ShuffleParty:一种新的区块链隐私保护机制

胡锋鸣[1,2],赵红武[1,2],金 瑜[1,2]
(1. 武汉科技大学 计算机科学与技术学院,湖北 武汉 430065;
2. 湖北省智能信息处理与实时工业系统重点实验室,湖北 武汉 430065)

摘 要:CoinParty 是目前区块链中最为流行的隐私保护技术之一,由于其引入了第三方节点进行混币,存在安全风险;又由于其通过节点之间的传递校验份额进行校验和检验最终地址,导致效率低下。基于此,为了解决 CoinParty 存在的安全性和效率问题,提出了 ShuffleParty,一种无第三方节点,完全分散的混币机制。首先,将所有的混币节点生成一个由每个参与者控制的托管地址,将需要混币的金额发送到托管地址中;然后,随机选出多个洗牌节点,将自己的混币交易输出地址用洗牌节点的公钥进行层层加密,交给洗牌节点进行层层解密和洗牌;最后,一个洗牌节点获得了完整的明文地址列表后,将地址列表广播给普通混合者进行验证,并生成共同最终置换。通过安全分析和实验表明,ShuffleParty 安全性优于 CoinParty,并且混币效率也高于 CoinParty。

关键词:区块链;隐私保护;混币机制;洗牌节点;多层加密
中图分类号:TP399 **文献标识码**:A **文章编号**:1673-629X(2023)03-0078-07
doi:10.3969/j.issn.1673-629X.2023.03.012

ShuffleParty:A New Privacy Protection Mechanism in Blockchain

HU Feng-ming[1,2],ZHAO Hong-wu[1,2],JIN Yu[1,2]
(1. Dept. of Computer Science & Technology,Wuhan University of Science & Technology,Wuhan 430065,China;
2. Hubei Province Key Laboratory of Intelligent Information Processing & Real-time Industrial,Wuhan 430065,China)

Abstract:CoinParty is one of the most popular privacy protection technologies in the block chain at present. Because it introduces third-party nodes to mix currencies,there are security risks. It is also inefficient because it uses checksum to check the final address by passing the share of checks between nodes. Based on this,in order to solve the security and efficiency problems of CoinParty,we propose ShuffleParty,a fully dispersed currency mixing mechanism without third-party nodes. All the mixed currency nodes are generated into a managed address controlled by each participant. The amount of the mixed currency needed is sent to the managed address. Then several shuffle nodes are randomly selected,the output address of the mixed currency transaction is encrypted layer by layer with the public key of the shuffle nodes,and give it to the shuffle nodes for decryption layer by layer and mixing. Finally,the address list is given to the ordinary mixers for verification,and the common final replacement is generated. The security analysis and experiments show that ShuffleParty is safer than CoinParty,and the efficiency of coin mixing is also higher than CoinParty.

Key words:blockchain;privacy protection;mixing mechanism;shuffle node;multi-layer encryption

0 引 言

2008 年 11 月 1 日,中本聪(Satoshi Nakamoto)发布了比特币白皮书,提出了比特币概念[1]。而后 2009 年 1 月,中本聪创建了第一个区块,被称为“上帝区块”,几天后第二个区块诞生,与创世区块连接形成了链,标志着区块链诞生。区块链作为一种具有去中心化、不可篡改、可追溯、可信特性的分布式数据库,受到了广泛关注[2]。经过了十几年时间,区块链技术不断发展革新,应用广泛,其所出现的隐私问题得到了许多研究者关注[3-5]。区块链的可溯源、可验证等特性是基于系统中所有交易数据公开产生的,因此这导致恶意攻击者可以通过分析数据窥探用户隐私。

近年来,由于研究者对区块链系统中隐私问题的关注,区块链隐私保护技术不断出现。其中,混币机制在保证原有交易结果不变的前提下对交易过程进行混淆,从而抵抗攻击者对账本进行分析。其中,去中心化

收稿日期:2022-05-10 **修回日期**:2022-09-14
基金项目:国家自然科学基金资助项目(61802286)
作者简介:胡锋鸣(1998-),男,硕士,研究方向为区块链隐私保护;通信作者:金 瑜(1973-),女,博士,副教授,CCF 会员(14140M),研究方向为云计算、对等计算和信任模型。

混币机制是目前最为流行的隐私保护技术之一。在该方法中,多方混币机制 CoinParty 通过构建门限托管账户,将参与者输入资产作为抵押,提高了攻击者拒绝服务攻击成本。另一方面,CoinParty 洗牌阶段要求最后一个洗牌节点以字典序排序,避免了最后一个用户操纵结果。但 CoinParty 存在一些不足:(1)引入了第三方节点来进行洗牌,降低了安全性;(2)由第三方节点用地址的校验和验证输出地址列表,导致最后一个洗牌节点可以用校验和相同的地址列表替换原有地址列表,从而盗取资产;(3)效率低,每一个节点都需要进行多层加密解密,洗牌结束后需要进行广播校验和验证,时间复杂度高。

