\left(P,R\right)$，并满足 $R=kP$。求解 k 的值。定义：

$$Success_A^{DLP} = PR\left[k \leftarrow \left(R,P\right)\right]$$

为破解 ECDLP 问题的概率，对于任意多项式时间算法 A，从概率上 $Success_A^{DLP}$ 是可忽略的。

2.2. 基于椭圆曲线密码的承诺系统

基于 ECC 的承诺方案是一种密码学原语，允许一方(称为提交者)提交一个秘密值，而不会将其泄露给另一方(称为验证者)。该方案由 Torben Pryds Pedersen 于 1992 年首次提出，作为一种确保安全电子通信的方法[23]。

基于椭圆曲线的承诺方案背后的原理是利用椭圆曲线的性质来创建一个单向函数。该函数接收提交者希望提交的秘密值和随机值作为输入，并生成椭圆曲线上的一个点作为输出。提交者将这一点发送给验证者，验证者将其存储为承诺。提交者也会对随机值保密。稍后，当提交者想要揭示秘密值时，他们将秘密值和随机值同时提供给验证者，验证者可以通过使用提供的值计算相同的单向函数来验证承诺的有效性[24]。

基于椭圆曲线密码的承诺方案的实现涉及到椭圆曲线密码。这包括选择一条合适的椭圆曲线，并在曲线上定义一个操作，比如点加法。提交者随机选择一个值，使用单向函数计算曲线上的点，并将结果点发送给验证者。提交者将秘密值和随机值同时发送给验证者，验证者可以利用提交者提供的值计算单向函数来验证是否合法。

2.3. 双密钥隐形地址协议(DKSAP)

双密钥隐身地址协议是对传统的隐身地址进行改进后的协议，双密钥隐身地址通过双密钥的方式更好的保护了交易地址的隐私[21]。双密钥隐私地址协议的工作原理如下所示：

初始化设置：选择椭圆曲线群 G 的 p 阶生成元 g，抗碰撞哈希函数为 $H\left(x\right)$。

密钥生成：接收方随机选取 $s,u \leftarrow Z_p$ 生成密钥对 $\left(s,S\right)$ 和 $\left(u,U\right)$，其中 u 为扫描私钥，s 为花费私钥，U 和 S 分别为对应的公钥，满足条件 $S=g^s, U=g^u$。

共享秘密计算：发送方随机选取 $r \leftarrow Z_n$ 生成临时密钥对 $\left(r,R\right)$ 满足 $R=g^r$，并将 R 发送给接收方。随后发送方和接收方可以计算共享秘密 $c=H\left(g^{ru}\right)=H\left(R^u\right)=H\left(U^r\right)$，其中发送方的计算方式为 $H\left(U^r\right)$，接收方的计算方式为 $H\left(R^u\right)$。

隐身地址计算：发送方计算接收方的隐身地址 P，作为交易的输出地址：$P=g^{c+s}=g^cS$，接收方可以计算 $c+s$ 来进行签名。

交易扫描：在 DKSAP 中，接收方可以共享扫描私钥 x 和支付公钥 S 给代理服务器。这些实体可以通过计算 $P=g^cS$ 得到接收方的隐身地址，从而代表接收方扫描这些区块链交易。但是，他们无法计算隐身地址私钥 $c+s$ 并花费资产。

3. 基于隐蔽地址协议的监督匿名交易方案(SATSAP)

这一节主要对 DKSAP 协议进行改进并构造了一个可以应用在参与方恶意模型下的匿名数据交易方案。在本方案中，不仅可以在区块链上进行安全的匿名交易，而且还可以对参与方中的非法行为进行监管和追踪，经过实验证明，本方案更有希望能适用于现实场景。

