软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2023,34(7):3292−3312 [doi: 10.13328/j.cnki.jos.006517]

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

区块链中可监管的身份隐私保护方案*

宋靖文[1,2], 张大伟[1,2], 韩 旭[1,2], 杜 晔[1,2]

[1](北京交通大学 计算机与信息技术学院, 北京 100044)

[2](智能交通数据安全与隐私保护技术北京市重点实验室 (北京交通大学), 北京 100044)

通信作者: 张大伟, E-mail: dwzhang@bjtu.edu.cn

摘 要: 在账本公开、多方共识情况下确保交易身份的隐私保护是区块链技术面临的主要挑战之一. 目前公有链中基于匿名认证和交易混淆的身份隐私保护方案由于缺乏监管又难于在行业应用中推广. 借鉴门罗币中的身份隐私保护方案, 引入监管方的角色, 基于一次性地址加密和零知识证明设计了可监管的交易接收方身份隐私保护方案; 结合可链接环签名和可撤销环签名设计了可链接可撤销环签名方案, 以实现基于自主混淆的可监管交易发送方身份隐私保护方案. 基于上述方案, 系统在保护交易方身份隐私的同时, 还支持监管方可离线恢复交易参与方的真实身份, 从而达到“可控匿名”的监管目的. 分析和测试结果表明, 方案设计的算法运算时间均为毫秒级, 可满足区块链非高频交易场景下的性能需求.

关键词: 区块链; 密码货币; 隐私保护; 零知识证明; 环签名

中图法分类号: TP309

中文引用格式: 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

英文引用格式: 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). http://www.jos.org.cn/1000-9825/6517.htm

Supervised Identity Privacy Protection Scheme in Blockchain

SONG Jing-Wen[1,2], ZHANG Da-Wei[1,2], HAN Xu[1,2], DU Ye[1,2]

[1](School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

[2](Beijing Key Laboratory of Security and Privacy in Intelligent Transportation (Beijing Jiaotong University), Beijing 100044, China)

Abstract: One of the main challenges of blockchain technology is to ensure the privacy protection of transaction identity under the condition of open ledger and multi-party consensus. At present, the identity privacy protection scheme based on anonymous authentication and transaction mixing in public blockchain is difficult to be popularized in the industry due to the lack of supervision. Based on the identity privacy protection scheme in Monero, this study introduces the role of the regulator, designs a supervised privacy protection scheme for the transaction receiver based on one-time address encryption and zero knowledge proof. It also designs a linkable revocable ring signature scheme based on linkable ring signature and revocable ring signature so as to implement the supervised privacy protection scheme for transaction sender based on autonomous mixing. The scheme can not only protect the identity privacy of the participants, but also support the offline transaction identity recovery for the regulator so as to achieve the regulatory purpose of “controllable anonymity”. The analysis and test results show that the algorithm operation time is millisecond in this scheme, which can meet the performance requirements of blockchain in non-high frequency transaction scenarios.

Key words: blockchain; cryptocurrency; privacy protection; zero knowledge proof; ring signature

---

* 基金项目: 国家重点研发计划 (2020YFB1005604, 2020YFB2103802)
收稿时间: 2021-05-21; 修改时间: 2021-07-20; 采用时间: 2021-10-14; jos 在线出版时间: 2022-11-30
CNKI 网络首发时间: 2022-12-01

1 引 言

区块链技术由于其去中心化、不可篡改、可追溯等特点引起了国内外的广泛关注, 并在密码货币、金融、存证、溯源等领域得到了广泛应用[1]. 然而, 区块链系统所采用的账本公开、多方确认的交易模式也对用户的身份隐私带来了极大的威胁. 比特币、门罗币和零币等密码货币采用未花费交易 (unspent transaction output, UTXO) 的表示方法来代表特定属主一定数值的数字资产. 对手方之间的一笔交易由若干的输入和输出 UTXO 组成, 输入的 UTXO 代表交易发送方支出的数字资产, 输出的 UTXO 代表交易接收方收到的数字资产, 可在将来作为输入的 UTXO 来使用. 载有交易身份信息的 UTXO 以编码形式在公开账本中记录和确认数字资产, 并清晰表达了交易的输入输出关系以及权属变更过程. 这一过程泄漏了大量的身份隐私信息, 如: 同一交易的多个输入可能属于使用不同地址 (公钥) 的同一所有者; 交易输出的找零地址同输入地址为同一持有人等. 这些特点都为身份追踪提供了极大的便利. 账本公开情况下缺乏有效的身份隐私保护极大阻碍了区块链技术在金融支付等强隐私需求领域的应用.