基于此,该文提出了一种新的隐私保护机制(ShuffleParty)。ShuffleParty 在 CoinParty 的基础上进行改进,通过在混币参与者内部选出洗牌节点,由洗牌节点进行洗牌后,将输出地址结果交给混币参与者进行验证,并生成最终置换。选出洗牌节点进行洗牌,避免了第三方节点进行校验和验证带来的安全性问题,增强了对隐私的保护,同时也减少了通信和洗牌所需时间,提高了混币效率。

1　相关工作

1.1　区块链隐私威胁

区块链技术为了保证区块链上记录数据的可溯源、可验证等特性,在分散节点之间维持数据同步并对交易达成共识,区块链上所有交易信息都存储在公开的全局账本中,攻击者很容易获取所有交易信息。攻击者可以通过分析区块链账本中记录的交易数据,发掘其中规律,将用户的不同地址、交易数据关联,并进一步对应到用户的现实身份,严重威胁到用户隐私。Reid 和 Harrigan[6] 通过对公布的账户进行数据分析,统计出了对应地址的资金余额、资金来源、资金流向等信息,并对区块链地址及 IP 地址对应关系进行分析,揭露了比特币用户与实际物理位置的对应关系。Androulaki 等人[7] 提出了挖掘找零地址方法,如果一个交易拥有两个输出,其中一个为已出现过的地址,另一个为新地址,则将新地址视为找零地址。Ron 和 Shamir[8] 获取比特币的交易数据,通过研究用户行为、比特币输入输出情况,得到了一些交易规律,结果表明用户无法通过资产转移有效保护个人隐私。

1.2　混币机制

为了解决区块链隐私问题,研究者提出了一种交换资产、混淆地址的防御机制,即混币机制。混币机制有着多种不同实现方式,根据混币操作者不同,分为中心化混币和去中心化混币两种技术。

中心化混币技术需要第三方节点中心化混币服务商参与,帮助希望进行混币交易的用户与其他用户匹配,构造混币交易,并从中收取一定额度的手续费。各个混币用户与混币服务商进行交易,混币服务商收到用户的资产后随机混淆,然后返回给用户指定的输出地址。混币服务商作为混币交易的中间者,让攻击者难以发现用户的资金流向,通过分析仅仅只能将混币参与者的地址聚类,无法分析出输入地址和输出地址之间对应关系。但混币服务提供商作为混币交易第三方,在交易中会获取所有用户资金以及交易信息,存在安全隐患。最早的中心化服务例如 BitLaundry 采用最基础的中心化混币协议,平台固定配置手续费等参数,导致攻击者可以分析混币交易的平台地址和固定手续费特征将其关联在一起,并且也存在混币服务商内部做恶风险[9]。Bonneau 等人[10] 提出了 Mixcoin 协议,通过基于电子签名承诺机制增强资产安全性,在混币过程中混币服务商需要进行签名。若混币服务商出现违规行为,用户可以公布服务提供商的签名举报混币服务商违规,该混币服务商将会失去信誉。但该方案保护用户资产安全,无法保证用户的信息不被服务提供商泄露。Valenta 等人[11] 提出了 Blindcoin 协议,在沿用 Mixcoin 协议的签名机制的基础上,用户采用盲签名技术进行签名交易,保证第三方混币服务商无法获取到用户输入输出地址关联关系,解决了混币服务商泄露混币用户信息的问题。

中心化混币技术因为中心化混币服务商的参与,方便用户匹配其他混币用户,便于用户使用。但与此同时带来了安全性问题,中心化混币服务提供商作为第三方节点,存在安全隐患,会带来一些潜在风险,例如遭受黑客攻击进而盗窃用户资产。为提高安全性,研究者提出了去中心化混币技术。

去中心化混币技术不需要第三方节点参与,通过多方参与协议代替中心化混币服务提供商,用户在网络中自行寻找其他需要混币用户,通过多方参与者运行协议方式构造混币交易。从根本上解决了中心化混币存在的安全问题,也替用户节省了混币服务提供商的手续费。Gmaxwall[12] 在 2013 年首先提出了第一个去中心化混币方案 CoinJoin,该协议通过将多个相同金额的输入输出放入同一交易中构造混币交易,将多笔相同金额的单出入-单输出交易合并成一个输入输出都相同的多输入-多输出交易,使外部攻击者无法通过分析该交易分辨不同的输出,得到每个输入和输出地址之间的关联关系,确保了外部隐私性。但由于没有第三方节点,混币参与者需要自行进行协商,在协商阶段,参与用户的输入输出地址关联信息会被其他混币用户获取,内部隐私性不能得到保障[13]。