总体方案

SATSAP 分为四个阶段：初始化阶段；事务发起阶段；事务验证阶段；挑战应答阶段。

1) 初始化阶段

密钥生成算法$\left(V_{pki},V_{ski}\right),\left(S_{pki},S_{ski}\right)\leftarrow Key.Gen\left(pp\right)$：输入公共参数 pp，交易成员生成自己的公私钥对$\left(V_{pki},V_{ski}\right),\left(S_{pki},S_{ski}\right)$，其中$\left(V_{pki},V_{ski}\right)$称为交易事务的扫描公私钥对，$\left(S_{pki},S_{ski}\right)$为消费公私钥对，满足$\left(V_{pki}=V_{ski}G,S_{pki}=S_{ski}G\right)$，$G$ 为椭圆曲线上的基点，算法 1 如下所示。认证中心 CA 为每个交易成员颁布证书，并储存注册每个用户的公钥信息，以便检验身份的合法性。假设交易中存在 n 个用户，分为有 i 个交易发起方用户 $User_i$ 和有 j 个交易接收方用户 $User_j$，在交易发起方用户中第 i 个用户 $User_i$ 的扫描私钥为$\left(V_{pki},V_{ski}\right)$，花费公钥为$\left(S_{pki},S_{ski}\right)$，其中$i\in\left(1,2,\cdots,n-j\right)$，交易接收方中第 j 个用户 $User_j$ 的扫描公私钥为$\left(V_{pki},V_{ski}\right)$，花费公私钥对为$\left(S_{pki},S_{ski}\right)$，$j\in\left(1,2,\cdots,n-i\right)$。监管方拥有接收方的一半公私钥$\left(V_{pki},V_{ski}\right)$对用于检验交易。接收方拥有$\left(S_{pki},S_{ski}\right)$来对收到的转账进行签名和消费。公私钥对返回成功后，由监管方在区块链上部署特定的保证金智能合约，并由接收方的私钥签名触发智能合约。

输入 系统参数pp, 用户的私钥V_{ski}, S_{ski}
输出 V_{pki}, S_{pki}
1: **for** $i=(1,n)$ **do**
2: $V_{pki}=V_{ski}G$
3: $S_{pki}=S_{ski}G$
4: **else**
5: **false**
6: **返回** $(V_{pk1},V_{sk1}),(S_{pk1},S_{sk1}),(V_{pkn},V_{skn}),(S_{pkn},S_{skn})$

Algorithm 1. Public key generation algorithm
算法 1. 公钥生成算法

在规定的时间锁内，需要各交易参与方们提交一定的保证金到智能合约，最后如果有人被查出作恶行为，则保证金将被没收，并分发给其它诚实的参与者。如果没有任何违法行为，那么就退还保证金。智能合约如算法 2 表示：

合约主体：
参与方：$User_i, User_j$
存款金额：$x$coins
创建时间：xxx.xxx.xxx
截止时间时间：xxx.xxx.xxx
.....
合约流程：
输入： xxx
输出： xxx
1: **for** 每一个用户$User_i$和$User_j$ **do**
2: 冻结$User_i$ 的b coins
3: 冻结$User_j$ 的b coins
4: **if** 完成协议 **then**
5: 归还$User_i$ 的b coins
6: 归还$User_j$ 的b coins

Algorithm 2. Smart contracts for margin
算法 2. 保证金智能合约

为了降低在交易过程中篡改交易金额的风险，并确保最终转账金额与接收方的期望一致，有必要对交易发起方施加承诺要求。承诺要求将事务发起者绑定到商定的交易金额，从而防止对事务数据进行任何未经授权的修改。

例如，假设事务发起者希望转移 m 的金额。为了引入额外的安全层，加入了一个盲因子 r_{cm1}，从而得到承诺值 $C_{mi}=m_1G+r_{cm1}G$。事务发起者提供了盲因子 r_{cm1}，承诺值 C_{mi} 随后被记录在区块链上。在所有事务完成后，将启动承诺披露阶段，在此期间，监督者和事务接收方根据提交的值验证最终收集金额。如果验证过程成功，则认为交易有效，并且将交易信息追加到区块链。

2) 事务发起阶段

密钥生成假设用户 $User_1$ 需要发起一笔交易，收款用户是 $User_2$，金额为 b，其中 $User_1$ 的公私钥对为 $\left(V_{pk1},V_{sk1}\right),\left(S_{pk1},S_{sk1}\right)$，$User_2$ 的公私钥对为 $\left(V_{pk2},V_{sk2}\right),\left(S_{pk2},S_{sk2}\right)$。监管方建立并发布地址维护列表，如表 1 所示：