基于上述区块链系统中交易模型的特点, 我们首先给出区块链身份隐私保护的定义: 区块链系统中的身份隐私是指观察者无法将某笔区块链交易与交易参与方的真实身份关联在一起. 这一定义包括 2 方面的内容: 首先, 观察者无法确认某笔交易发送方和接收方的身份; 其次, 系统应确保交易身份的不可链接性, 即观察者无法将同一用户的不同身份链接起来; 观察者也无法将一笔交易的发送方和接收方的身份链接起来. 在具体交易场景中, 交易身份隐私保护又可细分为交易发送方的身份隐私保护和交易接收方的身份隐私保护. 此外, 随着区块链技术在行业应用中的普及, 隐私保护同身份监管之间的矛盾也日益凸显. 基于公有链的密码货币采用无准入的匿名认证机制, 并通过混淆等手段提供交易身份的强隐私保护. 这降低了区块链系统公开记账所带来的隐私泄露风险, 但也为非法交易提供了便利. 密码货币在“暗网”交易和“勒索软件”中的广泛使用使得公有链的交易监管问题日益凸显. 不同于密码货币中的强隐私保护需求, 在金融支付、证券交易等区块链行业应用中为了防止非法交易和网络犯罪, 系统在保护交易方身份隐私的同时, 必须支持监管方在必要情况下 (如交易核查、调查取证等) 可有效恢复出交易方的真实身份, 从而实现前台匿名、后台实名的“可控匿名”机制. 如何实现“可控匿名”的监管目的也是目前区块链行业应用中亟待解决的关键问题[2].

为了解决交易过程中的身份隐私保护问题, 现有的区块链系统提供了多种隐私保护方案. 公有链中的比特币[3]、门罗币[4]和零币[5]等采用无准入机制的开放网络架构, 为用户提供了匿名身份认证. 系统中无类似 CA 的可信认证中心, 用户基于非对称密码算法自主生成和管理身份, 即采用公钥作为身份标识符、私钥签名实现身份认证. 此外, 为了实现交易过程中的身份隐私保护, 同一用户还可通过使用多个不同的身份标识符 (假名) 来降低交易地址与真实身份以及不同交易之间的关联性. 虽然匿名认证能够在一定程度上保护身份隐私, 但由于区块链交易系统账本公开、多方确认的特点, 简单使用匿名认证并不能完全有效地解决交易身份隐私保护的问题. 首先, 无论是在基于 UTXO 还是基于账户的交易系统中, 都必须采用身份标识来表示用户对数字资产的权属, 并采用相应的认证机制 (如数字签名) 来确保支付行为可确认. 这必然使得用户身份标识公开暴露于区块链账本上; 其次, 公开账本记录的交易流程也揭示了不同交易身份间的关联性. 例如, 在比特币 UTXO 交易模型中, 每笔交易的输入都对应某笔历史交易的输出; 一笔交易的多输入地址多属于同一用户; 找零地址同输入地址属于同一用户. 这些特点都泄漏了大量的用户身份隐私. 已有研究表明, 即使交易参与方每笔交易都生成新的区块链地址来参与交易, 攻击者仍可根据交易特征找到属于同一用户的区块链地址[6,7], 甚至能够通过交易规律恢复出交易参与方的真实身份. 为了解决上述问题, 相关工作在比特币交易模型的基础上提出了基于“交易混淆”的改进方案. 协同混币机制通过在交易过程中引入一组主动参与的并发交易用户, 并通过混淆器 (Mixer) 混淆来共同完成交易, 从而实现交易输入、输出地址对应关系的隐藏, 使得观察者无法通过账本分析获得身份隐私信息. 协同混币又包括中心化混币和去中心化混币, 中心化混币需要借助第三方才能完成, 而去中心化混币则无需借助第三方, 因此, 相较于中心化混币, 去中心化混币方案更安全、易操作. Mixcoin[8]和 BlindCoin[9]都是中心化混币, 不但要借助第三方, 还会产生额外的服务费用. CoinJoin[10]是一种去中心化的混币方案, 但需要参与混币的用户进行协商才能完成混币操作, 参

