软件学报 ISSN 1000-9825, CODEN RUXUEW
2026,37(4):1777−1800 [doi: 10.13328/j.cnki.jos.007427] [CSTR: 32375.14.jos.007427]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

高效的区块链中可监管身份隐私保护方案*

苏 航[1], 郭兆中[2], 徐茂智[1]

[1](北京大学 数学科学学院, 北京 100871)

[2](北京信息科技大学 计算机学院, 北京 102206)

通信作者: 徐茂智, E-mail: mzxu@math.pku.edu.cn

摘 要: 区块链, 又称分布式账本, 作为新一代信息技术的典型代表, 在金融、医疗、能源和政务等领域得到广泛应用. 区块链中可监管的隐私保护技术既能保护用户隐私, 增强用户对区块链应用的信任, 又能防止区块链被用于非法活动, 确保应用的合规性. 现有可监管区块链隐私保护方案通常基于双线性配对构造, 运算效率较低, 无法满足高并发场景应用需求. 针对上述问题, 提出一种高效的区块链中可监管身份隐私保护方案, 通过设计一种无需配对的接收者身份一致性零知识证明和可追踪环签名方案, 在保护交易双方身份隐私的同时保证监管的有效性. 实验结果表明, 当按照 Monero 参数配置将环签名成员数量设置为 16 时, 高效的区块链中可监管身份隐私保护方案中所有算法执行时间均在 5 ms 以内, 相较于同类型方案效率提升 14 倍以上, 消息长度缩短为原方案的 50%, 具有更高的计算效率和更短的消息长度.

关键词: 区块链身份监管; 环签名; 零知识证明

中图法分类号: TP309

中文引用格式: 苏航, 郭兆中, 徐茂智. 高效的区块链中可监管身份隐私保护方案. 软件学报, 2026, 37(4): 1777–1800. http://www.jos.org.cn/1000-9825/7427.htm

英文引用格式: Su H, Guo ZZ, Xu MZ. Efficient Regulatable Identity Privacy Protection Scheme in Blockchain. Ruan Jian Xue Bao/Journal of Software, 2026, 37(4): 1777–1800 (in Chinese). http://www.jos.org.cn/1000-9825/7427.htm

Efficient Regulatable Identity Privacy Protection Scheme in Blockchain

SU Hang[1], GUO Zhao-Zhong[2], XU Mao-Zhi[1]

[1](School of Mathematical Sciences, Peking University, Beijing 100871, China)

[2](College of Computer Science, Beijing Information Science and Technology University, Beijing 102206, China)

Abstract: Blockchain, also known as a distributed ledger, is a prominent example of next-generation information technology. It has been widely applied in various fields, including finance, healthcare, energy, and government affairs. Privacy protection technologies within the blockchain that can be regulated not only safeguard users' privacy and enhance trust but also prevent misuse of blockchain for illegal activities, ensuring compliance with regulations. Current privacy protection schemes for regulatable blockchains are typically based on bilinear pairing, which exhibit relatively low computational efficiency and fail to meet the demands of high-concurrency scenarios. To address these issues, this study proposes an efficient regulatable identity privacy protection scheme in blockchain. By designing a zero-knowledge proof to verify the consistency of the receiver's identity without bilinear pairing, along with a traceable ring signature scheme, this approach effectively protects the identity privacy of both parties in transactions while maintaining the effectiveness of supervision. The experimental results indicate that when the number of ring members is set to 16, as required by Monero, the execution time of all algorithms in the efficient regulatable identity privacy protection scheme in blockchain is within 5 milliseconds. Compared to similar schemes, efficiency has improved by more than 14 times, and the message length has been reduced to 50% of the original scheme, demonstrating enhanced computational efficiency and a shorter message length.

Key words: blockchain identity regulation; ring signature; zero-knowledge proof

* 基金项目: 国家重点研发计划 (2022YFB2703000); 国家自然科学基金重点项目 (12441105); 国家自然科学基金面上项目 (62072011)
收稿时间: 2024-12-31; 修改时间: 2025-02-11; 采用时间: 2025-03-12; jos 在线出版时间: 2025-07-09
CNKI 网络首发时间: 2025-07-10

1 引 言

区块链是一种源于比特币[1]的去中心化分布式账本, 集成了共识机制、P2P 网络、分布式存储、密码学算法等技术, 具备去中心化、公共可验证、数据不可篡改等技术特征[2]. 近年来, 区块链的应用已经从金融领域扩展到实体经济、社会治理、民生服务等多个重要领域[3]. 区块链数据公开透明的特性提高了区块链应用的可信度, 然而恶意节点也可能通过分析区块链上的公开数据推断或识别出用户的真实身份, 存在用户隐私泄露的风险[4], 严重制约了区块链技术的应用.

为了满足用户身份隐私保护需求, 结合区块链自身特点, 混币机制、零知识证明、一次性地址 (one time address, OTA)、群环签名等技术被应用于区块链隐私保护机制设计中[5−19]. 混币机制是一种地址混淆技术, 它利用比特币交易可以包含多个输入和输出的特点, 将多个交易混淆到一个交易中, 由此增加追踪交易来源和去向的难度. 混币机制包括 Mixcoin[5]等中心化混币以及 CoinJoin[6]、CoinShuffle[7]、CoinParty[8]、Xim[9]等去中心化混币两类技术路线, 已应用于 Dark Wallet[10]、Dash[11]等匿名比特币交易中. 虽然混币技术可通过混淆交易在一定程度上保护用户隐私, 但是混淆后的交易可能保留某些可被攻击者利用来识别用户的特征, 仍存在泄露用户隐私的风险. Zerocoin[12]采用密码累加器技术, 用户通过铸币交易将比特币加入密码累加器转化成零币, 实现与所有 Zerocoin 用户共同混币, 结合零知识证明等技术保证交易相关方的匿名性. 针对 Zerocoin 证明规模大、验证速度慢、仅支持固定面额的币、不支持直接转账等问题, Zerocash[13]使用 zk-SNARK[14]构建去中心化的匿名交易方案, 基于 zk-SNARK 构造的交易可在隐藏交易相关方身份信息的同时被公开验证交易正确性. zk-SNARK 技术的初始化阶段依赖于可信第三方, Aurora[15]引入 zk-STARK[16]技术解决了可信第三方的问题, 然而其产生的证明长度较长, 效率较低. CryptoNote 协议[17]通过一次性地址技术增强对交易参与方的身份隐私保护, 即交易发送者为交易接收者生成一次性地址, 并将新生成的一次性地址设置为交易中的输出账户, 通过一次性地址无法公开推导出交易接收者的身份, 采用可链接环签名[18]混淆发送者地址从而实现对交易发送者身份的隐私保护, 为用户及其交易提供了高度的匿名性. Monero[19]采用 CryptoNote 协议实现对交易参与者身份的隐私保护, 相较于 Zerocash, Aurora 方案也具有更高的效率.

区块链中的身份隐私保护方案使得只有区块链交易相关方才可获得交易的关键信息, 实现了交易身份匿名. 高度的匿名性为洗钱、毒品交易、逃税等非法行为提供了便利, 引发了区块链应用可监管性差的问题. 为了确保区块链技术被合法合规地应用, 在保护区块链用户身份隐私的同时实现对区块链应用的有效监管, 经可信第三方认证账户地址、基于监管者公钥加密交易敏感信息、支持追踪的匿名签名等可监管的区块链身份隐私保护技术得到大量研究[20−34].

引入可信第三方为用户生成经过认证的账户地址是一种可监管的隐私保护方式. Ateniese 等人[20]针对比特币设计了一种身份隐私保护与监管方案, 该方案引入可信权威机构为用户生成经过认证的比特币地址, 用户交易过程中需使用经过认证的地址, 经过认证的地址仍然是匿名的能够保证用户身份的隐私, 可信权威机构可通过经过认证的地址追踪用户身份. 文献 [21,22] 采用了类似的方式, 在此类方案中, 为了实现用户的匿名性, 要求可信权威机构实时在线为用户生成经过认证的地址, 方案存在可信权威单点失效的风险并且效率较低.

采用监管者密钥加密交易过程中的交易相关方身份等敏感信息是一种平衡区块链中身份隐私和监管矛盾的解决方案. 文献 [23,24] 使用监管者公钥加密交易发送者、接收者等信息, 由此实现对交易相关身份的监管. Li 等人[25]提出了采用可验证加密技术实现可监管的交易接收者身份隐私保护思路, 可验证加密允许公开验证密文是否由指定公钥正确加密, 用户生成交易时使用监管者的公钥结合可验证加密技术加密接收者身份等私有交易信息, 共识节点可验证私有交易信息加密的正确性, 当需要监管时, 监管者可通过解密私有交易信息实现对交易接收者身份的监管. Androulaki 等人[26]提出的方案为每个用户分配相应的监管者, 在交易过程中分别使用交易发送者和接收者监管者公钥对交易中的发送者, 接收者等信息进行加密得到监管密文, 并通过零知识证明将监管密文与传输细节联系起来确保监管密文的正确性. Yuen 等人[27]使用公钥加密结合零知识证明的方式进一步保证了对交易双方身份监管的有效性, 该方案通过使用审计员公钥加密发送者和接收者的身份, 使得审计员可获取发送者和

接收者的身份, 并对加密密文的正确性进行零知识证明, 由此保证对发送者和接收者身份监管的正确性. Li 等人[28]将 ElGamal 加密变体方案, 承诺和零知识证明集成到 Monero 系统中, 采用一次性地址实现接收者和发送者身份匿名, 使用监管者公钥结合 ElGamal 加密变体方案加密接收者真实身份, 使得监管者可获取接收者身份, 使用监管者公钥结合 ElGamal 加密变体方案加密发送者一次性公钥在环公钥集合中的位置, 使得监管者可获取发送者身份, 采用零知识证明保证监管者对接收者和发送者身份监管的有效性, 基于上述方式实现了对 Monero 系统中用户身份的监管. 为了实现对交易发送者和交易接收者身份的监管和隐私保护, 采用监管者公钥加密身份信息的解决方案需要针对接收者和发送者身份分别执行相应的加密和零知识证明. Hu 等人[29]针对可编辑区块链提出一种支持用户身份隐私保护与监管的方案 Redact4Trace. Redact4Trace 采用可追踪一次性地址和令牌实现对交易发送方的追踪, 监管者在用户注册过程中存储用户的身份信息, 通过身份信息可测试一次性地址是否为某用户的一次性地址, 如果一次性地址被篡改, 监管者可利用用户注册过程中的身份信息以及用户令牌恢复用户身份的哈希值. Redact4Trace 追踪过程需要追踪者使用所有已注册的用户信息进行测试以获取用户身份, 追踪时间复杂度与用户数量成正比, 方案效率较低. Sarencheh 等人[30]提出了一种可审计监管友好的隐私保护稳定币 PARScoin, 该方案采用 ElGamal 加密、阈值随机化盲签名、零知识证明等密码原语实现对发送者和接收者身份的隐私保护与监管. Li 等人[31]提出一种可审计隐私保护支付通道枢纽方案 AuditPCH, 该方案通过链接随机化谜题 LRP 方案的随机化特性实现对发送者和接收者身份的隐私保护, 审计者通过 LRP 方案中的陷门实现对发送者和接收者身份的揭示. PARScoin 和 AuditPCH 方案都是基于效率相对较低的双线性配对实现的.

支持追踪的匿名签名技术可在实现交易接收者身份隐私保护和监管的同时, 提供交易合法性证明. Li 等人[25]提出采用群签名技术实现交易发送者身份隐私保护和监管方案, 在群签名技术中, 群管理员为群组成员颁发群成员证书, 群成员使用其持有的群成员证书可代表群组生成群签名, 而群管理员具备打开群签名获取签名者的真实身份的能力, 基于群签名技术, 交易中的发送者可将真实身份混淆于群签名群组成员之中实现对发送方的身份隐私保护, 共识节点使用群公钥验证群签名的有效性, 监管者作为群管理员可通过交易中的群签名追踪付款人的真实身份实现对付款人身份的监管. Zhaolu 等人[32]提出一种同时支持区块链隐私保护与监管的方案 DAPCR. DAPCR 采用可追踪环签名方案实现对发送者身份的隐私保护和监管, 但是 DAPCR 仅能直接实现对发送者身份的监管. Yang 等人[33]提出一种支持隐私保护与监管的区块链系统 zkFabLedger, 系统采用环签名方案实现对发送者身份的隐私保护, 采用秘密握手协议实现监管者对发送者身份的监管, 采用表结构账本和 Pedersen 承诺实现对接收者身份的混淆, 采用审计令牌和零知识证明实现对接收者身份的监管. zkFabLedger 基于采用了双线性对的环签名方案构建, 环签名生成 (环成员数量为 20)、验证时间分别高达 180 ms 和 340 ms, 方案效率较低. 宋靖文等人[34]提出一种区块链中可监管的身份隐私保护方案, 基于一次性地址, 线性加密以及零知识证明方案设计了可监管的接收者身份隐私保护方案, 基于环签名方案设计了可监管的发送者身份隐私保护方案, 可实现发送者与接收者身份匿名, 发送者身份的监管与交易合法性验证进行融合, 不需要额外的零知识证明, 相较于已有的方案具有较好的性能. 现有的可监管区块链身份隐私保护方案, 包括文献 [34] 中的方案在内, 大都基于双线性配对设计, 双线性配对的计算复杂度较高并且元素长度较长, 导致其在实际应用中的效率相对较低.

本文设计了一种高效的区块链中可监管身份隐私保护方案. 主要贡献如下.

(1) 设计了一种无需配对的区块链可监管交易接收者身份隐私保护 (regulatable recipient identity privacy protection, RRIPP) 方案. RRIPP 方案采用一次性地址方案[34]在实现接收者身份匿名的同时使得接收者可确认与自身相关的交易, 采用 ElGamal 加密[35]结合监管者公钥加密接收者身份使得监管者可获取接收者身份, 结合一次性地址和 ElGamal 加密计算过程均使用了接收者身份的特点, 基于 Sigma 协议和 Fait-Shamir 转换[36], 设计了一种无需双线性配对的零知识证明, 用以证明一次性地址和 ElGamal 加密计算过程中使用的接收者身份的一致性, 通过交易接收者身份一致性零知识证明保证监管者对接收者身份监管的有效性, 由此实现在保护接收者身份隐私的同时实现有效监管. 一次性地址、ElGamal 加密和交易接收者身份一致性零知识证明均无需使用配对, 采用上述 3 种原语设计的区块链 RRIPP 方案也无需配对.

(2) 设计了一种无需配对的区块链可监管交易发送者身份隐私保护 (regulatable sender identity privacy

protection, RSIPP) 方案. 基于文献 [37] 中无需配对的可链接环签名方案, 针对追踪需求引入追踪者公钥产生链接标签, 使得可链接环签名方案具备追踪功能, 使用 Schnorr 签名[38]简化文献中的方案, 设计了一种无需配对的可追踪环签名方案, 结合本文设计的环签名方案提出 RSIPP 方案, RSIPP 方案通过将监管者设置为环签名中的追踪者, 使得监管者具备追踪环签名真实签名者身份的能力, 除了监管者, 其他用户无法通过环签名获取发送者身份, 由此实现在保护接收者身份隐私的同时实现有效监管. RRIPP 和 RSIPP 方案具有简单的密钥结构, 用户仅需持有一对长期公私钥对即可参与 RRIPP 和 RSIPP 方案, 监管者可使用相同的密钥监管发送者和接收者身份.