Table 1. Address maintenance table
表 1. 地址维护表

交易接收方	一次地址	发起方的临时公钥	发起方的承诺值
$User_1$	P_2	R_1	Cm_1
……	……	……	……

在发起交易之前，$User_1$ 应向主管查询最新版本的地址列表。如果 $User_2$ 的信息不存在于地址表中，则可以确定该交易为首次交易，从而需要计算双方的共享秘密。成功通过 RA 的验证后，用户 $User_1$ 生成一个随机数 r_1，用于派生一个临时公钥 $R_1=r_1G$，然后传输给 RA。RA 收到临时公钥后使用用户 $User_2$ 的公钥 V_{pk2}，计算共享密钥 c_1：

$$c_1=H\left(r_1V_{sk2}G\right)=H\left(R_1V_{sk2}\right)=H\left(r_1V_{pk2}\right) \tag{3-1}$$

由于监管方 RA 拥有接收方的验证公私钥对，所以监管方 RA 计算 $User_2$ 的一次地址 P_2，并更新地址表：

$$P_2=c_1G+S_{pk2} \tag{3-2}$$

3) 事务验证阶段

首先 $User_2$ 需要通过查询地址表中的临时公钥 R_1 计算公共秘密 $H\left(R_1V_{sk2}\right)$，并计算一次地址

$$P_2'=H\left(R_1V_{sk2}\right)G+S_{pk2} \tag{3-3}$$

随后，$User_2$ 进行验证检查，确认地址表中的地址 P_2 与 P_2' 是否匹配。这个验证步骤的成功完成使 $User_2$ 成为事务的合法接收者。在协议到期时，交易接收者可以通过利用其私钥 S_{sk2} 继续签署和执行交易，如下公式所示，该公式强调 $User_2$ 是消费密钥的唯一所有者。

$$P_2'=c_1G+S_{pk2}=\left(c_1+S_{sk2}\right)G \tag{3-4}$$

算法 3 的作用在于监管方需要对交易发起方 $User_1$ 最开始承诺的金额进行验证，防止交易发起方进行任何的篡改和作恶行为，如果承诺值验证通过则证明转账的金额是正确金额。那么 $User_2$ 向 RA 发送确认消息，RA 再执行转账事务，算法 3 如下所示。如果监管方此时发现交易事务中有作弊行为则触发进入下一小节的惩罚合约。

合约主体：

参与方：$User_i, User_j$

创建时间：xxx.xxx.xxx

截止时间时间：xxx.xxx.xxx

合约流程：

输入： 系统参数(G, H),用户1输入承诺值的(r_{cm1}, Cm_1),用户2输入应得到的金额b'

输出： 0 or 1

```
while 目前系统时间还未到截止时间 do
    if b'G + r_cm1 H == Cm_1 then
        验证通过，输出1
        else
        验证失败，输出0
        end
```

Algorithm 3. Smart contract to verify the value of the commitment
算法 3. 承诺验证智能合约

4) 挑战应答阶段

如上一节所述，如果 RA 在时间锁定到期时未能收到 $User_1$ 的确认，RA 将发起挑战合同。这种情况会导致四种可能：一是 $User_2$ 故意终止协议没有发送确认信息给 RA；二是在承诺验证智能合约中，$User_1$ 提供的承诺参数错误，导致合约输出为 0；三是 $User_1$ 提交了正确的数据，但是数据在网络传输中丢失了；四是 $User_1$ 和 $User_2$ 的行为完全正确，只是最后发给 RA 的确认信息在网络传输中丢失了。所以，我们接下来设计了一个挑战回应机制来解决上述问题，以达到可控可追溯的效果。

a) RA 在区块链上发布事务，告诉各交易参与方进入挑战回应阶段，并为此事务设置一个时间锁，首先需要 $User_2$ 随时检查区块链上的事务，如果发现进入挑战回应阶段并且 $User_1$ 自身并没有故意终止协议，只是确认信息在传输过程中丢失了，导致监管方 RA 并没有收到确认信息，那么必须 $User_2$ 需要在 RA 规定的时间内在区块链上发布相应的确认事务，否则 RA 会判定是 $User_2$ 作弊。

b) 如果 $User_2$ 觉得是 $User_1$ 的验证数据有误，导致 $User_2$ 没有向监管方 RA 发送确认事务，那么此时 $User_2$ 需要向区块链上公开发布挑战事务 $Tx_{challenge}$，如下图 1 所示，表中输入是交易发起方 $User_2$ 未花费的一小笔钱，输出的是 $User_1$ 的临时地址，数据域中包含了自己计算出的共享秘密 c_1，共享秘密用来证明 $User_2$ 是交易接收方的身份，并且规定了一个时间锁，所有操作需要在时间锁到来之前完成。