与协商的用户有可能会泄露其他用户的交易隐私. CoinShuffle[11]是 CoinJoin 的改进, 即使是参与混币的用户也不能获取其他用户的交易隐私, 但 CoinShuffle 要求参与混币的用户同时在线, 容易受到拒绝服务攻击 (denial of service, DOS) 攻击[12]. 由麻省理工学院和霍普金斯大学的研究者发起的零币 (Zcash)[5]则采用了一种全局混币技术来实现身份隐私保护. 零币在交易结构中隐去了交易地址, 通过引入承诺 Merkle 树、零知识证明和加密技术提供了更强的隐私保护特性, 它将传统电子货币的防伪验证由中心机构的签名验签问题转变为一个已发行货币列表的成员证明问题, 从而将交易发送方的当前支付货币隐藏在整个货币列表中, 并通过零知识证明实现货币的权属确认. 零币避免了协同混币中需要并发交易用户主动参与的问题, 通过货币列表实现了全局自主混币. 但由于系统大量引入了 zk-SNARK 零知识证明算法, 因此增加了交易操作的时间和空间复杂度; 此外, 零币的隐私保护方案无法实现用户身份的有效恢复和监管. 门罗币[4]同样采用了自主混币技术来实现用户身份的隐私保护. 与零币不同的是, 其利用环签名技术将交易发送方身份隐藏在匿名集合中, 同时采用一次性地址隐藏交易接收方的真实身份, 从而达到隐私保护的目的. 门罗币的身份隐藏机制可由交易发送方自主完成, 无需其他用户协同发起交易, 因此具有很好的灵活性, 但门罗币同样未考虑用户身份的监管问题. 文献 [13] 基于 Ring CT[14]设计了可追踪的门罗币交易机制, 通过对输入交易的分组索引追踪实现用户身份恢复. 但这一方案由于采用一次交易的全部输入分组为单位进行多组混淆, 因此容易暴露同一分组内交易的可链接性 (属于同一用户) 从而降低了隐私保护力度. 此外, 文献 [15] 提出了一种基于 identity mixer[16]和群签名构建的可监管匿名认证方案, 但并未给出在区块链系统中标识 UTXO 权属和确权支付的方式, 也没有给出交易混淆的身份隐藏机制. 文献 [17] 提出了一种基于 SM9 和群签名的区块链身份隐私方案, 但这一方案在支付过程中需交易双方在链下进行 2 次群签名交互, 且每次交易接收方需保存 UTXO 的部分群签名要素, 这带来了额外的通信和存储开销. 此外, 这一方案也未提供交易混淆机制. 文献 [18] 提出了一种基于 ZK-SNARK/CL-Signature 的用户匿名证书认证方案以应用于物联网场景, 但这一方案计算和通信开销均较大, 也并未提供交易场景下的身份隐藏和恢复机制. 文献 [19] 提出了基于一次性地址的身份隐私保护机制, 但这一方案并未提供交易发送方的身份混淆方案, 也未提供监管信息可靠性证明方法, 无法抵御交易发起方伪造数据进行监管欺诈.

综上所述, 匿名身份标识和交易混淆的结合才能真正有效地解决交易过程中的身份隐私保护问题. 因此本文借鉴门罗币的身份隐私保护方案, 引入监管方的角色, 基于改进的可控匿名身份标识和自主混币技术构建了可监管的身份隐私保护方案. 本文的主要贡献如下.

(1) 厘清了区块链可监管身份隐私保护方案中身份管理的安全目标和安全属性, 给出了安全模型的形式化定义和算法的安全证明.

(2) 将门罗币双密钥对身份标识方法简化为单密钥对身份标识方法, 并设计了相应的一次性交易地址派生算法. 从而简化了客户端密钥管理, 节省了密钥存储空间.

(3) 基于双线性群 (bilinear group) 上安全的 DLIN 加密方案[20]和零知识证明协议[21]生成一次性地址的监管信息, 可供验证方在线实时验证接收方身份密文信息的可靠性, 在保护交易接收方身份隐私的同时又实现了可靠监管. 监管方可通过账本数据离线恢复交易接收方的身份, 达到“可控匿名”的监管目的.