(3) 基于 128 bit 安全强度椭圆曲线实现了高效的区块链中可监管身份隐私保护方案. 实验结果表明, 当按照 Monero 参数配置将环签名成员数量设置为 16 时, 交易发送者、交易接收者、监管者和共识节点所需执行算法的运算速度均在 5 ms 以内. 相较于文献 [34] 中方案, 本文中方案的消息长度缩短为原方案的 50%, 各算法效率为原方案的 14 倍以上, 本文的方案具有更高的计算效率和更短的消息长度.

本文方案与已有类似方案的对比情况如表 1 所示. 相较于已有方案, 本文方案同时具备下述优势.

(1) 完备的身份隐私保护与监管: 同时考虑了对交易发送者和交易接收者的隐私保护与监管.

(2) 抗双花攻击: RSIPP 方案的可链接性使其可有效抵抗双花攻击.

(3) 轻量级监管: 监管者仅需对链上数据进行监管, 无需参与交易上链过程.

(4) 较高的方案效率: 在确保安全性的前提下避免使用配对, 具有较低的通信和计算开销 (详见第 6.2 节).

表 1　本文方案与已有方案的对比分析表

方案名称	交易发送者身份隐私保护与监管	交易接收者身份隐私保护与监管	抗双花攻击	监管者不参与交易上链	基于配对
ACA[22]	√	√	×	√	√
DCAP[24]	√	√	√	×	×
Traceable Monero[28]	√	√	√	√	√
Redact4Trace[29]	√	×	×	√	√
PARScoin[30]	仅隐私保护无监管	仅隐私保护无监管	√	√	√
AuditPCH[31]	√	√	√	√	√
DAPCR[32]	√	×	√	√	√
zkFabLedger[33]	√	仅隐私保护无监管	×	√	√
区块链中可监管的身份隐私保护方案[34]	√	√	√	√	√
本文方案	√	√	√	√	×

本文第 2 节介绍本文所需的基础知识, 包括 ElGamal 加密、一次性地址、零知识证明、可追踪环签名及安全模型, 以及方案安全性依赖的困难问题. 第 3 节介绍可监管身份隐私保护方案的基本情况, 包括参与方角色, 身份隐私保护与监管需求和工作流程. 第 4 节介绍本文设计的可监管交易接收者身份隐私保护方案 RRIPP, 包括本文设计的交易接收者身份一致性零知识证明、RRIPP 方案定义及描述、安全模型及证明. 第 5 节介绍本文设计的可监管交易发送者身份隐私保护方案 RSIPP, 包括本文设计的可追踪环签名、RSIPP 方案定义及描述、安全模型及证明. 第 6 节通过对比实验验证本文方案的高效性. 第 7 节总结全文.

2　基础知识

本节首先介绍高效的区块链中可监管身份隐私保护方案采用的已有的 ElGamal 加密方案和一次性地址方案, 随后介绍零知识证明定义、环签名定义及安全模型, 最后介绍本文安全性证明中采用的困难问题.

2.1　ElGamal 加密

ElGamal 加密方案由 $ElG.SetUp$、$ElG.KeyGen$、$ElG.Enc$、$ElG.Dec$ 这 4 个多项式时间算法组成. Tsiounis 等人[35]证明了 ElGamal 加密方案在 DDH (decisional Diffie-Hellman) 假设下的选择明文攻击不可区分 (indistinguishability under chosen plaintext attack, IND-CPA) 安全性. ElGamal 加密方案描述如下.

(1) 初始化算法 $ElG.SetUp(1^{\lambda}) \to pp$

输入安全参数 λ. 设为素数, 选取 q 阶循环群 G, 选取生成元 $g \in G$, 设置公开参数 $pp=(G,q,g)$, 输出公开参数 pp.

(2) 密钥生成算法 $ElG.KeyGen(pp) \to (sk,pk)$

输入公开参数 pp. 选取随机数 $d \in Z_q^*$, 令私钥 $sk=d$, 计算公钥 $pk=g^{sk}$. 输出公私钥对 (sk,pk).

(3) 公钥加密算法 $ElG.Enc(pk,m) \to C$

输入消息接收者的加密公钥 pk 和待加密消息 m. 选取随机数 $r \in Z_q^*$, 计算密文 $C=(C_0,C_1)=(g^r, pk^r m)$. 输出消息密文 C.

(4) 公钥解密算法 $ElG.Dec(sk,C) \to m$

输入消息接收者的私钥 sk 和密文 C. 计算 $m=C_1/C_0^{sk}$. 输出解密结果 m.

2.2 一次性地址

一次性地址 (OTA) 方案[34]由 *OTA.SetUp*、*OTA.Gen*、*OTA.Verif*、*OTA.OtskGen* 这 4 个多项式时间算法组成.

(1) 初始化算法 $OTA.SetUp(1^{\lambda}) \to pp$

输入安全参数 λ. 设 q 为素数, 选取 q 阶循环群 G, 选取生成元 $g \in G$, 选取密码学哈希函数 $H:(0,1)^* \to Z_q^*$, 设置公开参数 $pp=(G,q,g,H)$. 算法输出为 (pp), 其中 pp 为公开参数.

(2) 一次性地址生成算法 $OTA.Gen(pp,pk) \to (P,R)$

输入公开参数 pp, 待生成一次性地址的公钥 pk. 选取随机数 $r \in Z_q^*$, 计算 $R=g^r$, 计算 $r_t=H(pk^r)$, 计算一次性地址 $P=g^{r_t}pk$, 即一次性公钥 $otpk=P$. 输出一次性地址 P 以及生成一次性地址使用的随机数 R.

(3) 一次性地址验证算法 $OTA.Verif(pp,P,R,pk,sk) \to 0/1$

输入 (pp,P,R,pk,sk), 其中 pp 为公开参数, P 为一次性地址, R 为生成一次性地址使用的随机数, pk 为 P 对应的长期公钥, sk 为 P 对应的长期私钥. 计算 $P'=g^{H(R^{sk})}pk$, 若 $P'=P$, 则验证成功并输出 1, 否则验证失败并输出 0.

(4) 一次性私钥计算算法 $OTA.OtskGen(pp,P,R,sk) \to otsk$

输入 (pp,P,R,sk), 其中 pp 为公开参数, P 为一次性地址, R 为生成一次性地址使用的随机数, sk 为 P 对应的长期私钥. 计算一次性私钥 $otsk=H(R^{sk})+sk$. 输出一次性私钥 $otsk$.

2.3 零知识证明

若 $\mathcal{A}$ 是概率多项式时间敌手, 并且下述条件成立, 则 (P,V) 是对于语言 L 及其上定义的证据关系 R 的零知识证明.

(1) 完备性

对于所有 $x \in L$, $w \in R(x)$:$\Pr[\pi \leftarrow ProofGen(pp,x,w) : ProofVerif(pp,x,\pi)=1]=1$.

(2) 可靠性

存在一个可忽略的函数 $negl$, 对于所有 $x \notin L$, $\Pr[\exists \pi, \text{s.t. } ProofVerif(pp,x,\pi)=1] \leqslant negl(x)$.

(3) 零知识

对于所有多项式时间敌手 $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$, 存在一个可忽略的函数 $negl$, 使得 $\left|\Pr[Expt_{\mathcal{A}_1}(k)=1]-\Pr\left[Expt_{\mathcal{A}_2}^{S}(k)=1\right]\right| \leqslant negl(|x|)$. 其中, $Expt_{\mathcal{A}_1}(pp,x,w):\pi \leftarrow ProofGen(pp,x,w)$; $Expt_{\mathcal{A}_2}^{S}(pp,x):\pi \leftarrow S(pp,x)$.

2.4 可追踪环签名及安全模型

可追踪环签名 (traceable ring signature, TRS) 方案由 *TRS.SetUp*、*TRS.KeyGen*、*TRS.Sig*、*TRS.Ver*、*TRS.Link*、*TRS.Trace* 这 6 个多项式时间算法组成[34].

2.4.1 可追踪环签名方案定义

(1) 初始化 $TRS.SetUp(1^{\lambda}) \to pp$

算法输入为安全参数 λ, 输出为公开参数 pp.

(2) 密钥生成算法 $TRS.KeyGen(pp) \to (sk,pk)$

算法输入为公开参数 pp, 输出为公私钥对 (sk,pk).

(3) 环签名算法 $TRS.Sig(pp,S,sk_\pi,pk_a,M)\to\sigma$

算法输入为公开参数 pp、环签名成员公钥集合 S、签名者私钥 sk_π、追踪者公钥 pk_a、签名消息 M, 输出为环签名 σ.

(4) 环签名验证算法 $TRS.Ver(pp,\sigma,pk_a,M)\to(0/1)$

算法输入为公开参数 pp、环签名 σ、追踪者公钥 pk_a、签名消息 M, 输出为验签结果.

(5) 可链接性判断算法 $TRS.Link(pp,\sigma_1,\sigma_2,\mathcal{L})\to(0/1)$

算法输入为公开参数 pp、环签名 σ_1 和 σ_2、历史密钥镜像列表 $\mathcal{L}$, 输出为链接性验证结果.

(6) 签名者身份追踪算法 $TRS.Trace(pp,\sigma,sk_a)\to pk_i$

算法输入为公开参数 pp、环签名 σ、追踪者私钥 sk_a, 输出为真实签名者公钥 pk_i.

2.4.2 可追踪环签名安全模型

一个安全的可追踪环签名方案应该满足不可伪造性、匿名性、可链接性、不可诽谤性和可追踪性[34,37]. 下面基于仿真器 $\mathcal{S}$ 和敌手 $\mathcal{A}$ 之间的实验给出各安全属性的形式化定义.

(1) 查询预言机

O_R: 注册预言机, 敌手 $\mathcal{A}$ 查询此预言机, 注册一个新用户, 该预言机返回一个有效的用户公钥 pk_i, 并将其添加到 LIST_R 中.

O_S: 签名预言机, 敌手 $\mathcal{A}$ 使用环签名成员公钥集合 S, 追踪者公钥 pk_a 和消息 m' 查询此预言机, 该预言机返回一个有效的环签名 σ', 并将签名结果添加到 LIST_S 中.

O_C: 腐化成员预言机, 敌手 $\mathcal{A}$ 使用公钥 pk_i 查询此预言机, 该预言机返回对应的私钥 sk_i, 并将 (pk_i,sk_i,I_i) 添加到 LIST_C 中.

(2) 不可伪造性

不可伪造性是指敌手 $\mathcal{A}$ 不能在仅已知环签名成员公钥的前提下伪造任何环签名. 不可伪造性由敌手 $\mathcal{A}$ 和仿真器 $\mathcal{S}$ 之间的实验 $Expt_{\mathrm{TRS}}^{\mathrm{unfo}}(\lambda)$ 定义.

初始化阶段: $\mathcal{S}$ 运行初始化算法 $pp\leftarrow TRS.SetUp(1^\lambda)$, 将公开参数 pp 发送给 $\mathcal{A}$.

查询阶段: $\mathcal{A}$ 采用自适应策略查询预言机 O_R, O_S 和 O_C.

伪造阶段: $\mathcal{A}$ 将 (m,σ) 提交给 $\mathcal{S}$, 其中 m 为签名消息, σ 为签名.

若满足下述条件, $\mathcal{A}$ 获胜.

1) $1\leftarrow TRS.Ver(pp,\sigma,pk_a,M)$.

2) 对于所有 $pk_i\in S$, pk_i 是 O_R 预言机的输出.

3) 对于所有 $pk_i\in S$, pk_i 未被用于查询 O_C 预言机.

4) σ 不是 O_S 预言机的输出.

令 $Adv_{\mathrm{TRS}}^{\mathrm{unfo}}(\lambda)=\Pr\left[Expt_{\mathrm{TRS}}^{\mathrm{unfo}}(\lambda)=1\right]$ 表示敌手 $\mathcal{A}$ 打破方案不可伪造性的优势.

定义 1. 不可伪造性.

对于概率多项式时间敌手 $\mathcal{A}$, 若敌手打破方案不可伪造性的优势 $Adv_{\mathrm{TRS}}^{\mathrm{unfo}}(\lambda)$ 是可忽略的, 则 TRS 方案满足不可伪造性.

(3) 匿名性

匿名性是指从有效环签名中推断出签名者真实身份是不可行的. 匿名性由敌手 $\mathcal{A}$ 和仿真器 $\mathcal{S}$ 之间的实验 $Expt_{\mathrm{TRS}}^{\mathrm{anon}}(\lambda)$ 定义.

初始化阶段: $\mathcal{S}$ 运行初始化算法 $pp\leftarrow TRS.SetUp(1^\lambda)$, 将公开参数 pp 发送给 $\mathcal{A}$.

查询阶段: $\mathcal{A}$ 采用自适应策略查询预言机 O_R.

挑战阶段: $\mathcal{A}$ 将元组 (S,m) 提交给 $\mathcal{S}$, 该元组含有 n 个公钥的集合 S 以及消息 m. $\mathcal{S}$ 随机选择 $\pi\xleftarrow{R}\{1,\ldots,n\}$ 并计算 $\sigma_\pi\leftarrow TRS.Sig(pp,S,sk_\pi,pk_a,m)$, 其中 sk_π 为 pk_π 对应的私钥, 并将 σ_π 交给 $\mathcal{A}$.

猜测阶段: $\mathcal{A}$ 输出猜测 $\pi' \in \{1,\ldots,n'\}$.

若满足下述条件, $\mathcal{A}$ 获胜.

1) 对于所有 $pk_i \in S$, pk_i 是 O_R 预言机的输出.

2) $\pi' = \pi$.

令 $Adv_{\mathrm{TRS}}^{\mathrm{anon}}(\lambda) = \left|\Pr[Expt_{\mathrm{TRS}}^{\mathrm{anon}}(\lambda)=1] - 1/n\right|$ 表示敌手 $\mathcal{A}$ 打破方案匿名性的优势.

定义 2. 匿名性.

对于概率多项式时间敌手 $\mathcal{A}$, 若敌手打破方案匿名性的优势 $Adv_{\mathrm{TRS}}^{\mathrm{anon}}(\lambda)$ 是可忽略的, 则 TRS 方案满足匿名性.

(4) 可链接性

可链接性是指由同一个私钥生成的环签名是可以被链接的. 可链接性由敌手 $\mathcal{A}$ 和仿真器 $\mathcal{S}$ 之间的实验 $Expt_{\mathrm{TRS}}^{\mathrm{link}}(\lambda)$ 定义.

初始化阶段: $\mathcal{S}$ 运行初始化算法 $pp \leftarrow TRS.SetUp(1^{\lambda})$, 将公开参数 pp 发送给 $\mathcal{A}$.