为了解决 CoinJoin 方案的内部隐私性问题,

Ruffing 等人[14]提出了 CoinShuffle 协议。该方案延续了 CoinJoin 核心思想,多个混币参与者构造一个金额相同的多输入-输出交易,保证外部隐私性。在 CoinJoin 基础上,CoinShuffle 增加了输出地址洗牌机制,让参与者按照一定顺序进行排列,每个参与者按照顺序使用后面参与者的公钥进行加密,从第一个参与者开始将加密信息发给后面的参与者,后面的参与者接受用自己的私钥进行解密,并加入自己加密好的输出地址进行混淆,再发给下一个参与者,直到最后一个参与者,得到最终的输出地址明文。将输出地址列表交给参与者验证签名后,创建混币交易。CoinShuffle 通过多层加密,使得中间的参与者无法看到明文,而最后一个参与者无法知道输入输出地址之间的关联,解决了内部隐私性问题。但也存在一定缺陷:最后一个参与者可以决定输出地址列表顺序而非随机,攻击者可以控制顺序;多层加密解密、洗牌导致了该方案的效率偏低,若攻击者进行拒绝服务攻击,整个洗牌过程又需要重新进行[15]。

Ziegeldorf 等人[16]提出了去中心化混币协议 CoinParty,由多个第三方混合节点共同构建托管账户,混币参与者将混币资产放入托管账户进行抵押,从而提高拒绝服务攻击的代价,一定程度上抵御拒绝服务攻击。多个混合节点对输出地址进行洗牌,采用校验和对比所有输出地址哈希值的和对洗牌结果进行校验。并且要求最后一个洗牌节点按字典序排序生成地址列表,再以校验和作为伪随机数生成器的依据生成最终随机置换,在地址列表上加以最终置换得到最终混淆结果,避免了输出地址列表被最后一个洗牌节点操纵。该方案成功解决了 CoinShuffle 中无法预防拒绝服务攻击、最后一个洗牌节点操纵洗牌结果的问题,获得了更好的安全性。但同时也引入了新的问题:CoinParty 采取了多个第三方节点来进行洗牌以及创建交易,第三方节点的加入带来了安全性的降低;使用校验和对比所有输出地址哈希值的和对洗牌结果进行校验,导致最后一个洗牌节点可以用相同哈希值和的输出地址替换原有的输出地址,从而盗取资产,同时也带来了额外的效率问题。TTShuffle[17]在 CoinShuffle 的基础上进行改进,将混币参与者进行分组,由组内洗牌和组间洗牌两阶段完成洗牌过程,分散了洗牌操作,提高了混币效率。但 TTShuffle 是在 CoinShuffle 基础上改进,和 CoinShuffle 一样无法防止拒绝服务攻击。

2 ShuffleParty

2.1 机制概述

ShuffleParty 机制是在 CoinParty 的基础上进行改进,设计了基于洗牌节点选取的洗牌机制。该方案主要部分可以分成五个阶段,即协商阶段、承诺阶段、洗牌阶段、交易阶段,以及混币失败时进入问责阶段(2.2 节详述)。ShuffleParty 机制流程见图 1。

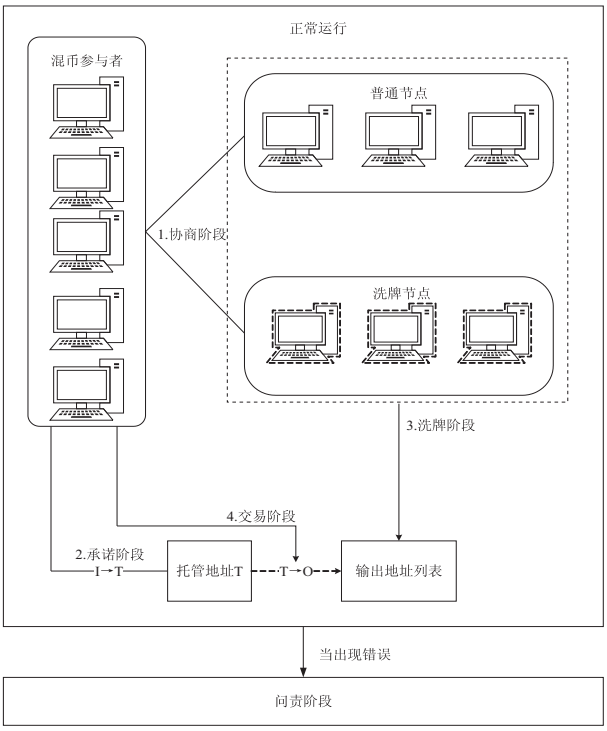

图 1 ShuffleParty 机制流程

为解决 CoinParty 方案所存在的问题,本方案在保留 CoinParty 方案托管地址的机制的同时,去除了第三方节点,由参与者内部选出洗牌节点进行洗牌,最后将洗牌结果交给其他参与者进行验证,避免最后一个洗牌节点通过替换相同哈希值和的输出地址盗取资金。