(4) 结合门罗币 CryptoNote[22]和文献 [23] 中的环签名技术, 设计了一种可链接可撤销环签名方案, 从而通过环签名实现了自主交易混淆, 保护了交易发送方的身份隐私. 在线交易时, 验证方利用签名方案的可链接性防止了用户的双花行为, 利用签名方案的不可伪造性确保签名身份的可靠恢复; 离线监管时, 利用签名的可撤销性恢复出交易发送方的签名公钥 (一次性地址), 再通过签名公钥对应的监管信息即可恢复出交易发送方的真实身份.

本文设计的可监管身份隐私保护方案同相关方案的功能比较如表 1 所示.

本文第 1 节介绍了区块链身份隐私保护的基本概念、相关工作和本文的主要贡献. 第 2 节介绍了方案设计和证明相关的预备知识. 第 3 节详细阐述了可监管的交易接收方身份隐私保护方案的算法设计和安全性证明. 第 4 节详细阐述了可监管的交易发送方身份隐私保护方案的算法设计和安全性证明. 第 5 节对算法进行了实现和测试, 并对测试结果进行了分析. 第 6 节总结.

表 1 不同方案的比较

方案	匿名性	交易混淆	可监管性	监管可靠性	不可链接性
文献[4]	是	是	否	否	强
文献[13]	是	是	是	是	弱
文献[15]	是	否	是	否	否
文献[17]	是	否	是	否	否
文献[18]	是	否	否	否	否
文献[19]	是	否	是	否	否
本方案	是	是	是	是	强

2 预备知识

2.1 双线性映射

令G_1和G_2表示阶为素数p的循环加法群, G_T表示阶为p的循环乘法群, 假设G_1, G_2和G_T上的离散对数问题都是难解的, 若G_1, G_2和G_T之间的映射$e:G_1\times G_2\to G_T$满足下列性质, 则此映射是双线性映射.

(1) 双线性: 对于$\forall\ S\in G_1$, $T\in G_2$, $a,b\in Z_p$, 有$e(aS,bT)=e(S,T)^{ab}$.

(2) 非退化性: 存在$S\in G_1$, $T\in G_2$, 使得$e(S,T)\neq 1$.

(3) 可计算性: 对于$\forall\ S\in G_1$, $T\in G_2$, 都存在有效的算法计算$e(S,T)$.

若$G_1=G_2$, 则双线性映射为对称的, 否则, 双线性映射为非对称的. 本文的研究都是基于对称双线性映射, 后续描述为$e:G_1\times G_1\to G_T$.

2.2 通用分叉引理

通用分叉引理[24]是指对于给定的集合H(集合H的基$h\geqslant 2$)以及整数$n\geqslant 1$, 算法D是一个输入为$(y,h_1,\ldots,h_n)$的随机算法, 该算法返回一个二元组(μ,σ). 其中, $h_1,\ldots,h_n\leftarrow H$(对应具体的签名算法, $h_1,\ldots,h_n$是随机预言机的回答), 假设IG是一个随机算法, 被称为"输入生成器", 且$y\leftarrow IG$(对应具体的签名算法, y是公钥); 二元组(μ,σ)中的第 1 个元素μ是$\{0,1,2,\ldots,n\}$中的一个元素, 第 2 个元素σ被称为侧输出. 定义acc为算法D成功的概率, 即在实验: $y\leftarrow IG$; $h_1,\ldots,h_n\leftarrow H$; $(\mu,\sigma)\leftarrow Dy,h_1,\ldots,h_n$中, $acc=P_r[\mu\geqslant 1]$, 则算法D的分叉算法$F_D(y)$描述如下.

(1) 选取运行算法D的随机数ρ;

(2) 随机生成$h_1,\ldots,h_n\leftarrow H$;

(3) 运行$(\mu,\sigma)\leftarrow D(y,h_1,\ldots,h_n;\rho)$;

(4) 如果$\mu=0$, $F_D(y)$返回$0,\perp,\perp$; 否则, 继续执行步骤 (5);

(5) 随机生成$h_1',\ldots,h_n'\leftarrow H$;

(6) 运行$(\mu',\sigma')\leftarrow D\left(y,h_1,\ldots,h_{\mu-1},h_\mu',\ldots,h_n';\rho\right)$;

(7) 如果$\mu=\mu'$且$h_\mu\neq h_\mu'$, 返回$1,\sigma,\sigma'$, 否则, 返回$(0,\perp,\perp)$.