查询阶段: $\mathcal{A}$ 采用自适应策略查询预言机 O_R, O_S 和 O_C.

伪造阶段: $\mathcal{A}$ 将元组 (m_1,σ_1) 和 (m_2,σ_2) 提交给 $\mathcal{S}$, 每个元组包括消息-签名对 (m_1,σ_1) 和 (m_2,σ_2).

若满足下述条件, 则 $\mathcal{A}$ 获胜.

1) 对于 $i \in \{1,2\}$: $1 \leftarrow TRS.Ver(pp,\sigma_i,pk_a,m_i)$, 并且 σ_i 不是 O_S 预言机的输出.

2) 对于 $pk_i \in S_1 \cup S_2$, pk_i 是 O_R 预言机的输出.

3) O_C 预言机查询次数少于 2 次, 即 $\mathcal{A}$ 最多只能获取 1 个私钥.

4) $0 \leftarrow TRS.Link(S_1,S_2,m_1,m_2,\sigma_1,\sigma_2)$.

令 $Adv_{\mathrm{TRS}}^{\mathrm{link}}(\lambda) = \Pr\left[Expt_{\mathrm{TRS}}^{\mathrm{link}}(\lambda)=1\right]$ 表示敌手 $\mathcal{A}$ 打破方案可链接性的优势.

定义 3. 可链接性.

对于概率多项式时间敌手 $\mathcal{A}$, 若敌手打破方案可链接性的优势 $Adv_{\mathrm{TRS}}^{\mathrm{link}}$ 是可忽略的, 则 TRS 方案满足可链接性.

(5) 不可诽谤性

不可诽谤性是指任何签名者均无法生成与另一个签名者生成的签名相链接的签名. 不可诽谤性由敌手 $\mathcal{A}$ 和仿真器 $\mathcal{S}$ 之间的实验 $Expt_{\mathrm{TRS}}^{\mathrm{exc}}(\lambda)$ 定义.

初始化阶段: $\mathcal{S}$ 运行初始化算法 $pp \leftarrow TRS.SetUp(1^{\lambda})$, 将公开参数 pp 发送给 $\mathcal{A}$.

查询阶段 1: $\mathcal{A}$ 采用自适应策略查询预言机 O_R, O_S 和 O_C.

挑战阶段: $\mathcal{A}$ 将元组 (m,S,pk_π) 提交给 $\mathcal{S}$, 该元组包含消息 m, 含有 n 公钥的集合 S, 签名者公钥 $pk_\pi \in S$, 其中 pk_π 未被用于查询 O_C 和 O_S 预言机. $\mathcal{S}$ 利用与 pk_π 对应的私钥 sk_π 运行签名算法: $\sigma' \leftarrow TRS.Sig(pp,S,sk_\pi,pk_a,m)$, 并将 σ' 交给 $\mathcal{A}$.

查询阶段 2: $\mathcal{A}$ 可任意查询预言机, 但是 pk_π 不能被用于查询 O_C 和 O_S 预言机.

伪造阶段: $\mathcal{A}$ 输出 (m^*,σ^*), 该元组包含消息 m^* 和签名 $\sigma^* \neq \sigma'$.

若满足下述条件, $\mathcal{A}$ 获胜.

1) $1 \leftarrow TRS.Ver(pp,\sigma^*,pk_a,m^*)$ 并且 σ^* 不是 O_S 预言机的输出.

2) 对于所有 $pk_i \in S \cup S^*$, pk_i 是 O_R 预言机的输出.

3) pk_π 未被用于查询 O_C 预言机.

4) $1 \leftarrow TRS.Link(pp,\sigma',\sigma^*,\mathcal{L})$.

令 $Adv_{\mathrm{TRS}}^{\mathrm{exc}}(\lambda) = \Pr[Expt_{\mathrm{TRS}}^{\mathrm{exc}}(\lambda)=1]$ 表示敌手 $\mathcal{A}$ 打破方案不可诽谤性的优势.

定义 4. 不可诽谤性.

对于概率多项式时间敌手 $\mathcal{A}$, 若敌手打破方案不可诽谤性的优势 $Adv_{\mathrm{TRS}}^{\mathrm{exc}}(\lambda)$ 是可忽略的, 则 TRS 方案满足不可诽谤性.

(6) 可追踪性

可追踪性是指监管者可通过有效环签名恢复出签名者的真实身份. 可追踪性由敌手 $\mathcal{A}$ 和仿真器 $\mathcal{S}$ 之间的实验 $Expt^{tra}_{TRS}(\lambda)$ 定义.

初始化阶段: $\mathcal{S}$ 运行初始化算法 $pp \leftarrow TRS.SetUp(1^{\lambda})$, 将公开参数 pp 发送给 $\mathcal{A}$.

查询阶段: $\mathcal{A}$ 采用自适应策略查询预言机 O_R, O_S, O_C.

伪造阶段: $\mathcal{A}$ 使用 sk_{π} 生成签名 σ^*, $\mathcal{A}$ 将 σ^* 发送给 $\mathcal{S}$.

判定阶段: $\mathcal{S}$ 使用监管者私钥 sk_a, 运行 $TRS.Trace(pp,\sigma^*,sk_a)$ 算法, 得到输出 pk_T.

若满足下述条件, $\mathcal{A}$ 获胜.

1) $TRS.Ver(pp,\sigma^*,pk_a,m)=1$, 并且 σ^* 不是 O_S 预言机的输出.

2) 对于所有 $pk_i \in S$, pk_i 是 O_R 预言机的输出.

3) $pk_T \neq pk_{\pi}$, pk_T 未被用于查询 O_C 预言机.

令 $Adv^{tra}_{TRS}(\lambda)=\Pr[Expt^{tra}_{TRS}(\lambda)=1]$ 表示敌手 $\mathcal{A}$ 打破方案可追踪性的优势.

定义 5. 可追踪性.

对于概率多项式时间敌手 $\mathcal{A}$, 若敌手打破方案可追踪性的优势 $Expt^{tra}_{TRS}(\lambda)$ 是可忽略的, 则 TRS 方案满足可追踪性.

2.5 困难问题

(1) 离散对数困难问题

设 G 是阶为大素数 p 的循环群, g 是群 G 的一个生成元, 给定 G 中的非零元素 $y=g^x \bmod p$, 计算 $x \in Z_p$ 是困难的.

(2) DDH 困难问题

设 G 是阶为大素数 p 的循环群, g 是群 G 的一个生成元, 对于任意给定的 Z_p 中的非零元素 (x,y,z), 四元组 (g,g^x,g^y,g^{xy}) 与 (g,g^x,g^y,g^z) 是计算上不可区分的.

3 方案概述

3.1 参与方角色

从区块链隐私保护与监管切入, 结合实际业务场景, 可以将方案参与角色宏观分为用户、监管者和共识节点这 3 类.

(1) 用户

用户为可监管隐私保护区块链系统的使用者, 即实际业务的执行者. 在区块链系统中, 用户扮演交易的发送者或交易的接收者. 作为交易发送者, 其根据业务需求, 构造出相应的区块链交易; 作为交易接收者, 其读取链上发送给自己的交易, 并对该交易进行解析, 获得相应的业务数据, 完成业务的执行.

(2) 监管者

监管者负责对链上业务进行监管, 可掌握相应的交易信息, 但是无法对用户账户进行控制.

(3) 共识节点

共识节点为系统中的用户节点, 负责验证交易的有效性, 打包有效的交易创建新的区块, 然后将该交易广播并最终上链.

3.2 身份隐私保护与监管需求

(1) 区块链交易数据结构抽象

本节对区块链交易数据结构进行抽象, 并基于抽象后的字段分析区块链身份隐私保护与监管需求. 区块链交易数据结构如图 1 所示, 各字段的含义如下.

图 1 交易结构抽象

1) 发送者身份和接收者身份

发送者身份和接收者身份字段即为发送者和接收者的账户地址.

2) 交易内容

交易内容字段是交易所承载的信息, 包括交易金额等交易原生的字段.

3) 交易合法性验证

交易合法性验证是保证交易合法性的字段, 通常为数字签名.

(2) 身份隐私保护与监管需求分析

交易身份隐私保护与监管包括对交易发送者身份和交易接收者身份的隐私保护与监管. 具体而言, 交易发送者应当知悉交易发送者和交易接收者身份信息; 交易接收者知悉交易接收者身份信息; 非交易关联用户 (除监管者外), 对交易发送者和接收者身份均不知悉.

1) 接收者身份隐私保护与监管需求

与接收者身份隐私保护和监管关联的字段为“接收者”字段, 其中, “接收者”字段为接收者的身份和监管者实现对交易身份监管的字段.

为保证交易数据接收过程顺利执行, 交易指定接收者应可基于接收者字段识别与自己相关的交易.

为保证接收者身份隐私, “接收者”字段应保证接收者身份的匿名性.

为保证监管过程的顺利执行, 监管者应可从“接收者”字段获取用于提取交易接收者身份信息, 并且可保证获取的接收者身份的真实性.

2) 发送者身份隐私保护与监管需求

与发送者身份隐私保护和监管关联的字段为“发送者”和“交易合法性验证”字段, 其中, “发送者”字段为发送者的身份以及监管者实现对交易身份监管的字段, “交易合法性验证”字段为发送者对交易数据的签名或证明, 该字段的验证由共识节点完成.

为保证交易数据验证过程顺利执行, 在交易内容上链之前, 共识节点应可公开验证, 或者使用“发送者”字段中的匿名身份协助验证“交易合法性验证”字段, 通过此种方式, 使得共识节点可初步验证交易数据的合法性.

为保证发送者身份隐私, “发送者”字段和“交易合法性验证”字段, 以及共识节点对“交易合法性验证”字段的验证过程, 均应保证发送者身份的匿名性.

为保证监管过程的顺利执行, 监管者应可从“发送者”字段和“交易合法性验证”字段中获取用于提取交易发送者真实身份的信息.

3.3 工作流程

本文设计的一种高效的区块链中可监管身份隐私保护方案主要工作流程包括用户注册、交易生成、交易合法性验证、交易接收以及交易监管这 5 个部分, 如图 2 所示.

(1) 用户注册

用户加入系统时, 为自己生成公私钥对, 将公钥作为真实身份向监管者注册.

(2) 交易生成

针对接收者身份隐私保护需求, 发送者获取接收者账户公钥, 结合一次性地址生成算法, 为接收者生成一次性公钥. 针对接收者身份监管需求, 使用监管者公钥结合公钥加密算法加密接收者身份得到接收者身份监管密文, 为保证监管者得到的接收者身份的真实性, 给出接收者身份监管密文与一次性地址对应的接收者身份一致性零知识证明. 可监管的接收者身份隐私保护方案 RRIPP 见第 4 节.

针对发送者身份隐私保护需求, 发送者使用其上次作为接收者时的一次性公钥作为发送者公钥, 发送者任意

选取多个用户作为环签名成员, 使用环签名成员的一次性公钥和发送者一次性公钥对应的私钥, 结合可追踪环签名为交易生成交易合法性签名. 针对发送者身份监管需求, 将监管者设置为可追踪环签名中的追踪者. 可监管发送者身份隐私保护方案 RSIPP 见第 5 节.

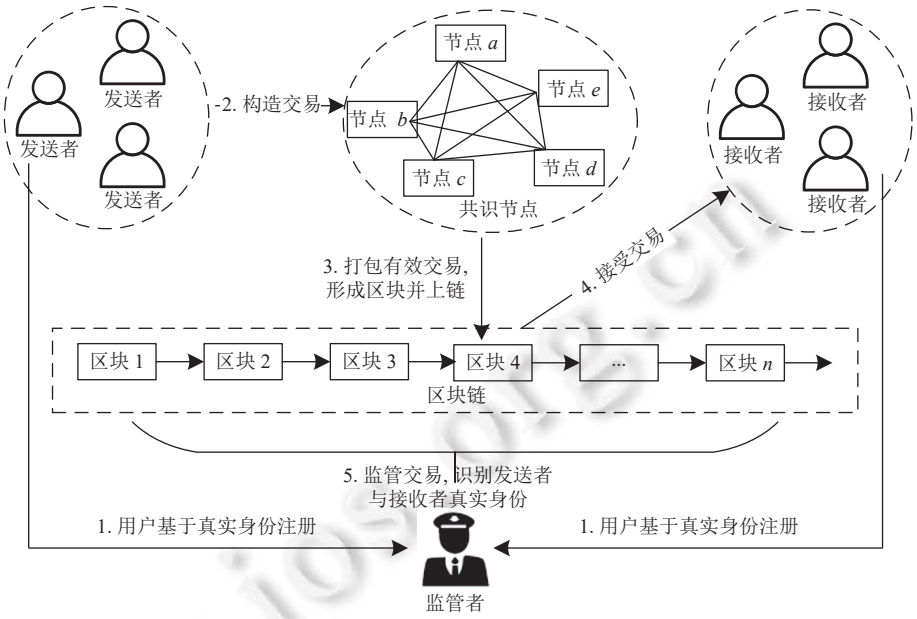

图 2 工作流程

发送者组合接收者一次性公钥, 接收者身份监管密文, 接收者身份一致性零知识证明作为接收者地址字段内容, 环签名成员一次性公钥集合作为发送者地址字段内容, 交易合法性签名作为交易合法性验证字段内容, 交易内容等形成交易.

(3) 交易合法性验证

共识节点打包交易之前, 验证交易中接收者身份一致性零知识证明和交易合法性签名的有效性.

(4) 交易接收

接收者基于一次性地址验证算法验证交易是否与自己相关, 并接收与自己相关的交易.

(5) 交易监管

对于接收者身份的监管, 监管者验证交易中接收者身份一致性零知识证明, 对于验证通过的交易, 解密接收者身份监管密文获取接收者真实身份.

对于发送者身份的监管, 监管者作为可追踪环签名中的追踪者, 获取发送者的一次性公钥, 随后获取以该一次性公钥作为接收者身份的关联交易, 使用该关联交易中的接收者身份监管密文获取发送者真实身份.

4 可监管交易接收者身份隐私保护方案 RRIPP

本节首先提出了 RRIPP 方案所用的交易接收者身份一致性零知识证明, 随后给出 RRIPP 方案的详细设计, 最后给出方案的安全模型及安全性证明.

4.1 交易接收者身份一致性零知识证明

(1) 交互式交易接收者身份一致性证明

交易接收者身份一致性证明用于证明一次性地址生成算法中用于生成一次性地址的公钥 (供交易接收者确认身份) 和 ElGamal 加密算法加密的公钥 (供监管者获取交易接收者身份) 的一致性.

设 ElGamal 加密算法中使用的解密方公私钥对为 (pk_a, sk_a), 加密的公钥明文为 pk_r, 加密时使用的随机数为 r_e, 加密得到的密文为 $C=(C_1=g^{r_e}, C_2=pk_r pk_a^{r_e})$; 一次性地址生成算法中生成的一次性地址为 $P=pk_r g^{r_t}$, 生成一次性地址使用的随机数为 r_t.

交易接收者身份一致性证明协议基于 Sigma 协议给出, 描述如下.