在混币参与者开始混币方案后,由参与者协商,随机选出 M 个洗牌节点,并协商洗牌节点的洗牌顺序,参与者计算出一组托管地址,将自己的资金转入托管地址。然后,每个洗牌节点广播其生成的密钥(ek,dk)中的公钥 ek,每个普通参与者按照确认好的洗牌顺序,使用洗牌节点的公钥 ek 对自己的输出地址进行层层加密,而洗牌节点使用后面洗牌节点的公钥 ek 对输出地址进行层层加密。完成加密后,普通参与者将加密好的输出地址发送给第一个洗牌节点,第一个洗牌节点将收到的信息用自己的私钥 dk 进行一层解密,并加入自己加密好的输出地址形成输出地址列表,随机洗牌后发送给下一个洗牌节点。层层解密后,最后一个洗牌节点加入自己的输出地址,对输出地址进行字典序排序后,广播最后的输出地址列表。参与者检查自己的输出地址是否在列表中,地址列表是否是字典序排序。若输出地址丢失或列表不是字典序排序,则洗牌无效,进入责备阶段找出违规者,托管地址中的资金退回到参与者输入地址。反之,参与者以所有输出地址哈希值的和作为伪随机数生成器的依据生成公

共随机置换,得到最终混淆结果,按照最终结果生成交易。

2.2　详细描述

(1)协商阶段:N 个混币用户在开始混币机制后,形成初始节点集合 U,公布自己的输入地址 I_n($n \in \{1,2,\cdots,N\}$),共同协商混币金额 v,洗牌节点数量 M($M < N$)。为了保证混币的安全性,要避免洗牌节点全部是恶意节点,因此洗牌节点的选取采取随机方式,减少恶意节点合谋侵犯其他节点隐私。如算法 1 所示,从 N 个混币参与者中随机选出 M 个洗牌节点,形成洗牌节点集合 sList,洗牌节点按随机顺序排列。所有节点分为了普通节点集合 $P = \{P_1, P_2, \cdots, P_{N-M}\}$ 和洗牌节点集合 $S = \{S_1, S_2, \cdots, S_M\}$。每个洗牌节点 $j \in \{1,2,\cdots,M\}$ 生成一个短暂的加密解密密钥对 $(\mathrm{ek}_j, \mathrm{dk}_j)$,将加密密钥公钥$\mathrm{ek}_j$ 广播给所有参与者。

算法 1:Shuffle nodes selection
输入:Collection of initial nodes U
　　Number of Shuffle nodes M
输出:Collection of shuffle nodes S
(1) $S \leftarrow$ null
(2) for i in range (1,M)
(3) node $\leftarrow$ random(U)
(4) S. add(node)
(5) return S

(2)承诺阶段:混币参与者共同生成一个托管账户 T,为了防止托管资金被恶意的混币参与者窃取,T 由一个新的 ECDSA 密钥对生成,T 在混币参与者的联合控制下。参与者 n 将混币金额 v 转入对应的托管账户 T,$\{I_1, I_2, \cdots, I_M\} \rightarrow T$,参与者可以互相验证是否将指定金额 v 转入了托管账户 T,确保每个参与者都完成承诺后进入下一阶段。托管地址 T 创建过程如下:

(a)随机生成一个值 k 作为托管地址 T 的私钥,通过伪随机秘密分享(Pseudo-Random Secret Sharing, PRSS)[18] 使参与者 i 获得私钥 d 的份额 $[k]_n$。

(b)每个参与者计算他的公钥份额 $[Q]_n = [d]_n \cdot G$,G 为椭圆曲线的基点。

(c)每个参与者 i 将他的公钥份额 $[Q]_n$ 广播给其他参与者。

(d)每个参与者收到其他人的公钥份额后,重构公钥 $Q = \sum_{n=1}^{N} \lambda_n [Q]_n = \sum_{n=1}^{N} \lambda_n [d]_n \cdot G = dG$,$\lambda_n$ 为 $x = 0$ 处的拉格朗日基础多项式。

(e)使用公钥 Q 创建托管地址 T。

(3)洗牌阶段:如图 2 所示,参与者将混币金额转入托管账户完成承诺后,每个普通参与者 P_i 将输出地址 O_i 用洗牌节点的加密密钥 ek_j 按顺序层层加密,用 m_a^b 表示节点 a 的输出地址加密了 b 层形成的加密信息,得到 $m_i^M = \mathrm{ek}_1(\mathrm{ek}_2(\cdots \mathrm{ek}_M(O_i)))$(加密结构见图 3)。每个洗牌节点 j 将输出地址 O_j 用排在自己后面的洗牌节点加密密钥按顺序层层加密,得到 $m_j^{M-j} = \mathrm{ek}_{j+1}(\mathrm{ek}_{j+2}(\cdots \mathrm{ek}_M(O_j)))$。该过程如算法 2 所示:

算法 2:Multilayered encryption
输入:Collection of common nodes S
　　Collection of shuffle nodes S
输出:Multilayered encrypted message m_i^M, m_j^{M-j}
(1) for P_i in P
(2) $m_i \leftarrow O_i$
(3) for j in range(1,M)
(4) $m_i \leftarrow$ AES. encrypt(m_i, dk_j)
(5) $m_i^M \leftarrow m_i$
(6) for S_j in S
(7) $m_j \leftarrow O_j$
(8) for n in range(j + 1,M)
(9) $m_j \leftarrow$ AES. encrypt(m_j, dk_n)
(10) $m_j^{M-j} \leftarrow m_j$
(11) return m_i^M, m_j^{M-j}

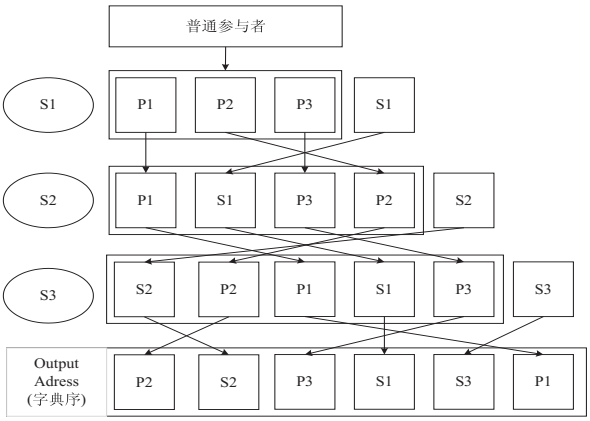

图 2　洗牌过程

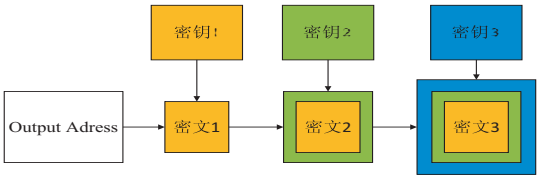

图 3　加密结构

每个普通参与者 P_i 各自将 m_i^M 发送给第一个洗牌节点 S_1,S_1 收到普通参与者的 m_i^M 后,使用自己的解密密钥 dk_1 对每一条消息 m_i 进行解密,解除掉第一层加密,得到 m_i^{M-1},将自己的 m_i^{M-1} 放入一起进行随机打乱,形成 mlist_1(Message List,数据结构见图 4),然后将 mlist_1 发送给下一个洗牌节点 S_2。洗牌节点 S_2 收到 mlist_1 后,将其中的每一个加密消息提取出来,使用自己的解密密钥 dk_2 进行解密,再放入自己的加密信息 m_2^{M-2},随机洗牌后形成 mlist_2,发送给下一个洗牌节点 S_3。依此类推,最终最后一名洗牌节点 S_M 获取到

mlist_{M-1}，取出每一条加密消息使用自己的解密密钥 dk_M 进行解密，得到其他参与者的输出地址列表，然后将自己的输出地址 O_M 插入，进行字典符排序得到完整的输出地址列表 mlist_M，最后将完整的输出地址列表 mlist_M 广播给所有的其他参与者进行验证。洗牌过程如算法 3 所示：

算法 3：Shuffle
输入：Collection of common nodes S
Collection of shuffle nodes S
Multilayered encrypted message m_i^M, m_j^{M-j}
输出：mlist_M
(1) for P_i in P
(2) send m_i^M in S_1
(3) for S_1
(4) decryptedMsg ← AES. decrypt(m_i^M, dk_1)
(5) mlist_1. add(decryptedMsg)
(6) mlist_1. add(m_1^{M-1})
(7) mlist_1 ← mlist_1. random
(8) send mlist_1 to S_1
(9) for S_j in $S(1 < j < M)$
(10) for m_n in mlist_{j-1}
(11) decryptedMsg ← AES. decrypt(m_n, dk_j)
(12) mlist_j. add(decryptedMsg)
(13) mlist_j. add(m_j^{M-j})
(14) mlist_{j+1} ← mlist_j. random
(15) send mlist_{j+1} to S_{j+1}
(16) for S_M
(17) $\text{mlist}_{M-1} = \{m_1, m_2, \cdots, m_{N-1}\}$
(18) for m_{M-1} in mlist_{M-1}
(19) decryptedMsg ← AES. decrypt(m_{M-1}, dk_M)
(20) mlist_M. add(decryptedMsg)
(21) mlist_M. add(O_i)
(22) mlist_M ← mlist_M. lexorder( )
(23) broadcast(mlist_M)
(24) return mlist_M