对于算法$F_D(y)$, 定义$frk=P_r[b=1:y\leftarrow IG,(b,\sigma,\sigma')\leftarrow F_D(y)]$, 则有: $frk\geqslant acc(acc/n\ -\ 1/h)$.

2.3 困难问题

(1) DL 问题

有限域上的离散对数 (discrete logarithm, DL) 问题是指已知G_q是一个阶为q的有限循环群, $g\in G_q$是其中的一个生成元, $y\in G_q$是群G_q中的另一个元素, 求解整数$x\in Z_q$, 使得$y=g^x$成立.

(2) DLIN 问题

DLIN (decisional linear, DLIN) 问题是指G_q是一个阶为素数q的有限循环群, $g\in G_q$是G_q的一个生成元, 给定一组元素$\left(g^a,g^b,g^{ac},g^{bd},g^t\right)$, 其中$a,b,c,d\in Z_q^*$是随机选择的, $t\in Z_q$, 判断$t=c+d$是否成立.

双线性群G_1中的 DLIN 问题是指给定双线性群G_1中的元素(U,V,H,aU,bV,cH), 判断$c=a+b$是否成立.

(3) BDDH 问题

BDDH (bilinear decisional Diffie-Hellman) 问题是指给定两个阶为素数q的加法循环群G_1和乘法循环群G_T, 存在双线性映射: $e:G_1\times G_1\to G_T$. P为G_1的随机生成元, 给定一组元素(P,aP,bP,cP)和$t\in G_T$, 其中, $a,b,c\in Z_q$是未知整数, 判断$t=e(P,P)^{abc}$是否成立.

3 可监管的身份隐私保护方案总体架构

通过引言部分的分析可知, 匿名身份标识和交易混淆的结合才能真正有效地解决交易过程中的身份隐私保护. 因此本文借鉴门罗币的身份隐私保护方案, 引入监管方的角色, 改进了身份认证和交易确权流程, 基于优化后的可控匿名的身份标识、一次性地址生成和自主混币算法实现可监管的隐私保护方案. 改进后的身份管理流程如图 1 所示.

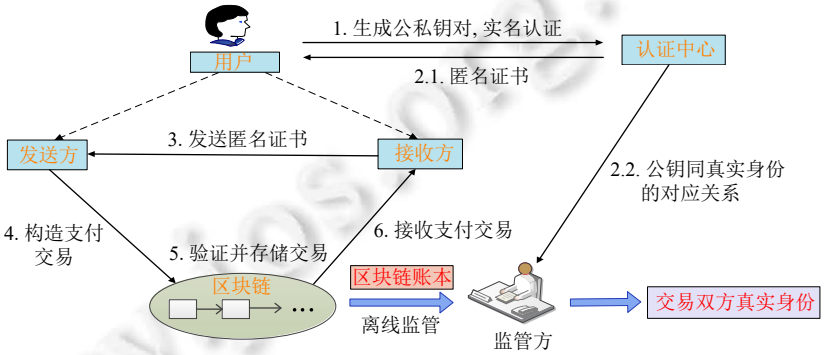


图 1 身份管理流程

具体流程说明如下.

(1) 用户自主生成公私钥对后, 用标识身份的长期公钥向认证中心提出实名认证申请.

(2) 认证中心使用该公钥向用户颁发匿名证书, 同时将公钥同真实身份的对应关系发送给监管方.

(3) 在进行交易时, 交易接收方将匿名证书发送给交易发送方, 发送方使用匿名证书中的长期公钥派生出一次性交易地址作为接收地址并附加密文监管信息写入交易 (详见第 3 节). 这一过程通过使用一次性地址取代固定长期公钥来确保同一交易接收方在不同交易中具有不同的交易地址, 从而避免交易可链接.

(4) 交易发送方在支付时使用可链接可撤销环签名 (详见第 4 节) 完成 UTXO 权属确认, 并将真实交易地址隐藏在环匿名地址集合中实现交易混淆, 使得观察者无法判断真正的交易输入地址, 从而实现身份隐私保护以避免交易追踪.