1) 共同输入

协议双方共有消息包括公开参数 pp, 解密方公钥 pk_a, 交易接收者一次性地址 P 和交易接收者身份监管密文 C.

2) 证明者证据

证明者拥有 r_t, r_e.

3) 协议

a) P (证明者): 选取随机数 s_t, $s_e \in Z_q^*$, 计算 $Q_1=g^{s_e}$, $Q_2=g^{s_t} pk_a^{s_e}$, 将 Q_1, Q_2 发送给验证者.

b) V (验证者): 选取随机数 $c \in Z_q^*$, 将 c 发送给证明者.

c) P (证明者): 计算 $w_t=s_t-cr_t$, $w_e=s_e+cr_e$.

d) V (验证者): 计算 $Q_1'=g^{w_e}/C_1^c$, $Q_2'=pk_a^{w_e} g^{w_t} P^c/C_2^c$, 若 $Q_1'=Q_1$ 并且 $Q_2'=Q_2$, 则交易接收者身份一致性证明验证通过, 否则验证不通过.

(2) 非交互交易接收者身份一致性证明描述

基于交互式交易接收者身份一致性证明和 Fait-Shamir 转换方法, 非交互交易接收者身份一致性证明描述如下.

1) 身份一致性证明生成算法 *ZK.ProofGen*

输入公开参数 pp, 公钥 pk_a, 交易接收者公钥 pk_r, 交易接收者一次性地址 P, 交易接收者身份监管密文 C, 生成一次性地址使用的随机数 r_t, 加密时使用的随机数 r_e, 其中 pp 为 ElGamal 加密方案和一次性地址方案中的所有公开参数.

选取随机数 s_t, $s_e \in Z_q^*$.

计算 $Q_1=g^{s_e}$, $Q_2=g^{s_t} pk_a^{s_e}$, $h=H(g \| pk_a \| C_1 \| C_2 \| P \| Q_1 \| Q_2)$, $w_t=s_t-hr_t$, $w_e=s_e+hr_e$.

输出身份一致性证明 $\pi=(w_t, w_e, h)$.

2) 身份一致性证明验证算法 *ZK.ProofVerif*

输入公开参数 pp, 交易监管者公钥 pk_a, 交易接收者一次性地址 P, 交易接收者身份监管密文 C, 交易接收者身份一致性证明 π.

验证方计算 $Q_1'=g^{w_e}/C_1^h$, $Q_2'=pk_a^{w_e} g^{w_t} P^h/C_2^h$, $h'=H(g \| pk_a \| C_1 \| C_2 \| P \| Q_1' \| Q_2')$.

若 $h=h'$, 则交易接收者身份一致性证明验证通过, 否则验证不通过.

(3) 非交互交易接收者身份一致性证明分析

定理 1. 在随机预言机模型下, 非交互交易接收者身份一致性证明是零知识证明协议.

定理的证明由下述 3 个引理给出, 3 个引理分别表明该协议具备 (1) 完备性; (2) 可靠性; (3) 零知识性.

引理 1. 非交互交易接收者身份一致性证明具备完备性.

证明: 对于拥有证据 $w=(r_t, r_e) \in R(x)$ 并且如实执行协议的诚实证明者, 其给出的证明验证过程如下:

$$\begin{cases} Q_1'=g^{w_e}/C_1^h=g^{s_e+hr_e}/g^{r_e h}=g^{s_e}=Q_1 \\ Q_2'=pk_a^{w_e} g^{w_t} P^h/C_2^h=pk_a^{s_e+hr_e} g^{s_t-hr_t} (pk_r g^{r_t})^h/(pk_r pk_a^{r_e})^h=g^{s_t} pk_a^{s_e}=Q_2 \\ h'=H(g \| pk_a \| C_1 \| C_2 \| P \| Q_1' \| Q_2')=H(g \| pk_a \| C_1 \| C_2 \| P \| Q_1 \| Q_2)=h \end{cases}.$$

综上, 非交互交易接收者身份一致性证明满足完备性要求. 证毕.

引理 2. 交易接收者身份一致性证明协议满足可靠性.

证明: 对于 $\forall x=(pk_a, P, C) \notin L$, 可归纳为以下两种情况.

1) 交易发送者产生 ElGamal 密文 C 时使用的 pk_a' 不是监管者的公钥.

在此种情况下, $pk_a \neq pk_a'$, $C_1=g^{r_e}$, $C_2=pk_r pk_a'^{r_e}$. 验证方验证时计算:

$$Q_2' = pk_a^{w_e} g^{w_t} P^h / C_2^h = pk_a^{s_e+hr_e} g^{s_t-hr_t} (pk_r g^{r_t})^h / (pk_r pk_a'^{r_e})^h = g^{s_t} pk_a'^{s_e} (pk_a / pk_a')^{hr_e} = Q_2 (pk_a / pk_a')^{hr_e}.$$

由于 $pk_a \neq pk_a'$, 故 $Q_2' \neq Q_2$, 由于 H 是抗碰撞哈希函数, $h' = H(g \| pk_a \| C_1 \| C_2 \| P \| Q_1 \| Q_2') = h = H(g \| pk_a \| C_1 \| C_2 \| P \| Q_1 \| Q_2)$ 成立的概率是可忽略的, 故对于 $x \notin L$, 在此种情况下证明可通过验证的概率是可忽略的.

2) 交易发送者产生 ElGamal 密文 C 时使用的交易接收者地址 pk_r 与生成一次性地址中使用的交易接收者地址 pk_r' 不一致.

在此种情况下, $pk_r \neq pk_r'$, $C_1 = g^{r_e}$, $C_2 = pk_r pk_a^{r_e}$, $P = pk_r' g^{r_t}$. 验证方验证时计算:

$$Q_2' = pk_a^{w_e} g^{w_t} P^h / C_2^h = pk_a^{s_e+hr_e} g^{s_t-hr_t} (pk_r' g^{r_t})^h / (pk_r pk_a^{r_e})^h = g^{s_t} pk_a^{s_e} (pk_r' / pk_r)^h = Q_2 (pk_r' / pk_r)^h.$$

由于 $pk_r \neq pk_r'$, 故 $Q_2' \neq Q_2$, 由于 H 是抗碰撞哈希函数, $h' = H(g \| pk_a \| C_1 \| C_2 \| P \| Q_1 \| Q_2') = h = H(g \| pk_a \| C_1 \| C_2 \| P \| Q_1 \| Q_2)$ 成立的概率是可忽略的, 故对于 $x \notin L$, 在此种情况下证明可通过验证的概率是可忽略的.

综上, 非交互交易接收者身份一致性证明满足可靠性要求. 证毕.

引理 3. 在随机预言机模型下, 交易接收者身份一致性协议具备零知识性.

证明: 模拟器 S 模拟产生模拟证明方式如下.

假设 (C_1, C_2, P) 是对某个 pk_r 生成的 ElGamal 密文和一次地址. 模拟器在未知 (r_e, r_t) 的情况下, 对 (C_1, C_2, P) 产生模拟证明的方式如下.

令随机预言机响应列表为 LIST_H.

1) 选取随机数 $h \leftarrow Z_q^*$ 作为挑战值, 选取随机数 w_e, $w_t \leftarrow Z_q^*$, 计算 $Q_1 = g^{w_e} / C_1^h$, $Q_2 = pk_a^{w_e} g^{w_t} P^h / C_2^h$.

2) 查询 $\delta = (g \| pk_a \| C_1 \| C_2 \| P \| Q_1 \| Q_2)$ 是否在 LIST_H 中, 若在, 则重新执行 1), 否则, 设置 $H(g \| pk_a \| C_1 \| C_2 \| P \| Q_1 \| Q_2) = h$, 并将 (h, δ) 记录到 LIST_H 中.

Q_1, Q_2 可通过验证, 故交易接收者身份一致性协议是零知识的. 证毕.

4.2 RRIPP 方案定义及描述

4.2.1 方案定义

RRIPP 方案由下述算法构成.

(1) 初始化算法 $RRIPP.SetUp(1^\lambda) \to (pp, sk_a)$

初始化算法通常由监管者执行, 用于生成公开参数. 算法输入为安全参数 λ, 算法输出为公开参数 pp 和监管者私钥 sk_a.

(2) 密钥生成算法 $RRIPP.KeyGen(pp) \to (sk, pk)$

密钥生成算法由用户执行, 用于用户生成公私钥. 算法输入为公开参数 pp, 输出公私钥对 (sk, pk).

(3) 交易生成算法 $RRIPP.TxGen(pp, pk_r, pk_a) \to (RRIPP.Tx)$

交易生成算法由交易发送者执行, 用于生成交易中与接收者身份隐私保护与监管相关的交易内容. 算法输入为公开参数 pp, 交易接收者公钥 pk_r 以及监管者公钥 pk_a, 算法输出为交易中接收者身份隐私保护与监管相关的交易内容 $RRIPP.Tx$.

(4) 合法性验证算法 $RRIPP.Verif(pp, pk_a, RRIPP.Tx) \to (0/1)$

合法性验证算法由共识节点和监管者执行, 用于验证交易的合法性. 算法输入为公开参数 pp, 监管者公钥 pk_a, 接收者身份隐私保护与监管相关的交易内容 $RRIPP.Tx$, 算法输出为验证结果.

(5) 交易接收算法 $RRIPP.Rev(pp, sk_r, RRIPP.Tx) \to (otsk_r / \perp)$

交易接收算法由交易接收者执行, 用于接收者识别出与自己相关的交易, 并计算一次性私钥 $otsk_r$. 算法输入为公开参数 pp, 交易接收者私钥 sk_r, 接收者身份隐私保护与监管相关的交易内容 $RRIPP.Tx$, 算法输出为交易接收者一次性私钥 $otsk_r$ 或者交易无关标识符 $\perp$.

(6) 监管算法 $RRIPP.Reg(pp, sk_a, RRIPP.Tx) \to (pk_r / \perp)$

监管算法由监管者执行, 用于监管者揭示交易接收者的真实身份. 算法输入为公开参数 pp, 监管者私钥 sk_a, 交易中接收者身份隐私保护与监管相关的交易内容 $RRIPP.Tx$, 算法输出为交易接收者的真实身份 pk_r 或者监管失

败标识 $\perp$.

4.2.2 方案描述

(1) 初始化算法 $RRIPP.SetUp(1^{\lambda}) \to (pp, sk_a)$

设 q 为素数, 选取 q 阶循环群 G, 选取生成元 $g \in G$, 选取密码学哈希函数 $H: \{0,1\}^* \to Z_q^*$. 监管者选取随机数 $r_r \in Z_q^*$ 作为监管私钥 sk_a, 计算 $pk_a = g^{r_r}$ 作为监管公钥, 监管者秘密保存监管私钥 sk_a. 设置公开参数 $pp = \{G, q, g, pk_a, H\}$. 输出公开参数 pp.

(2) 密钥生成算法 $RRIPP.KeyGen(pp) \to (sk, pk)$

选取随机数 $r_u \in Z_q^*$, 令私钥 $sk_u = r_u$, 计算公钥 $pk_u = g^{sk_u}$.

输出公私钥对 (sk_u, pk_u).

(3) 交易生成算法 $RRIPP.TxGen(pp, pk_r, pk_a) \to (RRIPP.Tx)$

依次执行下述 3 个子算法.

1) 一次性地址生成算法 $OTA.Gen$

选取随机数 $t \in Z_q^*$, 计算 $R = g^t$, 计算 $r_t = H(pk_r^t)$, 计算一次性地址 $P = pk_r g^{r_t}$.

2) 接收者监管密文生成算法 $ElG.Enc$

选取随机数 $r_e \in Z_q^*$, 计算 $C = (C_1, C_2) = (g^{r_e}, pk_r pk_a^{r_e})$.

3) 接收者身份一致性零知识证明生成算法 $ZK.ProofGen$

选取随机数 s_t, $s_e \in Z_q^*$, 计算 $Q_1 = g^{s_e}$, $Q_2 = g^{s_t} pk_a^{s_e}$, $h = H(g \| pk_a \| C_1 \| C_2 \| P \| Q_1 \| Q_2)$, $w_t = s_t - hr_t$, $w_e = s_e + hr_e$, 则身份一致性证明为 $\pi = (w_t, w_e, h)$.

输出 $RRIPP.Tx = (P, R, C, \pi)$.

(4) 合法性验证算法 $RRIPP.Verif(pp, pk_a, RRIPP.Tx) \to 0/1$

执行接收者身份一致性零知识证明生成算法 $ZK.ProofVerif$, 计算 $Q_1' = g^{w_e}/C_1^h$, $Q_2' = pk_a^{w_e} g^{w_t} P^h / C_2^h$, $h' = H(g \| pk_a \| C_1 \| C_2 \| P \| Q_1' \| Q_2')$. 若 $h = h'$, 则交易接收者身份一致性证明验证通过并输出 1, 否则验证不通过并输出 0.

(5) 交易接收算法 $RRIPP.Rev(pp, sk_r, RRIPP.Tx) \to (otsk_r/\perp)$

依次执行下述两个子算法.

1) 一次性地址验证算法 $OTA.Verif$

计算 $P' = pk_r g^{H(R^{sk_r})}$, 若 $P' = P$, 则验证成功并继续执行, 否则输出 $\perp$.

2) 一次性私钥计算算法 $OTA.OtskGen$

计算一次性私钥 $otsk_r = H(R^{sk_r}) + sk_r$, 输出 $otsk_r$.

(6) 监管算法 $RRIPP.Reg(pp, sk_a, RRIPP.Tx) \to (pk_r/\perp)$

执行接收者身份监管密文解密算法 $ElG.Dec$, 计算 $pk_r = C_2/C_1^{sk_a}$. 输出 pk_r.

4.3 RRIPP 方案安全模型及证明

4.3.1 安全模型

安全的 RRIPP 方案应该满足交易接收者身份匿名性和身份可监管性[34].

上述安全属性的形式化定义可基于仿真器 $\mathcal{S}$ 和敌手 $\mathcal{A}$ 之间的实验给出, 下面给出敌手 $\mathcal{A}$ 在实验中可查询的预言机.

(1) 查询预言机

O_R: 注册预言机, 敌手 $\mathcal{A}$ 查询此预言机注册一个新用户 I_i, 该预言机返回一个有效的用户公钥 pk_i, 并将 (pk_i, sk_i, I_i) 添加到集合 $\mathcal{L}$ 中.

O_C: 腐化成员预言机, 敌手 $\mathcal{A}$ 查询此预言机腐化公钥为 pk_i 的用户 I_i, 该预言机返回相对应的私钥 sk_i, 并将 (pk_i, sk_i, I_i) 添加到集合 $\mathcal{L}_M$ 中.

O_T: 生成交易预言机, 敌手 $\mathcal{A}$ 查询此预言机以获取接收者公钥为 pk_i 的接收者身份隐私保护与监管相关的交

易内容, 该预言机返回以 pk_i 为接收者公钥的接收者身份隐私保护与监管相关的交易内容 $RRIPP.Tx$.