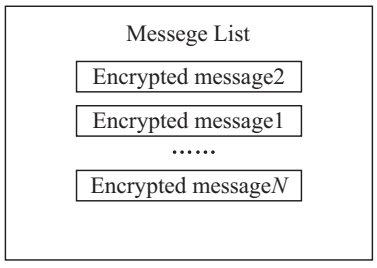

图 4 洗牌阶段 Message List 结构

参与者收到输出地址列表 mlist_M 后，验证自己的输出地址是否在输出地址列表中，地址列表是否按字典序排序。若输出地址不在或地址列表没有按字典序排序，将托管地址 T_n 中的资金 v 转移回输入地址 I_n，并进入责备阶段。反之，如果一切正常，将每个参与者计算输出地址的哈希和，以此为伪随机数生成器的依据生成公共随机置换，在 mlist_M 上增加公共随机置换得到最终输出地址列表 AddressList。该过程如算法 4 所示：

算法 4：final permutation
输入：mlist_M
输出：AdressList
//校验成功后进行最终置换
(1) $\text{mlist}_M = \{m_1, m_2, \cdots, m_N\}$
(2) for m_M in mlist_M
(3) hashsum ← hashsum + hash(m_M)
(4) AdressList ← mlist_M. random. seed(hashsum)

(4)交易阶段：混币参与者创建交易 $T \rightarrow \{O_1, O_2, \cdots, O_M\}$，托管地址 T_i 的私钥 d_i 由所有参与者共同生成，最终由混币参与者合作签署交易。由于参与者没有自己对应托管账户的私钥，所以各参与者广播出自己在承诺阶段获得的私钥份额，重构出托管地址 T 的私钥 k，然后生成有效的签名，将交易广播给区块链。

(a)生成托管地址到最终输出地址列表的交易 $T \rightarrow \{O_1, O_2, \cdots, O_M\}$。

(b)每个参与者验证交易的输出地址列表是否与自己生成的最终输出地址列表 AddressList 相同。若不同，交易取消，进入责备阶段。

(c)每个参与者广播自己的私钥份额 $[k]_n$。

(d)重构托管地址私钥 $k = \sum_{n=1}^{N} [k]_n$。

(e)使用私钥 k 生成对托管地址 T 的有效签名。

(f)将交易提交给区块链。

(5)问责阶段：在整个混币的流程中，每个参与者监督着其他的参与者没有进行违规操作。一旦发现违规现象，诚信的参与者会通告这一现象，混币流程将会停止并执行该阶段，识别出违规操作者并将其剔除，重新进行混币。这里分别讨论每个阶段的违规操作。

(a)承诺阶段：承诺阶段是使参与者将混币金额放入托管账户作为押金机制，是保证后续阶段安全的基础，参与者之间互相通过输入地址，在区块链上验证是否每个参与者都将混币金额 v 转入了托管地址，若超时后，仍有参与者未完成承诺，则将其剔除。

(b)洗牌阶段：该阶段会出现加密密钥错误、最后一个参与者不按字典符排序等错误，若发现违规操作，洗牌节点可以广播他们的解密密钥，以及他们收到的消息，从而允许参与者对每个洗牌节点的操作进行重放，找出其中的违规操作者。发现违规操作者后，将其剔除，然后将押金退回到输入地址。

若其中一个洗牌节点 S_j 宕机，下个洗牌节点 S_{j+1} 在限定时间内没有接受到 S_j 加密消息，S_{j+1} 将超时信息广播给其他参与者，然后剔除掉宕机的参与者，退回押金，剩余参与者重新进行混币。若最后一个洗牌节

点 S_M 宕机, S_M 没有在限定时间内广播输出地址列表,其他参与者将会判定超时,将其剔除,重新进行混币操作。

3　分　析

3.1　安全性分析

ShuffleParty 通过采取随机选择洗牌节点的方式,在混币参与者内部选出洗牌节点,且洗牌节点排列顺序随机,参与者无法自行参与洗牌过程。并沿用了 CoinParty 机制中的托管机制,由多个第三方节点共同生成托管账户改为参与者共同生成托管账户,在提高拒绝服务攻击成本的同时,避免了第三方节点的参与,提高了安全性,解决了 TTShuffle 所存在的无法防止拒绝服务攻击的问题,在安全性方面强于 TTShuffle。