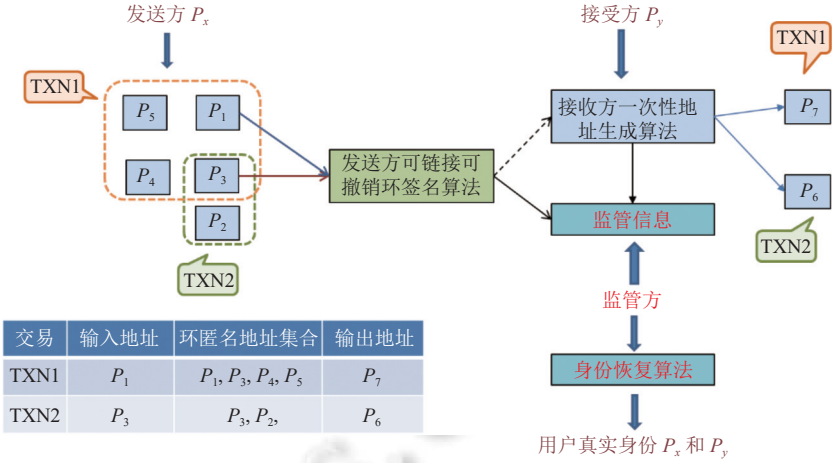
(5) 交易验证方 (如公链中的矿工节点或联盟链中的验证节点) 可通过上述信息验证交易信息的正确性. 如: 因为每个 UTXO 都对应于不同的一次性输出地址且该地址对应的私钥镜像也均不相同, 如果支付 UTXO 进行环签名时这一密钥镜像出现在已使用密钥镜像列表中, 则表示该 UTXO 已被花费, 验证方可拒绝该双花交易 (详见第 4 节). 同时, 交易验证方也可通过上述密态信息使用零知识证明算法验证监管信息的可靠性, 这一身份信息密态验证的方案既保证了交易隐私又实现了可靠监管. 如: 交易验证方可通过一次性地址的零知识证明验证算法 (详见第 3 节) 确保发送方正确使用监管方公钥加密一次性地址对应的接收方长期公钥, 以防止交易地址欺诈.

(6) 通过验证的交易被写入区块链账本. 当有监管需求时, 监管方可利用账本信息恢复出交易双方的真实身份, 实现离线监管.

交易构建过程中的具体流程示例如图 2 所示. 其中, 交易发送方的长期公钥为P_x, 接收方的长期公钥为P_y. 在交易 TXN1 中, 交易发送方的长期公钥P_x对应交易中的一次性地址为P_1, 发送方P_x通过环签名方案将P_1隐藏在

匿名地址集合$\{P_1, P_3, P_4, P_5\}$中, 接收方 P_y 在本次交易中的一次性地址为 P_7; 在交易 TXN2 中, 交易发送方的长期公钥 P_x 对应交易中的一次性地址为 P_3, 发送方通过环签名方案将 P_3 隐藏在匿名地址集合$\{P_3, P_2\}$中, 接收方 P_y 在本次交易中的一次性地址为 P_6.

监管方通过监管信息可恢复出用户的真实身份 P_x 和 P_y. 交易身份恢复的监管流程示例如图 3 所示.



交易	输入地址	环匿名地址集合	输出地址
TXN1	P_1	P_1, P_3, P_4, P_5	P_7
TXN2	P_3	$P_3, P_2,$	P_6

图 2 交易流程

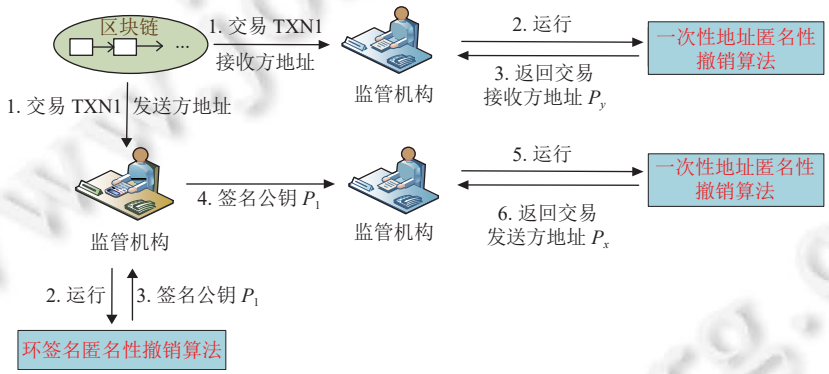


图 3 监管流程

(1) 当有监管需求时, 监管方从区块链账本获得待监管交易 TXN1 的监管信息.

(2) 监管方读取交易接收方地址 P_7, 通过一次性地址的匿名性撤销算法直接恢复出交易接收方的长期公钥 P_y, 从而确认接收方身份.