(2) 交易接收者身份匿名性

交易接收者身份匿名性是指除了交易发送者、交易接收者和监管者以外, 任何用户都不能获取交易接收者的真实身份.

不可伪造性由敌手 $\mathcal{A}$ 和仿真器 $\mathcal{S}$ 之间的实验 $Expt^{\text{anon}}_{\text{RRIPP}}(\lambda)$ 定义.

初始化阶段: $\mathcal{S}$ 运行初始化算法 $pp \leftarrow RRIPP.SetUp(1^{\lambda})$, 执行监管者密钥生成算法 $(sk_a, pk_a) \leftarrow RRIPP.KeyGen(pp)$, 将公开参数 pp 和监管者公钥 pk_a 发送给 $\mathcal{A}$.

查询阶段: $\mathcal{A}$ 采用自适应策略查询预言机 O_R, O_C, O_T.

挑战阶段: $\mathcal{A}$ 将 (pk_0, pk_1) 提交给 $\mathcal{S}$, 其中 pk_0, pk_1 为两个用户的长期公钥, $\mathcal{S}$ 随机选择 $b \xleftarrow{R} \{0,1\}$ 并计算 $RRIPP.Tx_b \leftarrow RRIPP.TxGen(pp, pk_b, pk_a)$, 并将 $RRIPP.Tx_b$ 交给 $\mathcal{A}$.

猜测阶段: $\mathcal{A}$ 输出对 b 的猜测 b'.

若满足下述条件, $\mathcal{A}$ 获胜.

1) pk_0, $pk_1 \in \mathcal{L}$, 并且 pk_0, $pk_1 \notin \mathcal{L}_{\mathcal{M}}$.

2) $b = b'$.

令 $Adv^{\text{anon}}_{\text{RRIPP}}(\lambda) = \left|\Pr[Expt^{\text{anon}}_{\text{RRIPP}}(\lambda) = 1] - \frac{1}{2}\right|$ 表示敌手 $\mathcal{A}$ 打破交易接收者身份匿名性的优势.

定义 6. 交易接收者身份匿名性.

对于概率多项式时间敌手 $\mathcal{A}$, 若敌手打破方案交易接收者身份匿名性的优势 $Adv^{\text{anon}}_{\text{RRIPP}}(\lambda)$ 是可忽略的, 则 RRIPP 方案满足交易接收者身份匿名性.

(3) 交易接收者身份可监管性

交易接收者身份可监管性是指监管者可通过 RRIPP 方案恢复出交易接收者的真实身份.

交易接收者身份可监管性由敌手 $\mathcal{A}$ 和仿真器 $\mathcal{S}$ 之间的实验 $Expt^{\text{tra}}_{\text{RRIPP}}(\lambda)$ 定义.

初始化阶段: $\mathcal{S}$ 运行初始化算法 $pp \leftarrow RRIPP.SetUp(1^{\lambda})$, 执行监管者密钥生成算法 $(sk_a, pk_a) \leftarrow RRIPP.KeyGen(pp)$, 将公开参数 pp 和监管者公钥 pk_a 发送给 $\mathcal{A}$.

查询阶段: $\mathcal{A}$ 采用自适应策略查询预言机 O_R, O_C, O_T.

伪造阶段: $\mathcal{A}$ 将 $\left(RRIPP.Tx^* = (P^*, R^*, C^*, \pi^*)\right)$, 其中 $P^* = pk'_r g^{r_t}$, $C^* = (C_1 = g^{r_e}, C_2 = pk_r pk'^{r_e}_a)$, 提交给 $\mathcal{S}$.

判定阶段: $\mathcal{S}$ 运行 $pk_r / \perp \leftarrow RRIPP.Reg\left(pp, sk_a, RRIPP.Tx^*\right)$.

若满足下述条件, $\mathcal{A}$ 获胜.

1) $RRIPP.Verif\left(pp, pk_a, RRIPP.Tx^*\right)$ 验证通过.

2) $pk_r \neq pk'_r$ 或者 $pk'_a \neq pk_a$.

令 $Adv^{\text{tra}}_{\text{RRIPP}}(\lambda) = \Pr[Expt^{\text{tra}}_{\text{RRIPP}}(\lambda) = 1]$ 表示敌手 $\mathcal{A}$ 打破方案可监管性的优势.

定义 7. 交易接收者身份可监管性.

对于概率多项式时间敌手 $\mathcal{A}$, 若敌手打破方案交易接收者身份可监管性的优势 $Adv^{\text{tra}}_{\text{RRIPP}}(\lambda)$ 是可忽略的, 则 RRIPP 方案满足交易接收者身份可监管性.

4.3.2 安全性证明

(1) 交易接收者匿名性证明

定理 2. 在随机预言机模型下, 若 ElGamal 加密方案是 IND-CPA 安全的, 交易接收者身份一致性证明满足零知识性, 则 RRIPP 方案满足交易接收者匿名性.

证明: 假设 $\mathcal{A}$ 为可赢得 RRIPP 方案交易接收者匿名性实验的概率多项式时间敌手, 则仿真器 $\mathcal{S}$ 可利用 $\mathcal{A}$ 的能力, 构造打破 ElGamal 加密方案 IND-CPA 安全性, 交易接收者身份一致性证明零知识性的算法 $\mathcal{B}$.

敌手 $\mathcal{A}$ 与算法 $\mathcal{B}$ 定义之间的实验如下.

初始化阶段: $\mathcal{B}$ 运行初始化算法 $pp \leftarrow RRIPP.SetUp(1^{\lambda})$, 执行监管者密钥生成算法 $(sk_a, pk_a) \leftarrow RRIPP.KeyGen$

(pp), 将公开参数 pp 和监管者公钥 pk_a 发送给 $\mathcal{A}$.

查询阶段: $\mathcal{A}$ 采用自适应策略查询预言机 O_R, O_C, O_T. $\mathcal{B}$ 按照 ElGamal 加密方案 IND-CPA 安全性实验中的生成方式响应 O_R、O_C 预言机查询. $\mathcal{B}$ 按照 ElGamal 加密方案 IND-CPA 安全性实验中的方式模拟执行 *ElG.Enc* 算法和引理 3 中的模拟执行 *ZK.ProofGen* 算法, 由此响应 O_T.

挑战阶段: $\mathcal{A}$ 将 (pk_0, pk_1) 提交给 $\mathcal{B}$, 其中 pk_0, pk_1 为两个用户的长期公钥. $\mathcal{B}$ 将 (pk_0, pk_1) 作为挑战公钥交给 ElGamal 加密方案 IND-CPA 安全性实验中的仿真器 $\mathcal{S}_e$. $\mathcal{B}$ 从 $\mathcal{S}_e$ 处接收 ElGamal 加密方案 IND-CPA 安全性实验中的挑战密文作为接收者身份监管密文 C. $\mathcal{B}$ 随机选取 $r \xleftarrow{R} \{0,1\}$, 计算 pk_r 对应的一次性地址 P_r, R_r. 将 C, P_r, R_r 交给交易接收者身份一致性证明实验中仿真器 $\mathcal{S}_z$. $\mathcal{B}$ 从 $\mathcal{S}_z$ 处接收交易接收者身份一致性证明 π. $\mathcal{B}$ 将 C, P_r, R_r, π 作为 *RRIPP.Tx* 交给敌手 $\mathcal{A}$.

猜测阶段: $\mathcal{A}$ 输出对 b 的猜测 b'. $\mathcal{B}$ 将 b 作为对 ElGamal 加密方案 IND-CPA 安全性实验的猜测交给 $\mathcal{S}_e$. 若 $r \neq b$, 则 $\mathcal{B}$ 可将 $RRIPP.Tx_b$ 作为打破零知识证明的零知识性的实例.

假设 $\mathcal{A}$ 可以不可忽略的概率 $\epsilon_{\mathcal{A}}$ 赢得实验, 则 $\mathcal{B}$ 能够以不可忽略的概率赢得 ElGamal 加密方案 IND-CPA 安全游戏, 打破交易接收者身份一致性证明的零知识性, 与假设矛盾, 定理得证, 即方案满足匿名性. 证毕.

(2) 交易接收者身份可监管性证明

定理 3. 在随机预言机模型下, 若零知识证明满足可靠性, 则 RRIPP 方案满足交易接收者身份可监管性.

证明: 假设 $\mathcal{A}$ 为可赢得 RRIPP 方案交易接收者身份可监管性实验的概率多项式时间敌手, 则仿真器 $\mathcal{S}$ 可利用 $\mathcal{A}$ 的能力, 构造打破交易接收者身份一致性证明可靠性的算法 $\mathcal{B}$.

敌手 $\mathcal{A}$ 与算法 $\mathcal{B}$ 定义之间的实验如下.

初始化阶段: $\mathcal{B}$ 运行初始化算法 $pp \leftarrow RRIPP.SetUp(1^{\lambda})$, 执行监管者密钥生成算法 $(sk_a, pk_a) \leftarrow RRIPP.KeyGen(pp)$, 将公开参数 pp 和监管者公钥 pk_a 发送给 $\mathcal{A}$.

查询阶段: $\mathcal{A}$ 采用自适应策略查询预言机 O_R, O_C, O_T.

伪造阶段: $\mathcal{A}$ 计算 $\left(RRIPP.Tx^* = (P^*, R^*, C^*, \pi^*)\right)$, 其中 $P^* = pk'_r g^{r_t}$, $C^* = (C_1 = g^{r_e}, C_2 = pk_r pk_a^{r_e})$ 或者 $P^* = pk_r g^{r_t}$, $C^* = (C_1 = g^{r_e}, C_2 = pk_r pk_a'^{r_e})$ 提交给 $\mathcal{B}$.

假设 $\mathcal{A}$ 可以不可忽略的概率 $\epsilon_{\mathcal{A}}$ 赢得实验, 则 $\mathcal{B}$ 可将 (P^*, C^*, π^*) 作为打破交易接收者身份一致性证明可靠性的实例, 与假设矛盾, 定理得证, 即方案满足交易接收者身份可监管性. 证毕.

5 可监管的交易发送者身份隐私保护方案 RSIPP

本节首先提出了 RSIPP 方案所用的无需配对的可追踪环签名方案及其安全性证明, 随后给出了 RSIPP 方案的详细设计, 最后给出方案的安全模型及安全性证明.

5.1 可追踪环签名

5.1.1 方案描述

(1) 初始化算法 *TRS.SetUp*

设 G 为素数阶 q 的循环群, g 为 G 的生成元. 设 $H: \{0,1\}^* \rightarrow Z_q^*$ 为密码学哈希函数. 环签名追踪者选择 $sk_a \in Z_q^*$ 作为追踪者私钥, 计算 $pk_a = g^{sk_a}$ 作为追踪者公钥. 输出公开参数 $pp = \{G, q, g, pk_a, H\}$.

(2) 密钥生成算法 *TRS.KeyGen*

用户 i 随机选取 $sk_i \in Z_q^*$ 作为用户私钥, 计算用户公钥为 $pk_i = g^{sk_i}$.

(3) 环签名算法 *TRS.Sig*

环签名生成过程使用知识签名 $PoK\{\exists sk_i, i \in (1,\ldots,n) : \log_g(pk_i) = \log_{pk_a}(I)\}(M)$.

假设待签名消息为 M, 签名者生成环签名的步骤如下.

1) 签名者任意选取 $n-1$ 个用户公钥, 与自身公钥进行混合得到集合 $S = \{pk_1, pk_2, \ldots, pk_n\}$, 假设签名者为第 π 个用户 $(1 \leqslant \pi \leqslant r)$, 签名者公钥为 pk_π, 签名者私钥为 sk_π.

2) 计算 $I = pk_a^{sk_\pi}$.

3) 选取 $k \in Z_q^*$, 计算 $L_\pi = g^k$ 和 $R_\pi = pk_a^k$, 计算 $c_{\pi+1} = H(M \| pk_1 \| pk_2 \| \ldots \| pk_n \| pk_a \| I \| L_\pi \| R_\pi)$.

4) 对于 $i \in \{\pi+1, \ldots, n, 1, \ldots, \pi-1\}$, 依次随机选取 $q_i \in Z_q^*$, 计算 $L_i = g^{q_i} pk_i^{c_i}$, $R_i = pk_a^{q_i} I^{c_i}$, $c_{i+1} = H(M \| pk_1 \| pk_2 \| \ldots \| pk_n \| pk_a \| I \| L_i \| R_i)$, 令 $c_1 = c_{n+1}$.

5) 计算 $q_\pi = k - c_\pi sk_\pi \bmod q$.

6) 生成环签名 $\sigma = (S, I, c_1, q_1, \ldots, q_n)$.

(4) 环签名验证算法 *TRS.Ver*

验证者获取环签名 σ, 消息 M 和追踪者公钥 pk_a 后, 对 σ 进行验证, 验证过程如下.

1) 检查 $c_1, q_1, \ldots, q_n \in Z_q^*$ 是否成立.

2) 对于 $i \in \{1, \ldots, n\}$, 依次计算 $L_i' = g^{q_i} pk_i^{c_i}$, $R_i' = pk_a^{q_i} I^{c_i}$, $c_{i+1}' = H(M \| pk_1 \| pk_2 \| \ldots \| pk_n \| pk_a \| I \| L_i' \| R_i')$.

3) 检查 $c_1 = c_{n+1}'$ 是否成立, 若成立, 则验证通过, 算法输出 1; 否则, 验证失败, 算法输出 0.

(5) 可链接性判断算法 *TRS.Link*

验证者获取环签名 σ 中的 I, 若 I 在历史密钥镜像列表 $\mathcal{L}$ 中出现过, 则算法输出 1, 拒绝此签名; 否则, 算法输出 0, 并将 I 加入 $\mathcal{L}$ 中.

(6) 签名者身份追踪算法 *TRS.Trace*

追踪者获取环签名 σ, 对于环签名成员公钥 $pk_i \in S$, 追踪者使用追踪者私钥 sk_a, 计算 $pk_i^{sk_a}$, 若 $I = pk_i^{sk_a}$, 则输出签名者公钥 pk_i.

5.1.2 安全性证明

(1) 不可伪造性证明

定理 4. 在随机预言机模型下, 若 G 中的离散对数问题是困难的, 则方案满足不可伪造性.

证明: 假设 $\mathcal{A}$ 为可有效伪造环签名的概率多项式时间敌手, 则仿真器 $\mathcal{S}$ 可利用 $\mathcal{A}$ 的能力, 构造解决离散对数问题的算法 $\mathcal{B}$. $\mathcal{S}$ 将离散对数问题实例 $(g, Q = g^x)$ 交给算法 $\mathcal{B}$. 敌手 $\mathcal{A}$ 与算法 $\mathcal{B}$ 之间的实验定义如下.