交易事务$Tx_{challenge}$
$T_x.Input$：$User_2$一笔未花费的钱 m
$T_x.Output$：交易接收方的临时公钥地址R_1
$T_x.Time$：时间锁
$T_x.Date$：共享秘密c_1

Figure 1. Challenge transaction table
图 1. 挑战事务表

表中输入是交易发起方 $User_2$ 未花费的一小笔钱，输出的是 $User_1$ 的临时地址，数据域中包含了自己计算出的共享秘密 c_1，共享秘密用来证明 $User_2$ 是交易接收方的身份。

c) 同理，$User_1$ 需要随时检查区块链上发布的事务中有没有关于自己的挑战事务，有的话必须发布回应事务 $Tx_{response}$，回应事务表如下图 2 所示。如果在 $User_2$ 规定的时间锁结束之前，依旧没有收到任何来自 $User_1$ 的回应事务，那么直接 RA 直接判定 $User_1$ 作弊，终止协议，扣除保证金。

交易事务$Tx_{response}$
$T_x.Input$：由挑战事务中收到$User_2$的一笔钱m
$T_x.Output$：交易接收方的一次地址P_2
$T_x.Time$：时间锁
$T_x.Date$：承诺值的参数和共享秘密

Figure 2. Response transaction table
图 2. 回应事务表

d) 如果 $User_1$ 并没有作恶，只是数据丢失，或者数据输入错误。那么 $User_1$ 需要证明自己的清白，并及时发布回应事务 $Tx_{response}$。

在此交易中，输入是挑战交易中包含的货币，而输出是接收者的隐形地址 p_2。数据字段包括重新提供的承诺值和共享秘密。$User_2$ 在验证 $Tx_{challenge}$ 中的所有信息后，向监管机构提交确认消息，罚款合同终止，并执行转账和保证金退款。

此时完整的匿名交易协议结束，本方案中的计算过程大致和双密钥对隐身地址方案相同，但是相比之下本方案简化了交易双方之间需要协同计算的计算量。方案中引入了监管方这个主体来对每笔交易事务进行监管，使得每一笔交易都是正确的，安全的并且是匿名的，并且最后提出的“挑战回应”机制可以有效地防止参与方之间的某些非法行为，并且能够有效的惩罚恶意的参与方。

4. 安全性与实验分析

本节介绍我们提出的匿名方案的安全性分析和实验分析，该解决方案假设与 DKSAP 相同的威胁模型。威胁模型考虑了两种类型的恶意对手。第一种类型是外部恶意攻击者，他们试图通过观察区块链上的交易记录来获取有关交易地址或金额的私人信息。第二种类型涉及交易中的不诚实方，他们从事欺诈行为以获得对自己有利的信息。

4.1. 安全性分析

在初始化阶段中，假设存在一个恶意敌手 A'随时观察区块链上的数据信息，由于 $Key.Gen(pp)$公私钥生成算法都是由 CA 派发的，每一个用户所注册的公私钥对以及身份信息都由 CA 进行储存，所以此阶段中，恶意敌手 A'无法通过区块链上的信息来冒充或者窃取任何用户的私钥及公钥信息。在此阶段的最后，交易发起方需要向监管方 RA 提交一个对交易金额的承诺 Cm_i，且 $Cm_i\left(m,r_{cm}\right)=mG+r_{cm}H$，又因为基于 EC 的 Pedersen 承诺本身的安全性定义具有隐私性和绑定性，隐私性指就算恶意敌手 A'能够窃取到 Cm_i 和 r_{cm}，根据椭圆曲线离散对数的困难性，也无法得到隐藏金额的值。绑定性指除非是承诺打开阶段由交易发起方主动向监管方公布 m 和 r_{cm}，否则任何人不能找到另一个 m 和 r_{cm} 来构造等式 $Cm_i'=m'G+r_{cm}H$ 使得 $Cm_i'=Cm_i$。所以在初始化阶段中，本文的方案是安全的。