(3) 监管方可通过可链接可撤销环签名的匿名性撤销算法获得交易发送方地址 P_1, 并搜索账本获得输出为 P_1 的 UTXO, 再通过一次性地址的匿名性撤销算法恢复出长期公钥 P_x 从而确认发送方真实身份.

4 可监管的交易接收方身份隐私保护方案

4.1 相关定义及安全模型

4.1.1 方案定义

方案包括交易发送方、交易接收方和一次性地址的匿名性撤销机构 (监管方) 这 3 个角色. 系统可以提供多个一次性地址的匿名性撤销机构, 交易发送方可以从中选择自己信任的监管方为其提供撤销匿名性参数.

可监管的一次性地址方案 OTA 由 10 个算法构成, 描述如下.

(1) 公共系统参数生成算法 $pp \leftarrow Setup\left(1^{\lambda}\right)$: 输入安全参数 λ, 输出公共系统参数 pp.

(2) 用户密钥生成算法 $(usk,upk) \leftarrow GenUserKey(pp)$: 由用户执行, 输入公共参数 pp, 输出用户的私钥 usk 和公钥 upk, 并使用标识用户真实身份的长期公钥 upk 向认证中心实名注册.

(3) 监管方密钥生成算法 $(rsk,rpk) \leftarrow GenRegulatorKey(pp)$: 输入公共参数 pp, 输出监管方的私钥 rsk 和公钥 rpk.

(4) 一次性地址生成算法 $addr \leftarrow GenOTA(pp,upk)$: 由交易发送方执行, 输入公共参数 pp, 接收方的长期公钥 upk, 输出接收方的一次性地址 $addr$. 该算法用于确保同一用户在不同交易中的接收地址是不同的.

(5) 加密算法 $ct \leftarrow Enc(pp,upk,rpk)$: 由交易发送方执行, 输入公共参数 pp, 交易接收方的长期公钥 upk, 监管方的公钥 rpk, 输出监管密文 ct.

(6) 零知识证明生成算法 $proof \leftarrow GenProof(pp,ct,rpk,addr)$: 由交易发送方执行, 输入公共参数 pp, 密文 ct, 监管方的公钥 rpk, 接收方的一次性地址 $addr$, 输出监管证明参数 $proof$. ct 和 $proof$ 构成了交易身份的监管信息.

(7) 零知识证明验证算法 $0/1 \leftarrow VerProof(pp,addr,ct,proof,rpk)$: 由交易验证方执行, 输入公共参数 pp, 监管方的公钥 rpk, 接收方的一次性地址 $addr$ 以及监管信息 ct 和 $proof$, 输出 0 或 1. 1 表示验证通过, 可进行后续操作; 0 表示验证失败, 拒绝进行下一步操作.

(8) 一次性地址验证算法 $0/1 \leftarrow VerAddr(pp,addr,usk)$: 由私钥为 usk 的用户执行, 输入公共参数 pp, 某笔交易中的一次性地址 $addr$ 和用户私钥 usk, 输出 0 或 1. 1 表示验证通过, 私钥为 usk 的用户是交易的接收方; 0 表示验证失败, 私钥为 usk 的用户不是交易的接收方.

(9) 一次性私钥计算算法 $otsk \leftarrow CalOTPrivKey(pp,addr,R,usk)$: 假设一次性地址 $addr$ 属于私钥为 usk 的用户, 私钥为 usk 的用户执行此算法, 输入 $pp,addr,usk$ 以及此次交易信息中的随机数 R, 输出一次性私钥 $otsk$. 该算法支持用户通过拥有的唯一私钥计算出交易过程中一次性地址对应的一次性私钥, 交易时使用一次性私钥来生成签名完成支付.

(10) 解密算法 (一次性地址匿名性撤销算法) $pt \leftarrow Dec(pp,ct,rsk)$: 由监管方执行, 输入公共参数 pp, 密文 ct 和监管方的私钥 rsk, 输出解密后的明文 pt. pt 即为用户长期公钥 upk, 从而监管方依据注册信息即可恢复出用户的真实身份.

4.1.2 安全模型

可监管的一次性地址方案应满足交易接收方的匿名性、可监管性、监管可靠性和交易接收方一次性私钥的安全性. 在给出上述安全属性的形式化定义前, 我们先通过挑战者 $\mathcal{CH}$ 提供的一系列查询预言机 (Oracle) 来描述敌手 $\mathcal{A}$ 的攻击行为.