初始化: $\mathcal{B}$ 运行初始化算法 $pp \leftarrow TRS.SetUp(1^\lambda)$, 并将公开参数 $pp = \{G, q, g, pk_a\}$ 交给敌手 $\mathcal{A}$. 算法 $\mathcal{B}$ 随机选取 n 个不同的环签名成员 $A^* = \{A_1^*, A_2^*, \ldots, A_n^*\}$, 随机生成 $x_i \in Z_q^*$ $(1 \leqslant i \leqslant n)$, 计算 $P_i^* = Q^{x_i}$ $(i \in \{1, 2, \ldots, n\})$ 作为每个环签名成员对应的公钥, 记为 $S^* = \{P_1^*, P_2^*, \ldots, P_n^*\}$, 仿真器$\mathcal{S}$ 设置 S^* 为 $\mathcal{A}$ 的目标公钥集合, 并将 S^* 发送给 $\mathcal{A}$.

查询: $\mathcal{B}$ 控制随机预言机 $\mathcal{H}$, O_R, O_C 和 O_S, $\mathcal{A}$ 查询上述预言机. 查询与应答过程如下.

1) $\mathcal{H}$ 查询: $\mathcal{B}$ 维护一个初始为空的列表 LIST_H. 对于查询 δ, $\mathcal{B}$ 查询其之前是否被查询过, 若之前被查询过, 则在 LIST_H 中找到对应的 $\tau = \mathcal{H}(\delta)$, 并返回 τ, 若之前未被查询过, 则 $\mathcal{B}$ 随机选取 $\tau \in Z_q^*$, 在 LIST_H 中添加 $\{\delta, \tau\}$, 并返回 τ.

2) O_R 查询: $\mathcal{B}$ 维护一个初始为空的列表 LIST_R. $\mathcal{A}$查询预言机O_R, 注册新用户 A_i. $\mathcal{B}$ 查询 A_i 是否属于 A^*, 若是, 则不响应查询; 否则, $\mathcal{B}$ 查询 A_i 是否被查询过, 若被查询过, 在 LIST_R 中找到对应的公钥 pk_i 并返回 pk_i; 否则, $\mathcal{B}$ 选取 $sk_i \in Z_q^*$, 计算公钥为 $pk_i = g^{sk_i}$, 在 LIST_R 中添加 $\{A_i, sk_i, pk_i\}$, 并返回 pk_i.

3) O_S 查询: $\mathcal{B}$ 维护一个初始为空的列表 LIST_S. $\mathcal{A}$ 将环签名成员公钥集合 $S = \{pk_1, pk_2, \ldots, pk_n\}$、签名者公钥 pk_π 以及消息 M' 提交给 O_S 查询环签名. 对于查询 (S, pk_π, M'), $\mathcal{B}$ 按照下述步骤模拟生成环签名.

a) 计算 $I = pk_\pi^{sk_a}$.

b) 对于 $i \in \{1, 2, \ldots, n-1\}$, 依次执行下述操作.

i) 随机选取 q_i, $c_i \in Z_q^*$, 计算 $L_i = g^{q_i} pk_i^{c_i}$ 和 $R_i = pk_a^{q_i} I^{c_i}$.

ii) 令 $\delta_{i+1} = (M \| pk_1 \| pk_2 \| \ldots \| pk_n \| pk_a \| I \| L_i \| R_i)$, 检查 δ_{i+1} 是否在列表 LIST_H 中, 若在, 则重新执行 i); 否则, 将 (δ_i, c_i) 添加到 LIST_H 中.

c) 随机选取 q_n, $c_n \in Z_q^*$, 计算 $L_n = g^{q_n} pk_n^{c_n}$ 和 $R_n = pk_a^{q_n} I^{c_n}$, 令 $\delta_{n+1} = (M \| pk_1 \| pk_2 \| \ldots \| pk_n \| pk_a \| I \| L_n \| R_n)$, 检查 δ_{n+1} 是否在列表 LIST_H 中, 若在, 则重新执行 c); 否则, 将 (δ_{n+1}, c_1) 添加到 LIST_H 中.

d) $\mathcal{B}$ 返回 $\sigma' = (I, c_1, q_1, \ldots, q_n)$ 作为应答.

4) O_C 查询: $\mathcal{A}$查询预言机O_C, 获取用户 A_i 私钥. $\mathcal{B}$ 查询 A_i 是否属于 A^*, 若是, 则不响应查询; 否则, $\mathcal{B}$ 查询 A_i 是否在 LIST_R 中, 若在, 则在 LIST_R 中找到记录中对应的 sk_i, 并返回 sk_i; 否则返回 A_i 未注册.

伪造: 敌手 $\mathcal{A}$ 以不可忽略的概率输出其所选择 (S^*, m) 的伪造签名 $\sigma = (I, c_1, q_1, \ldots, q_n)$.

若敌手 $\mathcal{A}$ 伪造的σ 为有效签名, 根据文献 [39] 定理 2, $\mathcal{B}$ 能够以不可忽略的概率得到两个有效环签名: $\sigma_0 = (I, c_1, q_1, \ldots, q_n)$ 和 $\sigma_1 = (I, c_1', q_1', \ldots, q_n')$, 在两个环签名中, 存在 $j \in \{1,2,\ldots,n\}$, 使得 $q_j \neq q_j'$, $c_j \neq c_j'$, $L_j = g^{q_j} pk_j^{c_j} = L_j' = g^{q_j'} pk_j^{c_j'}$ 成立, 则 $\log_g pk_j = \dfrac{q_j - q_j'}{c_j' - c_j} \bmod p$. 由于 $pk_j = Q^{sk_j}$, 则 $\log_g Q = x = \dfrac{(q_j - q_j')}{sk_j(c_j - c_j')} \bmod p$, 即算法 $\mathcal{B}$ 可攻破离散对数问题.

假设敌手 $\mathcal{A}$ 可以不可忽略的优势赢得实验, 则存在算法 $\mathcal{B}$ 可以不可忽略的概率解决 G 中的离散对数问题, 与假设矛盾, 定理得证, 即方案满足不可伪造性. 证毕.

(2) 匿名性证明

定理 5. 在随机预言机模型下, 若 DDH 问题是困难的, 则方案满足匿名性.

证明: 假设 $\mathcal{A}$ 为可有效伪造环签名概率多项式时间敌手, 则仿真器 $\mathcal{S}$ 可利用 $\mathcal{A}$ 的能力, 构造解决 DDH 困难问题的算法 $\mathcal{B}$. $\mathcal{S}$ 将 DDH 问题的挑战实例 (g, g^a, g^b, g^z) 交给算法 $\mathcal{B}$, 其中 g, g^a, g^b, $g^z \in G$, 若 (g, g^a, g^b, g^z) 是一个 DDH 元组, 则 $g^z = g^{ab}$ 成立. 敌手 $\mathcal{A}$ 与算法 $\mathcal{B}$ 定义之间的实验如下.

初始化: $\mathcal{B}$ 运行初始化算法, 令 $pk_a = g^a$, 并将公开参数 $pp = \{G, p, g, pk_a\}$ 发送给敌手 $\mathcal{A}$.

查询: $\mathcal{B}$ 控制随机预言机 $\mathcal{H}$ 和 O_R, $\mathcal{A}$ 查询上述预言机, 查询与应答过程如下.

1) $\mathcal{H}$ 查询: $\mathcal{H}$ 查询过程同定理 4 中的查询过程.

2) O_R 查询: $\mathcal{B}$ 维护一个初始为空的列表 LIST_R. $\mathcal{A}$查询预言机O_R, 注册新用户 A_i. $\mathcal{B}$ 查询 A_i 是否被查询过, 若被查询过, 则在 LIST_R 中找到对应的 pk_i, 并返回 pk_i; 否则, $\mathcal{B}$ 选取 $sk_i \in Z_q^*$, 计算公钥为 $pk_i = (g^b)^{sk_i}$, 在 LIST_R 中添加 $\{A_i, sk_i, pk_i\}$, 并返回 pk_i.

挑战: 敌手 $\mathcal{A}$ 任意选择消息 M^*, 将签名公钥集合 $S = \{pk_1, pk_2, \ldots, pk_n\}$, 签名公钥 pk_i 发送给算法 $\mathcal{B}$; $\mathcal{B}$ 构造相应的环签名, 操作如下.

1) $\mathcal{B}$ 选择 $b \xleftarrow{R} \{1, \ldots, n\}$, 计算 $I = (g^z)^{sk_b}$.

2) 对于 $i \in \{1,2,\ldots,n-1\}$, 依次执行下述操作.

a) 随机选取 q_i, $c_i \in Z_q^*$, 计算 $L_i = g^{q_i} pk_i^{c_i}$ 和 $R_i = pk_a^{q_i} I^{c_i}$.

b) 令$\delta_{i+1} = (M \| pk_1 \| pk_2 \| \ldots \| pk_n \| pk_a \| I \| L_i \| R_i)$, 检查 δ_{i+1} 是否在列表 LIST_H 中, 若在, 则重新执行 a); 否则, 将 (δ_i, c_i) 记录到 LIST_H 中.

3) 随机选取 q_n, $c_n \in Z_q^*$, 计算 $L_n = g^{q_n} pk_n^{c_n}$ 和 $R_n = pk_a^{q_n} I^{c_n}$, 令 $\delta_{n+1} = (M \| pk_1 \| pk_2 \| \ldots \| pk_n \| pk_a \| I \| L_n \| R_n)$, 检查 δ_{n+1} 是否在列表 LIST_H 中, 若在, 则重新执行 3); 否则, 将 (δ_{n+1}, c_1) 添加到 LIST_H 中.

4) $\mathcal{B}$ 返回 $\sigma' = (I, c_1, q_1, \ldots, q_n)$ 作为应答.

猜测: 敌手 $\mathcal{A}$ 输出 b 的猜测值 b'. 若 $b' = b$, 则 $\mathcal{B}$ 输出 $g^z = g^{ab}$, 否则 $\mathcal{B}$ 输出 $g^z \neq g^{ab}$.

假设 $\mathcal{A}$ 以不可忽略的概率 $\epsilon_{\mathcal{A}}$ 赢得实验, 则算法 $\mathcal{B}$ 能够根据 $\mathcal{A}$ 的能力可以不可忽略的概率解决 DDH 问题, 与假设矛盾, 故定理得证, 即方案满足匿名性. 证毕.

(3) 可链接性证明

定理 6. 在随机预言机模型下, 若 G 中的离散对数问题是困难的, 则方案满足可链接性.

证明: 假设 $\mathcal{A}$ 为可有效伪造环签名概率多项式时间敌手, 则仿真器 $\mathcal{S}$ 利用 $\mathcal{A}$ 的能力, 构造解决离散对数困难问题的算法 $\mathcal{B}$. $\mathcal{S}$ 将离散对数问题实例 $(g, Q = g^x)$ 交给算法 $\mathcal{B}$. 敌手 $\mathcal{A}$ 与算法 $\mathcal{B}$ 之间的实验定义如下.

初始化: $\mathcal{B}$ 运行初始化算法 $pp \leftarrow TRS.SetUp(1^\lambda)$, 并将公开参数 $pp = \{G, q, g, pk_a\}$ 交给敌手 $\mathcal{A}$.

查询: $\mathcal{B}$ 控制随机预言机 $\mathcal{H}$, O_R, O_C 和 O_S, $\mathcal{A}$ 查询上述预言机, 查询与应答过程同定理 4 中的查询过程. $\mathcal{A}$ 至多且只能对 S^* 中一个公钥进行 O_C 查询, 即 $\mathcal{A}$ 至多只有 S^* 中某一个公钥对应的私钥, 假设 $\mathcal{A}$ 拥有 S^* 中公钥 P_π 对

应的私钥 x_π.

伪造: $\mathcal{A}$ 选择2 个消息(M_0, M_1), 使用 pk_π 对应的私钥 sk_π 签名, 输出两个有效的签名$\sigma_0 = (I, c_1, q_1, \ldots, q_n)$ 和 $\sigma_1 = (I', c_1', q_1', \ldots, q_n')$, 并且$I \neq I'$.

若 $I = g^{sk_\pi} \neq I' = g^{sk_x}$, 则 $\mathcal{A}$ 在仅拥有私钥 sk_π 的情况下伪造了由私钥 $sk_x \neq sk_\pi$ 生成的签名 σ_1. 由环签名方案的不可伪造性可知, 在离散对数困难问题的假设条件下, 敌手伪造出有效签名 σ_1 的概率是可忽略的, 因此有 $I = I'$, 与假设矛盾. 定理得证, 即方案满足可链接性.

(4) 不可诽谤性证明

定理 7. 在随机预言机模型下, 若 G 中的离散对数问题是困难的, 则方案满足不可诽谤性.

证明: 假设 $\mathcal{A}$ 为可有效伪造环签名概率多项式时间敌手, 则仿真器 $\mathcal{S}$ 可利用 $\mathcal{A}$ 的能力, 构造解决离散对数困难问题的算法 $\mathcal{B}$. $\mathcal{S}$ 将离散对数问题实例 $(g, Q = g^x)$ 交给算法 $\mathcal{B}$. 敌手 $\mathcal{A}$ 与算法 $\mathcal{B}$ 定义之间的实验如下.

初始化: $\mathcal{B}$ 运行初始化算法 $pp \leftarrow TRS.SetUp(1^\lambda)$, 并将公开参数 $pp = \{G, q, g, pk_a, H\}$ 交给敌手 $\mathcal{A}$. $\mathcal{B}$ 随机选取 n 个不同的环签名成员 $A^* = \{A_1^*, A_2^*, \ldots, A_n^*\}$, 随机生成 $x_i \in Z_q^*$ $(1 \leqslant i \leqslant n)$, 计算 $P_i^* = Q^{x_i}$, $i \in \{1, 2, \ldots, n\}$ 作为每个环签名成员对应的公钥, 记为 $S^* = \{P_1^*, P_2^*, \ldots, P_n^*\}$, $\mathcal{S}$ 设置 S^* 为 $\mathcal{A}$ 的目标公钥集合, 并将 $S^* = \{P_1^*, P_2^*, \ldots, P_n^*\}$ 发送给 $\mathcal{A}$.

查询 1: $\mathcal{B}$ 控制随机预言机 $\mathcal{H}$ 和 O_R, O_C, O_S, $\mathcal{A}$ 查询上述预言机, 查询与应答过程同定理 4 中的查询过程.

挑战: $\mathcal{A}$ 将元组 (S, m, σ, pk_π) 提交给 $\mathcal{B}$, 该元组包含消息 m, 含有 n 个环签名成员公钥的集合 S, 签名者公钥 $pk_\pi \in S^*$, 其中 pk_π 未被用于查询 O_C 和 O_S 预言机. $\mathcal{B}$ 按照定理 4 证明过程中 O_S 预言机在没有 pk_π 私钥的情况下模拟生成环签名 $\sigma = (I, c_1, q_1, \ldots, q_n)$, 并将 σ' 交给 $\mathcal{A}$.

查询 2: $\mathcal{B}$ 控制随机预言机 $\mathcal{H}$ 和 O_R, O_C, O_S, $\mathcal{A}$ 查询上述预言机, pk_π 不能被用于查询 O_C 和 O_S 预言机, 查询与应答过程同定理 4 中的查询过程.

伪造: 敌手 $\mathcal{A}$ 以 ϵ 的概率输出 (m^*, σ^*, pk_π), 其中伪造签名 $\sigma^* = (I^*, c_1^*, q_1^*, \ldots, q_n^*)$ 为有效的环签名且与 σ 可链接.