在洗牌阶段,ShuffleParty 机制中加密方案使用 AES-CBC 加密,通过多层加密解密,使每个参与者的输出地址加入到输出地址列表的过程中,输出地址明文不会被其他用户知道,而最后一个洗牌节点解密得到明文后无法得知其他用户与输出地址的对应关系,保障了内部隐私性。与 CoinParty 相同,使用托管账户,要求最后一个洗牌节点以字典序排序,避免攻击者可以控制输出地址列表的顺序。CoinParty 在洗牌开始前将输出地址的哈希值秘密分享给每一个洗牌节点,洗牌结束后洗牌节点广播自己的哈希份额,然后每个洗牌节点重构输出地址的哈希和,与洗牌结果的输出地址列表哈希和进行对比,如果相同,则进行下一步。第三方节点使用校验和进行输出地址列表的验证,带来了一定安全性问题:由于输出地址列表是由最后一个洗牌节点产生的,最后一个洗牌节点可以将正确的输出地址列表替换为哈希和相同的地址列表,从而最后通过验证,将混币用户的资金转入到错误的输出地址,窃取混币用户资金。而 ShuffleParty 没有引入任何第三方节点,验证环节由每一个混币用户进行。最后一个洗牌节点将输出地址列表广播,每一个混币用户验证自己的输出地址是否在输出地址列表中,若不在,则进入责备阶段,反之则继续进行下一步。通过由每个混币用户参与验证,保证了输出地址列表的正确性,避免了 CoinParty 中第三方节点带来的安全性问题。

3.2　实验分析

3.2.1　实验环境

操作系统:64 位 Windows10;编程语言:Java;编程工具:Eclipse,jdk1.8.0_101;JXTA 版本:2.4;JXTA CMS 版本:2.4.1;加密 API:bcprov-jdk14。

实验使用 JXTA 模拟区块链底层的 P2P(Peer-to-Peer)网络,建立多个节点加入同一个 PeerGroup,模拟多人进行混币的过程。洗牌过程中的多层加密采用 AES-CBC 模式。

3.2.2　时间分析

该文在 CoinParty 基础上进行改进,为了对比 ShuffleParty 机制和 CoinParty 机制的性能情况,在同样实验环境下对 ShuffleParty 和 CoinParty 进行多次实验,分别从参与者数量和洗牌节点数量控制变量(ShuffleParty 洗牌节点为参与者选出的洗牌节点,CoinParty 洗牌节点为第三方节点),记录实验结果。并在控制混币参与者数量相同的情况下,将 ShuffleParty 和其他混币方案进行对比

实验 1 固定混币参与者数量,增加洗牌节点数量。分别统计 ShuffleParty 和 CoinParty 在 30 个混币参与者参加混币,有 3,4,5,6,7,8 个洗牌节点进行洗牌操作时两个方案正常运行所需要的时间,两个方案的时间对比如图 5 所示。

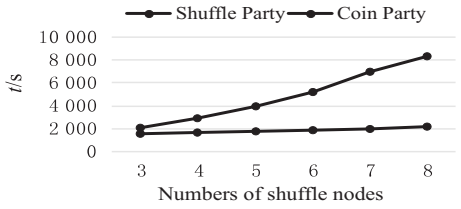

图 5　固定混币参与者数量的运行时间对比

在图 5 中,当混币参与者人数固定时,随着洗牌节点的数量增多,ShuffleParty 所需时间始终小于 CoinParty,两个方案的时间都在递增,但 ShuffleParty 的递增幅度不大。因为 ShuffleParty 和 CoinParty 增加一个洗牌节点,就增加了一层加密解密洗牌的过程,CoinParty 所有节点都需要加密洗牌节点数量的层数,ShuffleParty 只有普通节点需要加密全部层数,而洗牌节点只需要进行后续洗牌节点数量的多层加密。除此之外,CoinParty 通过校验和验证输出地址列表的机制,需要洗牌节点将所持有的校验和份额发送给其他洗牌节点,每增加一个洗牌节点,就增加了更多的时间消耗。

实验 2 固定洗牌节点数量,增加混币参与者数量。分别统计 ShuffleParty 和 CoinParty 在 5 个洗牌节点进行洗牌,有 10,15,20,25,30,35 个混币参与者参加混币时所需要花费的时间,两个方案消耗时间对比如图 6 所示。

在图 6 中,随着混币参与者人数的递增,ShuffleParty 和 CoinParty 两个机制成功运行时间没有很大的波动,近乎平直,TTShuffle 的运行时间随混币参与者人数递增而递增。因为混币参与者的增多并不会改变加密解密洗牌的层数变化,仅仅是在每一层洗牌增加了一个加密消息。而由于 CoinParty 多了一个洗牌节点发送校验和份额的过程,所需花费的时间更

多。TTShuffle 随着混币参与者人数增加 5 人,分组会多出一个,组间洗牌阶段将会多进行一轮传递,导致了时间的增加。

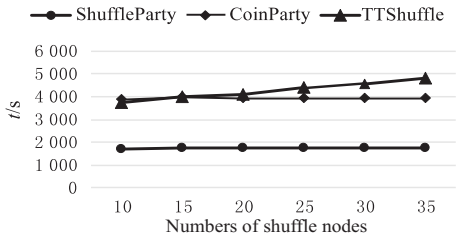

图 6 固定洗牌节点数的运行时间对比

结合图 5 和 6 可以看出,由于 ShuffleParty 直接将输出地址列表发给混币参与者验证,没有 CoinParty 洗牌节点之间互相发送校验和份额的过程,ShuffleParty 机制的时间开销比 CoinParty 小很多,洗牌节点数量越多,前者的混币时间优势越明显。并且 ShuffleParty 和 CoinParty 一样,即使在参与人数众多的情况下,依旧能够保持自身的效率。TTShuffle 尽管在 CoinShuffle 的基础上有了提升,但 ShuffleParty 在效率上还是具有优势。