在交易发起阶段，如果是双方第一次进行交易，那么就交易放起方给监管方需要提交交易接收方的一次性公钥 P_j 的相关信息，监管方验证通过之后在地址表里面进行更新。此阶段假设存在恶意敌手 A'，

交易发起方 $User_1$，交易接收方 $User_2$，监管方 RA。$User_1$ 发起交易需要选取随机数 r_1 计算出自己的临时公钥 R_1，$User_2$ 的验证公钥 V_{pk2} 计算 $User_2$ 的公共秘密 c_1，由于这里的公共秘密计算是通过基于椭圆曲线的密钥交换算法得到的，所以根据椭圆曲线的离散对数难题，恶意敌手 A'在这一阶段只能得到$\left(R_1,V_{pk2},G\right)$，给定点 R_1 和 G，求出 r_1 是不可行的，其中的 V_{pk2} 同理。

在交易验证阶段，由于监管方 RA 可以用 $User_2$ 的花费公钥 S_{pk2} 计算出一次性地址并更新地址表，$User_2$ 计算自己的一次地址 P_2' 并对照地址表的 P_2，如果匹配就向监管方 RA 加密发送需要收到的金额数，如果没有参与方作弊的话那么监管方 RA 就需要触发承诺验证智能合约，本合约的作用是保证交易发起方 $User_1$ 所承诺的金额与交易发起方 $User_2$ 所提供的金额所保持一致，如果一致，那么就执行最后的转账合约和最后的保证金退还合约，将交易发起方 $User_1$ 的金额值转给交易发起方 $User_2$，协议结束。在这个阶段，如果有敌手 A'窃听传输消息，在没有私钥的情况下，也只能得到关于承诺打开阶段的密文，这对承诺系统的安全性并无影响。如果敌手 A'在链下能够得到 $User_2$ 的花费公钥 S_{pk2}，同样根据椭圆曲线的离散对数问题，定点 S_{pk2} 和 G，求出花费私钥 S_{sk2} 是不可行的。

在挑战验证阶段，因为监管方 RA 并没有收到交易双方的确认信息，所以 RA 需要进行此阶段来判断出作恶的参与方。假设恶意敌手 A'将交易发起方 $User_2$ 发送的确认信息给恶意丢弃了，导致监管方 RA 并没有收到 $User_2$ 的数据，所以 $User_2$ 需要在 RA 触发挑战验证合约之后向 RA 证明自己的清白，同理，$User_1$ 也需要证明自己的清白，在这个过程中，由于每一步都进行上链广播，不仅公开可见，并且恶意敌手无法从密文消息中解密出明文，所以此阶段也是安全的。

由于本方案使用了基于椭圆曲线的一次地址协议而不是真实地址来进行交易，就算此地址在区块链上公开，由于椭圆曲线的离散对数难题所以不会泄露出交易参与各方的地址信息，同时利用了承诺系统的不可抵赖性以及绑定性来对需要转账的金额进行隐藏，并引入发布挑战事务，可以快速帮助监管方审查作弊的参与者，这样就可以适用于恶意参与方模型下，最后构造了一个完整的匿名的安全的交易方案。

4.2. 实验结果分析

本小节对本文的方案进行了算法实现和性能测试。测试程序使用了 python 语言，实验环境配置为：AMD Ryzen 5 3600 6-Core Processor 六核处理器，16 GB DDR4 2400 MHz (8 GB + 8 GB)内存，Windows 10 操作系统，NVIDIA GeForce GTX 1660 SUPER (6 GB)显卡，python 版本为 3.8，下面将从交易发起方，交易接收方，总体方案三个方向进行仿真 N = 1，5 和 10 次隐私交易的计算开销对比，代码结果如下图 3 所示。