假设 $\mathcal{A}$ 可以不可忽略的概率 $\epsilon_{\mathcal{A}}$ 赢得实验, 由于 σ^* 与 σ 可链接并且 σ^* 为有效的环签名, 即 $\mathcal{A}$ 伪造了由私钥 sk_π 生成的签名 σ^*. 由环签名方案的不可伪造性可知, 在离散对数困难性的假设条件下, 敌手在未知 sk_π 的情况下, 伪造出有效签名 σ^* 的概率是可忽略的, 与 pk_π 未被用于查询 O_C 和 O_S 预言机的假设矛盾. 故定理得证, 即方案满足不可诽谤性. 证毕.

(5) 可追踪性证明

定理 8. 在随机预言机模型下, 若 G 中的离散对数问题是困难的, 方案满足签名者可追踪性.

证明: 假设 $\mathcal{A}$ 为可有效打破方案可追踪性的概率多项式时间敌手, $\mathcal{A}$ 可赢得其与仿真器 $\mathcal{S}$ 之间的实验 $Expt_{\mathrm{TRS}}^{\mathrm{tra}}(\lambda)$, 则以下条件成立.

1) $pk_T \leftarrow TRS.Trace(pp, \sigma^*, sk_a)$.

2) $TRS.Ver(pp, \sigma^*, pk_a, m) = 1$, 并且 σ^* 不是 O_S 预言机的输出.

3) 对于所有 $pk_i \in S$, pk_i 是 O_R 预言机的输出.

4) $pk_T \neq pk_\pi$, pk_T 未被用于查询 O_C 预言机.

$\mathcal{S}$ 运行追踪算法得到签名者公钥 $pk_T \leftarrow TRS.Trace(pp, \sigma^*, sk_a)$, 又由 σ^* 为有效环签名, 而 σ^* 可与私钥为 sk_T 的签名者签署的环签名相链接. 由方案的不可诽谤性定义, 在离散对数困难性的假设条件下, 敌手在未知 sk_T 的情况下, 伪造出有效签名 σ^* 的概率是可忽略的. 由方案的不可诽谤性, σ^* 为私钥 sk_T 签署的有效签名与 pk_T 未被用于查询 O_C 预言机的条件矛盾, 因此, 定理得证, 即方案满足可追踪性. 证毕.

5.2 RSIPP 方案定义及描述

5.2.1 方案定义

RSIPP 方案由下述算法构成.

(1) 初始化算法 $RSIPP.SetUp(1^\lambda) \rightarrow (pp)$

初始化算法通常由监管者执行, 用于生成公开参数.

(2) 密钥生成算法 $RSIPP.KeyGen(pp) \rightarrow (sk, pk)$

密钥生成算法由用户执行, 用于用户生成公私钥对.

(3) 交易生成算法 $RSIPP.TxGen(pp, S, otsk_r, M, pk_a) \rightarrow (\sigma)$

交易生成算法由交易发送者执行, 用于生成交易中与发送者身份隐私保护与监管相关的交易内容. 算法输入为公开参数 pp, 环签名成员公钥集合 S, 交易发送者在上次交易中作为交易接收者时的一次性私钥 $otsk_r$, 签名消息 M 以及监管者公钥 pk_a, 算法输出为环签名 σ, 将 σ 作为交易合法性验证字段内容.

(4) 合法性验证算法 $RSIPP.Verif(pp, S, M, \sigma, pk_a) \rightarrow (0/1)$

合法性验证算法由共识节点和监管者执行, 用于验证 σ 的合法性. 算法输入为公开参数 pp, 环签名成员公钥集合 S, 签名消息 M, 环签名 σ, 监管者公钥 pk_a, 算法输出为验证结果.

(5) 监管算法 $RSIPP.Reg(pp, sk_a, \sigma) \rightarrow (pk_s/\perp)$

监管算法由监管者执行, 用于监管者揭示交易发送者的真实身份. 算法输入为公开参数 pp, 监管者私钥 sk_a, 环签名 σ, 算法输出为交易发送者的真实身份 pk_s 或者监管失败标识 $\perp$.

5.2.2 方案描述

(1) 初始化算法 *RSIPP.SetUp*

初始化算法即为环签名初始化算法 *TRS.SetUp*, 将监管者设置为环签名算法中的追踪者, 该算法与 *RRIPP.SetUp* 算法保持一致, 采用 *RRIPP.SetUp* 算法的输出, 即 $pp = \{G, q, g, pk_a, H\}$.

(2) 密钥生成算法 *RSIPP.KeyGen*

密钥生成算法即为环签名密钥生成算法 *TRS.KeyGen*, 该算法与 *RRIPP.KeyGen* 算法保持一致, 采用 *RRIPP.KeyGen* 算法的输出.

(3) 交易生成算法 *RSIPP.TxGen*

执行环签名算法 $TRS.Sig(pp, S, otsk_r, M, pk_a)$, 输出 σ.

(4) 合法性验证算法 *RSIPP.Verif*

执行环签名验证算法 $TRS.Sig(pp, S, M, \sigma, pk_a)$ 和环签名的链接性算法 *TRS.Link*, 输出验证结果.

(5) 监管算法 *RSIPP.Reg*

为执行环签名追踪算法 $TRS.Trace(pp, sk_a, \sigma)$, 输出监管结果.

5.3 RSIPP 方案安全模型及证明

安全的 RSIPP 方案应满足交易发送者身份匿名性和可监管性. 由于本文的 RSIPP 方案合并实现了交易有效性验证功能, 该方案还应满足不可伪造性以及抵抗“双花”攻击安全性.

由于 RSIPP 方案采用可追踪环签名方案实现, 交易发送者身份匿名性的安全模型与可追踪环签名的匿名性安全模型一致, 交易发送者身份可监管性的安全模型与可追踪环签名的不可诽谤性和可追踪性安全模型一致, 不可伪造性安全模型与可追踪环签名的不可伪造性安全模型一致, 抵抗“双花”攻击安全模型与环签名方案的可链接性安全模型一致.

可监管的交易发送者身份隐私保护方案的交易发送者身份匿名性安全性证明与可追踪环签名方案匿名性安全性证明一致, 交易发送者身份可监管安全性证明与可追踪环签名的不可诽谤性和可追踪性安全性证明一致, 不可伪造性安全性证明与可追踪环签名的不可伪造性安全性证明一致, 抵抗“双花”攻击安全性证明与环签名方案的可链接性安全性证明一致.

6 方案实现测试

本节首先对本文提出的可监管交易接收者身份隐私保护方案 RRIPP 与可监管交易发送者身份隐私保护方案 RSIPP 进行实现测试, 随后分别从理论层面和实现层面对比本文方案与已有方案的性能.

6.1 实现与测试

本节以 RRIPP、RSIPP 方案安全强度为 128 bit 为例进行实现测试. 实验环境采用的处理器为 Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40 GHz, 内存为 320 MB, 操作系统为 Ubuntu 20.04.2. 实验采用 Go 语言编写测试程序, Go 语言版本为 1.23.3.

调用 Go 语言 Crypto 库[40]中的 P256 椭圆曲线和椭圆曲线上的点加和标量乘运算实例化 RRIPP 方案中的 ElGamal 加密算法、一次性地址算法和零知识证明, 以及 RSIPP 方案的环签名算法中的群及群上运算, 调用 Crypto 库中的 SHA256[41]实例化 RRIPP 与 RSIPP 方案中的哈希算法.

测试过程中, 根据 Monero 中的设置令 RSIPP 算法环签名成员数量为 16, 对 RRIPP 与 RSIPP 方案中的每个算法分别运行 1 000 次, 每个算法的单次执行时间如表 2.

表 2 RRIPP 与 RSIPP 方案中的算法测试结果 (ms)

方案	算法	时间
RSIPP	TIME_TxGen	4.53
	TIME_Verify	4.62
	TIME_Reg	0.66
RRIPP	TIME_TxGen	0.68
	TIME_Verify	0.71
	TIME_Rev	0.33
	TIME_Reg	0.17

将 RSIPP 算法中的环签名成员数量分别设置为 2–20, 对 RSIPP 方案中的每个算法分别运行 1 000 次, 记录运行时间计算平均值, RSIPP 方案环签名成员数量与各算法性能关系的测试结果如图 3 所示. 由图 3 易见, 方案中的各算法运行时间随环签名成员数量增加呈线性增长趋势.

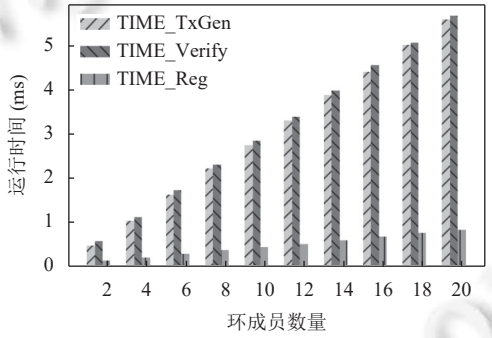

图 3 环签名成员数量与算法性能关系图

根据测试结果, RSIPP 算法设置环签名成员数量为 16 时, 可监管身份隐私保护算法的执行时间均在 5 ms 以内.

6.2 与已有方案的对比

本节首先从理论层面分别给出 RRIPP 方案和 RSIPP 方案与采用类似技术路线方案的性能对比. 文献 [28,34] 中的交易接收者隐私保护与监管方案与本文中的 RRIPP 方案技术路线类似, 均采用一次性地址实现交易接收者身份隐私保护、使用监管公钥加密接收者身份结合零知识证明实现对接收者身份的有效监管, RRIPP 方案理论层面的性能主要与上述文献中的方案进行对比. 文献 [33,34] 中的交易发送者隐私保护与监管方案与本文中的 RSIPP 方案技术路线类似, 均采用环签名实现交易发送者身份隐私保护、结合监管公钥生成交易发送者追踪消息实现对交易发送者身份的监管, RSIPP 方案理论层面的性能主要与上述文献中的方案进行对比. 对于消息长度的对比, 统计方案各字段中使用的各类元素数量, 并结合各类元素长度, 评估各字段长度. 对于计算开销的对比, 统计方案中使用的各类耗时运算的执行次数, 并通过实验测试给出各类耗时运算单次执行时间, 预估各算法的运行时间的近似值.

由于本文中的 RRIPP 方案和 RSIPP 方案的技术路线与文献 [34] 中的方案均类似, 本节基于实验测试结果重点给出本文的方案与文献 [34] 方案的整体性能对比.

6.2.1 理论层面对比

为了便于比较, 本节设置文献 [28,33,34] 和本文中的方案安全强度均为 128 bit, 结合各文献中的设置实例化各方案, 在此基础上对比分析方案性能, 即文献 [33,34] 采用 Pbc 库[41]中的类型 A 配对 $e(G_1\times G_1)\to G_T$, 群 G_1、G_T 阶的位长设置为 256 bit, 文献 [28] 与本文采用 P256 椭圆曲线 G, 群 G 阶的位长设置为 256 bit. 文献 [28] 中的 zk-SNARK 选用具有 128 bit 安全强度且零知识证明长度较短的 Groth16 方案[14]进行实例化, zk-SNARK 证明长度约为 1 526 bit. 各类元素的长度如表 3 所示. 基于 Go 语言 Pbc 库、Crypto 库实例化上述配对 e 与椭圆曲线 G, 通过执行配对 e 与椭圆曲线 G 上的各类主要耗时运算各 1 000 次, 计算各类主要耗时运算平均执行时间, 结果如表 3 所示. 由于文献 [28] 中未给出 l 的建议取值, 考虑到交易接收者隐私保护和方案效率之间的平衡, 此处设 l=3. 根据 Monero 中的设置令 RSIPP 算法环签名成员数量为 16. 符号定义如表 3 所示.

表 3 符号定义

符号	含义	长度或平均单次执行时间
$\|G\|$, $\|G_1\|$, $\|G_T\|$	分别表示群 G、G_1、G_T 中的元素长度	$\|G\|$=512 bit, $\|G_1\|$=1024 bit, $\|G_T\|$=1024 bit
q, q_1, q_T	分别表示群 G、群 G_1 和群 G_T 的阶	位长为256 bit
$\|Z_q\|$, $\|Z_{q_1}\|$	分别表示有限域 Z_q、Z_{q_1} 中的元素长度	$\|Z_q\|$=256 bit, $\|Z_{q_1}\|$=256 bit
|zk-SNARK|	zk-SNARK零知识证明长度	|zk-SNARK|=1536 bit
BP	执行1次 $e(G_1\times G_1)\to G_T$ 配对运算的时间	1.31 ms
PM_G	执行1次群 G 中标量乘法运算的时间	0.07 ms
PM_{G_1}	执行1次群 G_1 中标量乘法运算的时间	1.84 ms
M_{G_T}	执行1次群 G_T 中模幂运算的时间	0.18 ms
HTP	执行1次hash to point运算的时间	1.87 ms
l	用于隐藏真实接收者的账户数量	3
n	环签名成员数量	16

RRIPP 方案与已有类似方案的性能对比如表 4、表 5 所示. 本文中的 RRIPP 方案较文献 [28,34] 中的方案具有较短的长度与较高的运算效率.

表 4 RRIPP 方案与已有类似方案的消息长度对比

字段类型	文献[28]	文献[34]	本方案
接收者地址长度 (含接收者身份监管信息)	$3l\|G\|+\|$zk-SNARK$\|$	$5\|G_1\|+4\|Z_{q_1}\|$	$4\|G\|+3\|Z_q\|$
实际长度 l=3 (bit)	6 144	6 144	2 816

表 5 RRIPP 方案与已有类似方案的运算效率对比

效率对比项	文献[28]	文献[34]	本方案
交易接收计算量	$l(BP+2HTP)$	$2PM_{G_1}$	$2PM_G$
交易接收时间近似值 (ms)	≈15.15	≈3.68	≈0.14
交易接收者身份监管计算量	lPM_{G_1}	$2PM_{G_1}$	PM_G
交易接收者身份监管时间近似值 (ms)	≈5.52	≈3.68	≈0.07

RSIPP 方案与已有类似方案的性能对比如表 6 和表 7 所示. 本文中的 RSIPP 方案较文献 [33,34] 中的方案具有较短的长度与较高的运算效率. 文献 [33] 方案中的交易生成和交易验证算法执行时间不随环签名成员数量线性增长, 当环签名成员数量 n>105 时, 文献 [33] 方案中的一些算法效率优于 RSIPP 方案中的相应算法. 当前 Monero 中环签名成员数量设置为 16, 可在保证发送者身份匿名性的同时避免消息长度和运算效率的过度增加, 在当前及未来的一段时间内, RSIPP 方案的性能较文献 [33] 中的相应方案仍具有较大优势.