4 结束语

为了解决 CoinParty 机制中的第三方节点以及校验和验证输出列表导致的安全问题,提出了 ShuffleParty 机制。通过在混币参与者中随机选取洗牌节点,由洗牌节点按随机顺序进行多层洗牌,最后将输出列表发送给参与者验证,避免了第三方节点的参与,保证了混币过程中的隐私安全和资产安全。通过实验表明,提出的 ShuffleParty 机制相对于 CoinParty 机制,在提升安全性的同时,也提高了混币的效率,减少了所需要的时间。

虽然解决了 CoinParty 所存在的一些问题,提高了运行效率,但还存在着一些缺陷。比如,被选出的洗牌节点相较于普通参与者进行了更多的工作量,应设立一定奖励机制对进行洗牌操作的洗牌节点进行奖励。文中洗牌节点由普通参与者随机选出,若有信誉值机制来选取洗牌节点,然后洗牌节点获得奖励,能够更好地提高混币机制的稳定性及安全性。

参考文献:

[1] NAKAMOTO S. Bitcoin:a peer-to-peer electronic cash system[EB/OL]. 2008[2019-11-13]. https://bitcoin.org/bitcoin.pdf.

[2] 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.

[3] KHALILOVM C K, LEVI A. A survey on anonymity and privacy in bitcoin-like digital cash systems[J]. IEEE Communications Surveys & Tutorials,2018,20(3):2543-2585.

[4] MEIKLEJOHN S, ORLANDI C. Privacy-enhancing overlays in bitcoin[C]//International conference on financial cryptography and data security. San Juan: Springer, 2015: 127-141.

[5] 李旭东,牛玉坤,魏凌波,等. 比特币隐私保护综述[J]. 密码学报,2019,6(2):133-149.

[6] REID F, HARRIGAN M. An analysis of anonymity in the bitcoin system[C]//International conference on privacy, security, risk and trust (PASSAT 2011). Boston: IEEE, 2011: 1318-1326.

[7] ANDROULAKI E, KARAME G O, ROESCHLIN M, et al. Evaluating user privacy in bitcoin[C]//International conference on financial cryptography and data security. Okinawa: Springer, 2013: 34-51.

[8] RON D, SHAMIR A. Quantitative analysis of the full bitcoin transaction graph[C]//International conference on financial cryptography and data security. Okinawa: Springer, 2013: 6-24.

[9] MOSER M, BOHME R, BREUKER D. An inquiry into money laundering tools in the bitcoin ecosystem[C]//Ecrime researchers summit. San Francisco: IEEE, 2013: 1-14.

[10] BONNEAU J, NARAYANAN A, MILLER A, et al. Mixcoin: anonymity for Bitcoin with accountable mixes[C]//International conference on financial cryptography and data security. Christ Church: Springer, 2014: 486-504.

[11] VALENTA L, ROWAN B. Blindcoin: blinded, accountable mixes for bitcoin[C]//International conference on financial cryptography and data security. Berlin: Springer, 2015: 112-126.

[12] MAXWELL G. CoinJoin: bitcoin privacy for the real world [EB/OL]. 2013[2019-11-13]. https://bitcointalk.org/index.php? topic=279249.0.

[13] 祝烈煌,高 峰,沈 蒙,等. 区块链隐私保护研究综述[J]. 计算机研究与发展,2017,54(10):2170-2186.

[14] RUFFING T, MORENO-SANCHEZ P, KATE A. Coinshuffle: practical decentralized coin mixing for bitcoin[C]//Proc of the 19th European symposium on research in computer security. Wroclaw: Springer, 2014: 345-364.

[15] 张 奥,白晓颖. 区块链隐私保护研究与实践综述[J]. 软件学报,2020,31(5):1406-1434.

[16] ZIEGELDORF J H, GROSSMANN F, HENZE M, et al. CoinParty: secure multi-party mixing of bitcoins[C]//Fifth ACM conference on data and application security and privacy. San Antonio: ACM, 2015: 75-86.

[17] 程其玲,金 瑜. TTShuffle:一种区块链中基于两层洗牌的隐私保护机制[J]. 计算机应用研究,2021,38(2):363-366.

[18] CRAMER R, DAMGÅRD I, ISHAI Y. Share conversion, pseudorandom secret-sharing and applications to secure computation[C]//Theory of cryptography conference. Cambridge: Springer, 2005: 342-362.