请注意，本方案将原隐藏地址协议的验证一次地址部分转交给监管方 RA 进行处理，这样的好处是可以让交易接收方减少一半的密钥存储空间，并且减少交易接收方的计算量，在整个过程中，交易接收方需要做的事情仅仅是对承诺做出验证，验证通过后，发送验证成功事务给监管方 RA。从实验结果可以看出，本方案的交易双方的计算开销与原来的方案相比，随着交易次数的增加，计算开销小于 DKSAP 方案中的计算量。

由上述实验结果图 4 和图 5 可以得出结论，本文提出的匿名方案不仅可以在保证完全匿名的场景下进行安全的交易，并且相对于传统的 DKSAP 协议拥有更小的计算开销，后续提出的“挑战-回应”机制还可以保证监管方达到“可控监管”的目的。

随着网络技术的快速发展，尽管区块链为匿名交易带来了诸多便利，但交易参与者的不诚实行为和来自外部攻击者的恶意攻击仍对用户隐私构成挑战。文中基于 DKSAP 提出了一种可在区块链上监督的完全匿名交易方案，并设计了一种“挑战－响应”机制来跟踪和识别交易参与者的恶意行为。最后，实现了一种既能实现完全匿名交易又适用于恶意参与者模型的交易方案。实验结果表明，与原方案相比，所提方案对交易参与者的计算开销相对较小。

```
****************** Stealth Address - Python Implementation ********************

The User1's commit is:  020bb70ecaf3c41c0400607d2f6b99340633f917b369dc11956ba350a8d92c084a

0.004004240036010742
The User2's VPK2 is:  0208c4ca6f582f219547obf602e1bcc08b1b4ca099e058e4fdd212357ba5a0caa0
The User2's SPK2 is:  02e67adb91d2d0b010fe3626d74d53ac818cae1bcb3f078fa1807192337154bc35
0.007006406784057617
The User1's one-time-nonce is:  02fde75698f5c00f58f797e263b7875a5d46822d386263c19c80480e1a887bbfcf

0.004002809524536133
The User1's secret is  f0620e1d72a601f8c7e7ad7d6751e8692ceec0a042c3de49ecca3de214b42881
0.007006406784057617
The User2's secret is  f0620e1d72a601f8c7e7ad7d6751e8692ceec0a042c3de49ecca3de214b42881
0.00700688362121582


The Secret is the same for both User1 and User2.

RA calculates and validates the stealth address:  67e87e422a80049c1dccb90edf8c57179212cbd17ce8261e6496e178035dd6f

0.007006645202636719
The Same stealth adddress is publicly visible on the Blockchain.

User2 calculates and commitment cm1:020bb70ecaf3c41c0400607d2f6b99340633f917b369dc11956ba350a8d92c084a
User2 Verify commitment cm: 1
success
Verify commitment time: 0.004003763198852539

USer2 can get transactions using:  f0620e1d72a601f8c7e7ad7d6751e8692ceec0a042c3de49ecca3de214b428810x5d5dace2017ee74f117ab56ca81d60dbcabb7984c341ff08bf915e2c21ddbe12
0.0040035247802734375
0.045041561126708984

Process finished with exit code 0
```

Figure 3. Experimental results graph
图 3. 实验结果图

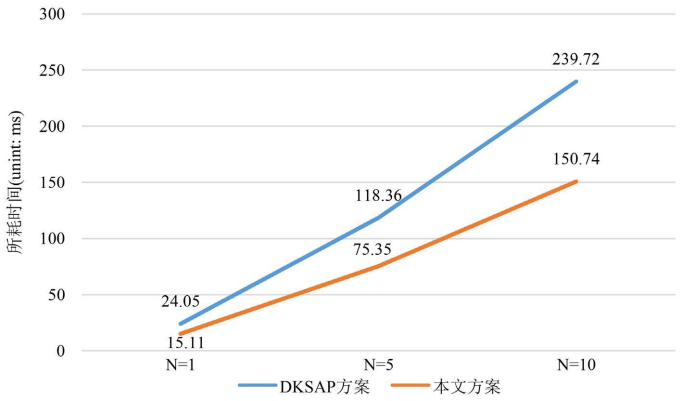

Figure 4. Schematic of the calculation overhead of the transaction initiator
图 4. 交易发起方的计算开销示意图