表 6 RSIPP 方案与已有类似方案的消息长度对比

字段类型	文献[33]	文献[34]	本方案
发送者地址长度 (环签名成员公钥长度)	$n\|G_1\|$	$n\|G_1\|$	$n\|G\|$
发送者地址实际长度 (n=16) (bit)	16 384	16 384	8 192
发送者身份监管信息长度 (理论值)	$6\|G_1\|+3\|G_T\|+11\|Z_{q_1}\|$	$\|G_T\|+2n\|Z_{q_1}\|$	$\|G\|+(n+1)\|Z_q\|$
发送者身份监管信息实际长度 (n=16) (bit)	12 032	9 216	4 864

表 7 RSIPP 方案与已有类似方案的运算效率对比

效率对比项	文献[33]	文献[34]	本方案
交易生成计算量	$2HTP+14PM_{G_1}+7BP+6M_{G_T}$	$(2n-1)PM_{G_1}+1BP+(2n-1)M_{G_T}$	$(4n-1)PM_G$
交易生成时间近似值 (ms)	≈39.75	≈63.93	≈4.41
交易验证计算量	$8PM_{G_1}+10BP+10M_{G_T}$	$2nPM_{G_1}+1BP+1M_{G_T}$	$4nPM_G$
交易验证时间近似值 (ms)	≈29.62	≈60.37	≈4.48
交易发送者身份监管计算量	$\frac{n}{2}(BP+2HTP)$	$\frac{n}{2}(M_{G_T}+BP)$	$\frac{n}{2}PM_G$
交易发送者身份监管时间近似值 (ms)	≈40.4	≈11.92	≈0.56

6.2.2 实验测试对比

采用与第 6.2.1 节中相同的方案实例化设置, 本文的可监管身份隐私保护方案消息长度与文献 [34] 中的对比如表 8 所示, 当环签名成员数量为 16 时, 本文的方案消息总体长度仅为文献 [34] 中方案的一半, 本文的方案具有更短的消息长度.

表 8 可监管身份隐私保护方案的消息长度对比

方案	身份隐私保护与监管相关消息总长度
文献[34]	$(n+5)\|G_1\|+\|G_T\|+(2n+4)\|Z_{q_1}\|$
本方案	$(n+5)\|G\|+(n+4)\|Z_q\|$
长度对比 (n=16)	50%

为对比本文的方案与文献 [34] 中方案的效率, 调用 Go Pbc 库实现文献 [34] 方案群 G_1、G_T 中的运算, 调用 Go crypto 库中的 SHA256 实例化, 实现文献 [34] 中的哈希算法. 令环签名成员数量为 16, 每种算法执行 1 000 次, 取算法单次执行时间的平均值. 本文的可监管身份隐私保护方案与文献 [34] 中的运算效率对比如表 9 所示, 表中最后一行效率对比为文献 [34] 中各算法执行时间与本文中各算法执行时间的比值, 本文方案中的各算法效率是文献 [34] 中的 14 倍以上, 各算法运算效率均优于文献 [34] 中的各算法.

表 9 可监管身份隐私保护方案的运算效率对比

方案	RRIPP				RSIPP		
	交易生成时间	交易验证时间	交易接收时间	交易监管时间	交易生成时间	交易验证时间	交易监管时间
本方案 (ms)	0.68	0.71	0.33	0.17	4.53	4.62	0.66
文献[34] (ms)	18.72	15.11	5.67	3.67	65.79	66.60	12.90
效率对比	27.53	21.28	17.19	21.58	14.52	14.42	19.55

7 总 结

本文设计了一种高效的区块链中可监管身份隐私保护方案, 该方案采用一次性地址, ElGamal 加密算法, 以及本文设计的一种无需配对的接收者身份一致性零知识证明以及无需配对的可追踪环签名等密码原语, 不仅可有效地保护用户在交易过程中的身份隐私, 还确保了监管机构可在必要时对用户身份隐私进行有效的监管. 实现测试证实了本方案具有较高的效率, 相较于其他类似方案, 本方案在计算效率和消息长度方面均展现出了明显的优势.

References

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. 2008. https://nakamotoinstitute.org/library/bitcoin/

[2] Maidamwar P, Saraf P, Chavhan N. Blockchain applications, challenges, and opportunities: A survey of a decade of research and future outlook. In: Proc. of the 2021 Int'l Conf. on Computational Intelligence and Computing Applications (ICCICA). Nagpur: IEEE, 2021. 1–5. [doi: 10.1109/ICCICA52458.2021.9697256]

[3] Central Cyberspace Affairs Commission Data and Technology Security Center. China blockchain innovative application development report (2023). Beijing: Cyberspace Administration of China, 2024 (in Chinese).

[4] Yao Q, Zhang DW. Survey on identity management in blockchain. Ruan Jian Xue Bao/Journal of Software, 2021, 32(7): 2260–2286 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6309.htm [doi: 10.13328/j.cnki.jos.006309]

[5] Bonneau J, Narayanan A, Miller A, Clark J, Kroll JA, Felten EW. Mixcoin: Anonymity for Bitcoin with accountable mixes. In: Proc. of the 18th Int'l Conf. on Financial Cryptography and Data Security. Christ Church: Springer, 2014. 486–504. [doi: 10.1007/978-3-662-45472-5_31]

[6] Gregory M. CoinJoin: Bitcoin privacy for the real world. 2013. https://bitcointalk.org/index.php?topic=279249.0

[7] Selij JW. CoinShuffle anonymity in the block chain. 2015. https://rp.os3.nl/2014-2015/p77/presentation.pdf

[8] Ziegeldorf JH, Grossmann F, Henze M, Inden N, Wehrle K. CoinParty: Secure multi-party mixing of Bitcoins. In: Proc. of the 5th ACM Conf. on Data and Application Security and Privacy. San Antonio: ACM, 2015. 75–86. [doi: 10.1145/2699026.2699100]

[9] Bissias G, Ozisik AP, Levine BN, Liberatore M. Sybil-resistant mixing for Bitcoin. In: Proc. of the 13th Workshop on Privacy in the Electronic Society. Scottsdale: ACM, 2014. 149–158. [doi: 10.1145/2665943.2665955]

[10] Andy G. 'Dark wallet' is about to make Bitcoin money laundering easier than ever. 2014. http://www.makingmoneyuncovered.com/make-money-online/dark-wallet-is-about-to-make-bitcoin-money-laundering-easier-than-ever/

[11] Duffield E. Dash: A privacy-centric cryptocurrency. 2017. https://satoshiwatch.com/dash/

[12] Miers I, Garman C, Green M, Rubin AD. Zerocoin: Anonymous distributed e-cash from Bitcoin. In: Proc. of the 2013 IEEE Symp. on Security and Privacy. Berkeley: IEEE, 2013. 397–411. [doi: 10.1109/SP.2013.34]

[13] Sasson EB, Chiesa A, Garman C, Green M, Miers I, Tromer E, Virza M. Zerocash: Decentralized anonymous payments from Bitcoin. In: Proc. of the 2014 IEEE Symp. on Security and Privacy. Berkeley: IEEE, 2014. 459–474. [doi: 10.1109/SP.2014.36]

[14] Groth J. On the size of pairing-based non-interactive arguments. In: Proc. of the 35th Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Vienna: Springer, 2016. 305–326. [doi: 10.1007/978-3-662-49896-5_11]

[15] Ben-Sasson E, Chiesa A, Riabzev M, Spooner N, Virza M, Ward NP. Aurora: Transparent succinct arguments for R1CS. In: Proc. of the 38th Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Darmstadt: Springer, 2019. 103–128. [doi: 10.1007/978-3-030-17653-2_4]

[16] Ben-Sasson E, Bentov I, Horesh Y, Riabzev M. Scalable, transparent, and post-quantum secure computational integrity. 2018. https://eprint.iacr.org/2018/046.pdf

[17] Sun SF, Au MH, Liu JK, Yuen TH. RingCT 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency Monero. In: Proc. of the 22nd European Symp. on Research in Computer Security. Oslo: Springer, 2017. 456–474. [doi: 10.1007/978-3-319-66399-9_25]

[18] Liu JK, Wong DS. Linkable ring signatures: Security models and new schemes. In: Proc. of the 2005 Int'l Conf. on Computational Science and Its Applications. Singapore: Springer, 2005. 614–623. [doi: 10.1007/11424826_65]

[19] Noether S. Review of cryptonote white paper. 2014. https://downloads.getmonero.org/whitepaper_review.pdf

[20] Ateniese G, Faonio A, Magri B, de Medeiros B. Certified Bitcoins. In: Proc. of the 12th Int'l Conf. on Applied Cryptography and Network Security. Lausanne: Springer, 2014. 80–96. [doi: 10.1007/978-3-319-07536-5_6]

[21] Wu YB, Fan HN, Wang XY, Zou GN. A regulated digital currency. Science China Information Sciences, 2019, 62(3): 32109. [doi: 10.1007/s11432-018-9611-3]

[22] Lin C, Huang XY, Ning JT, He DB. ACA: Anonymous, confidential and auditable transaction systems for blockchain. IEEE Trans. on Dependable and Secure Computing, 2023, 20(6): 4536–4550. [doi: 10.1109/TDSC.2022.3228236]

[23] Garman C, Green M, Miers I. Accountable privacy for decentralized anonymous payments. In: Proc. of the 20th Int'l Conf. on Financial Cryptography and Data Security. Berlin: Springer, 2016. 81–98. [doi: 10.1007/978-3-662-54970-4_5]

[24] Lin C, He DB, Huang XY, Khan MK, Choo KKR. DCAP: A secure and efficient decentralized conditional anonymous payment system based on blockchain. IEEE Trans. on Information Forensics and Security, 2020, 15: 2440–2452. [doi: 10.1109/TIFS.2020.2969565]

[25] Li YN, Susilo W, Yang GM, Yu Y, Du XJ, Liu DX, Guizani N. Toward privacy and regulation in blockchain-based cryptocurrencies. IEEE Network, 2019, 33(5): 111–117. [doi: 10.1109/MNET.2019.1800271]

[26] Androulaki E, Camenisch J, de Caro A, Dubovitskaya M, Elkhiyaoui K, Tackmann B. Privacy-preserving auditable token payments in a permissioned blockchain system. In: Proc. of the 2nd ACM Conf. on Advances in Financial Technologies. New York: ACM, 2020. 255–267. [doi: 10.1145/3419614.3423259]

[27] Yuen TH. PAChain: Private, authenticated & auditable consortium blockchain and its implementation. Future Generation Computer

Systems, 2020, 112: 913–929. [doi: 10.1016/j.future.2020.05.011]
[28] Li YN, Yang GM, Susilo W, Yu Y, Au MH, Liu DX. Traceable Monero: Anonymous cryptocurrency with enhanced accountability. IEEE Trans. on Dependable and Secure Computing, 2021, 18(2): 679–691. [doi: 10.1109/TDSC.2019.2910058]
[29] Hu JW, Huang KQ, Bian GQ, Cui YP. Redact4Trace: A solution for auditing the data and tracing the users in the redactable blockchain. Computer Networks, 2024, 245: 110360. [doi: 10.1016/j.comnet.2024.110360]
[30] Sarencheh A, Kiayias A, Kohlweiss M. PARScoin: A privacy-preserving, auditable, and regulation-friendly stablecoin. In: Proc. of the 23rd Int'l Conf. on Cryptology and Network Security. Cambridge: Springer, 2024. 289–313. [doi: 10.1007/978-981-97-8013-6_13]
[31] Li YX, Weng J, Lai JZ, Li YJ, Wu JH, Li M, Sun JF, Wu PF, Deng RH. AuditPCH: Auditable payment channel hub with privacy protection. IEEE Trans. on Information Forensics and Security, 2025, 20: 1251–1261. [doi: 10.1109/TIFS.2024.3515820]
[32] Zhaolu TY, Wan ZG, Wang HQ. Division of regulatory power: Collaborative regulation for privacy-preserving blockchains. IEEE Trans. on Information Forensics and Security, 2024, 19: 2533–2548. [doi: 10.1109/TIFS.2023.3348268]
[33] Yang XY, Hou JP, Xu L, Zhu LH. zkFabLedger: Enabling privacy preserving and regulatory compliance in hyperledger fabric. IEEE Trans. on Network and Service Management, 2025, 22(2): 2243–2263. [doi: 10.1109/TNSM.2024.3525045]
[34] Song JW, Zhang DW, Han X, Du Y. Supervised identity privacy protection scheme in blockchain. Ruan Jian Xue Bao/Journal of Software, 2023, 34(7): 3292–3312 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6517.htm [doi: 10.13328/j.cnki.jos.006517]
[35] Tsiounis Y, Yung M. On the security of ElGamal based encryption. In: Proc. of the 1st Int'l Workshop on Practice and Theory in Public Key Cryptography. Pacifico Yokohama: Springer, 1998. 117–134. [doi: 10.1007/BFb0054019]
[36] Fiat A, Shamir A. How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko AM, ed. Advances in Cryptology—CRYPTO 1986. Berlin: Springer, 1987. 186–194. [doi: 10.1007/3-540-47721-7_12]
[37] Fan Q, He DB, Luo M, Huang XY, Li DW. Ring signature schemes based on SM2 digital signature algorithm. Journal of Cryptologic Research, 2021, 8(4): 710–723 (in Chinese with English abstract). [doi: 10.13868/j.cnki.jcr.000472]
[38] Boneh D. Schnorr digital signature scheme. In: van Tilborg HCA, Jajodia S, eds. Encyclopedia of Cryptography and Security. New York: Springer, 2011. 1082–1083. [doi: 10.1007/978-1-4419-5906-5_154]
[39] Herranz J, Sáez G. Forking lemmas for ring signature schemes. In: Proc. of the 4th Int'l Conf. on Cryptology in India. New Delhi: Springer, 2003. 266–279. [doi: 10.1007/978-3-540-24582-7_20]
[40] Go. Crypto. 2025. https://pkg.go.dev/crypto
[41] Pbc. The PBC Go Wrapper. 2018. https://pkg.go.dev/github.com/Nik-U/pbc

附中文参考文献

[3] 中华人民共和国国家互联网信息办公室. 中国区块链创新应用发展报告 (2023). 北京: 中华人民共和国国家互联网信息办公室, 2024.
[4] 姚前, 张大伟. 区块链系统中身份管理技术研究综述. 软件学报, 2021, 32(7): 2260–2286. http://www.jos.org.cn/1000-9825/6309.htm [doi: 10.13328/j.cnki.jos.006309]
[34] 宋靖文, 张大伟, 韩旭, 杜晔. 区块链中可监管的身份隐私保护方案. 软件学报, 2023, 34(7): 3292–3312. http://www.jos.org.cn/1000-9825/6517.htm [doi: 10.13328/j.cnki.jos.006517]
[37] 范青, 何德彪, 罗敏, 黄欣沂, 李大为. 基于 SM2 数字签名算法的环签名方案. 密码学报, 2021, 8(4): 710–723. [doi: 10.13868/j.cnki.jcr.000472]

作者简介

苏航, 博士生, 主要研究领域为密码学, 区块链.
郭兆中, 博士, 教授, 主要研究领域为区块链, 公钥密码学, 共识算法, 跨链机制.
徐茂智, 博士, 教授, 博士生导师, 主要研究领域为数学, 密码学, 信息安全.