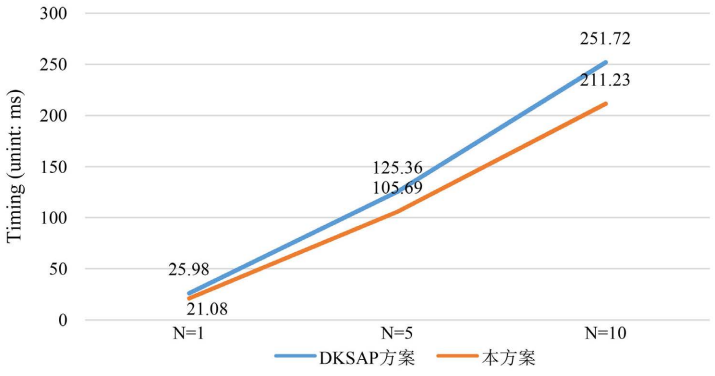

Figure 5. Schematic of the calculation overhead of the transaction recipient
图 5. 交易接受方的计算开销示意图

5. 结论

随着网络技术的快速发展，尽管区块链为匿名交易带来了诸多便利，但交易参与者的不诚实行为和来自外部攻击者的恶意攻击仍对用户隐私构成挑战。文中基于 DKSAP 提出了一种可在区块链上监督的完全匿名交易方案，并设计了一种“挑战－响应”机制来跟踪和识别交易参与者的恶意行为。最后，实现了一种既能实现完全匿名交易又适用于恶意参与者模型的交易方案。实验结果表明，与原方案相比，所提方案对交易参与者的计算开销相对较小。然而，本文提出的方案引入了监管机构的概念，以实现“可控监管”，即交易参与者需要对监管机构完全信任。因此，在未来的工作中，将会研究零知识证明、同态加密等方向来构造不需要监管机构的匿名交易方案。

基金项目

国家自然科学基金项目(61962009、62202118)，国家重点研发计划项目“航空装备制造业集聚区协同制造集成技术研究与应用示范”(2020YFB1713300)、2020.10~2023.09，贵州省教育厅尖端技术人才工程项目([2022] 073)。

参考文献

[1] 袁勇, 王飞跃. 区块链技术发展现状与展望[J]. 自动化学报, 2016, 42(4): 481-494. https://doi.org/10.16383/j.aas.2016.c160158

[2] Nakamoto, S. (2008) Bitcoin: A Peer-to-Peer Electronic Cash System. Decentralized Business Review, 21260. https://bitcoin.fr/public/bitcoin.pdf

[3] Li, T., Wang, Z., Yang, G., *et al*. (2021) Semi-Selfish Mining Based on Hidden Markov Decision Process. *International Journal of Intelligent Systems*, **36**, 3596-3612. https://doi.org/10.1002/int.22428

[4] Li, T., Chen, Y., Wang, Y., *et al*. (2020) Rational Protocols and Attacks in Blockchain System. *Security and Communication Networks*, **2020**, 1-11. https://doi.org/10.1155/2020/8839047

[5] Chen, Y., Sun, J., Yang, Y., *et al*. (2022) PSSPR: A Source Location Privacy Protection Scheme Based on Sector Phantom Routing in WSNs. *International Journal of Intelligent Systems*, **37**, 1204-1221. https://doi.org/10.1002/int.22666

[6] 邵奇峰, 金澈清, 张召, 钱卫宁, 周傲英. 区块链技术: 架构及进展[J]. 计算机学报, 2018, 41(5): 969-988.

[7] 何蒲, 于戈, 张岩峰, 鲍玉斌. 区块链技术与应用前瞻综述[J]. 计算机科学, 2017, 44(4): 1-7+15.

[8] 沈鑫, 裴庆祺, 刘雪峰. 区块链技术综述[J]. 网络与信息安全学报, 2016, 2(11): 11-20.

[9] Huang, C., Zhao, Y., Chen, H., *et al*. (2021) ZkRep: A Privacy-Preserving Scheme for Reputation-Based Blockchain System. *IEEE Internet of Things Journal*, **9**, 4330-4342. https://doi.org/10.1109/JIOT.2021.3105273

[10] dos Santos Abreu, A.W., Coutinho, E.F. and Bezerra, C.I.M. (2021) Performance Evaluation of Data Transactions in Blockchain. *IEEE Latin America Transactions*, **20**, 409-416. https://doi.org/10.1109/TLA.2022.9667139

[11] Goldwasser, S., Micali, S. and Rackoff, C. (2019) The Knowledge Complexity of Interactive Proof-Systems. In: Goldreich, O., Ed., *Providing Sound Foundations for Cryptography*: *On the Work of Shafi Goldwasser and Silvio Micali*, Association for Computing Machinery, New York, 203-225. https://doi.org/10.1145/3335741.3335750

[12] Li, X., Mei, Y., Gong, J., Xiang, F. and Sun, Z. (2020) A Blockchain Privacy Protection Scheme Based on Ring Signature. *IEEE Access*, **8**, 76765-76772. https://doi.org/10.1109/ACCESS.2020.2987831

[13] Liu, Z., Yang, G., Wong, D.S., *et al*. (2021) Secure Deterministic Wallet and Stealth Address: Key-Insulated and Privacy-Preserving Signature Scheme with Publicly Derived Public Key. *IEEE Transactions on Dependable and Secure Computing*, **19**, 2934-2951. https://doi.org/10.1109/TDSC.2021.3078463

[14] 王化群, 吴涛. 区块链中的密码学技术[J]. 南京邮电大学学报(自然科学版), 2017, 37(6): 61-67. https://doi.org/10.14132/j.cnki.1673-5439.2017.06.010

[15] Sasson, E.B., Chiesa, A., Garman, C., *et al*. (2014) Zerocash: Decentralized Anonymous Payments from Bitcoin. 2014 *IEEE Symposium on Security and Privacy*, Berkeley, 18-21 May 2014, 459-474.

[16] Noether, S. (2015) Ring Signature Confidential Transactions for Monero. Cryptology ePrint Archive, Paper 2015/1098. https://eprint.iacr.org/2015/1098

[17] 张思亮, 凌捷, 陈家辉. 可追踪的区块链账本隐私保护方案[J]. 计算机工程与应用, 2020, 56(23): 31-37.

[18] 罗聪. 基于零知识证明的 UTXO 模型区块链隐私保护方法研究[D]: [硕士学位论文]. 北京: 北京交通大学, 2021. https://doi.org/10.26944/d.cnki.gbfju.2021.002721

[19] Feng, C., Tan, L., Xiao, H., *et al*. (2020) PDKSAP: Perfected Double-Key Stealth Address Protocol without Temporary Key Leakage in Blockchain. 2020 *IEEE*/*CIC International Conference on Communications in China* (*ICCC Workshops*), Chongqing, 9-11 August 2020, 151-155. https://doi.org/10.1109/ICCCWorkshops49972.2020.9209929

[20] Feng, C., Tan, L., Xiao, H., *et al*. (2021) EDKSAP: Efficient Double-Key Stealth Address Protocol in Blockchain. 2021 *IEEE* 20*th International Conference on Trust*, *Security and Privacy in Computing and Communications* (*TrustCom*), Shenyang, 20-22 October 2021, 1196-1201. https://doi.org/10.1109/TrustCom53373.2021.00162

[21] Fan, X. (2018) Faster Dual-Key Stealth Address for Blockchain-Based Internet of Things Systems. In: Chen, S., Wang, H. and Zhang, L.-J., Eds., *Blockchain—ICBC* 2018. *ICBC* 2018. *Lecture Notes in Computer Science*, Vol. 10974, Springer, Cham, 127-138. https://doi.org/10.1007/978-3-319-94478-4_9

[22] Koblitz, N. (1987) Elliptic Curve Cryptosystems. *Mathematics of Computation*, **48**, 203-209. https://doi.org/10.1090/S0025-5718-1987-0866109-5

[23] Pedersen, T.P. (1992) Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In: Feigenbaum, J., Ed., *Advances in Cryptology—CRYPTO* '91. *CRYPTO* 1991. *Lecture Notes in Computer Science*, Vol. 576, Springer, Berlin, 129-140. https://doi.org/10.1007/3-540-46766-1_9

[24] 张小艳, 李秦伟, 付福杰. 基于数字承诺的区块链交易金额保密验证方法[J]. 计算机科学, 2021, 48(9): 324-329.