$O_{\mathcal{R}}$: 敌手 $\mathcal{A}$ 查询此预言机以获得系统中诚实用户的长期公钥 upk, 并将公钥 upk 添加到诚实用户列表 $\mathcal{L}_{\mathcal{H}}$ 中. $\mathcal{CH}$ 收到查询后, 调用 $(usk,upk) \leftarrow GenUserKey(pp)$ 生成诚实用户的公私钥对 (usk,upk), 执行 $\mathcal{L}_{\mathcal{H}} = \mathcal{L}_{\mathcal{H}} \cup \{upk\}$, 返回公钥 upk 给敌手 $\mathcal{A}$. $O_{\mathcal{R}}$ 预言机用于刻画敌手可获得某些诚实用户的长期公钥 upk.

$O_{\mathcal{W}}$: 敌手 $\mathcal{A}$ 查询此预言机以获得区块链公开账本上一次性地址及其监管信息 $(addr,ct,proof)$. $\mathcal{CH}$ 收到查询后, 访问公开账本获得 $(addr,ct,proof)$ 并返回给敌手 $\mathcal{A}$. $O_{\mathcal{W}}$ 预言机用于刻画敌手可通过观察公开账本获得诚实用户交易过程中生成的一次性地址 $addr$、监管密文 ct 和监管证明信息 $proof$.

$O_{\mathcal{C}}$: 敌手 $\mathcal{A}$ 使用特定公钥 $upk \in \mathcal{L}_{\mathcal{H}}$ 查询此预言机, 预言机返回对应的私钥 usk, 并将此用户移入恶意用户列表 $\mathcal{L}_{\mathcal{M}}$. $\mathcal{CH}$ 收到查询后, 执行 $\mathcal{L}_{\mathcal{H}} = \mathcal{L}_{\mathcal{H}} - \{upk\}$, $\mathcal{L}_{\mathcal{M}} = \mathcal{L}_{\mathcal{M}} \cup \{upk\}$. 返回私钥 usk 给敌手 $\mathcal{A}$. $O_{\mathcal{C}}$ 预言机用于刻画敌手可控制系统中某一用户的公私钥对 (upk,usk).

$O_{\mathcal{T}}$: 敌手 $\mathcal{A}$ 使用特定公钥 $upk \in \mathcal{L}_{\mathcal{M}}$ 查询此预言机, 预言机生成以 upk 为接收地址的合法交易. $\mathcal{CH}$ 收到查询后, 运行 $GenOTA(pp,upk)$ 和 $GenProof(pp,ct,rpk,addr)$ 算法生成并返回该交易中 upk 对应的 $(addr,ct,proof)$. $O_{\mathcal{T}}$ 预言机用于刻画敌手可作为交易接收方通过系统生成合法交易并获得对应的一次性地址 $addr$、密文 ct 和监管证明信息 $proof$.

$O_{\mathcal{S}}$: 敌手 $\mathcal{A}$ 使用特定公钥 $upk' \in \mathcal{L}_{\mathcal{M}}$ 和 $upk \in \mathcal{L}_{\mathcal{H}}$ 查询此预言机, 预言机生成以 upk 为接收地址的合法交易.

$\mathcal{CH}$ 收到查询后, 以 upk' 作为交易发送方地址, 运行 $GenOTA(pp,upk)$ 和 $GenProof(pp,ct,rpk,addr)$ 算法生成并返回该交易中 upk 对应的 $(addr,ct,proof)$ 和交易随机数 r_{tx}. O_S 预言机用于刻画敌手可作为交易发送方通过系统生成合法交易并获得对应的一次性地址 $addr$、密文 ct、监管证明信息 $proof$ 和交易随机数 r_{tx}.

(1) 交易接收方的匿名性

交易接收方的匿名性是指除了交易发送方、交易接收方和监管方以外, 任何用户都不能获取交易接收方的区块链地址. 我们通过敌手 $\mathcal{A}$ 和挑战者 $\mathcal{CH}$ 间进行的如下实验 $Expt_{\mathrm{OTA}}^{\mathrm{ano}}(\lambda)$ 来定义交易接收方的匿名性.

① 生成阶段: $\mathcal{CH}$ 执行系统公共参数生成算法 $pp \leftarrow Setup\left(1^{\lambda}\right)$ 和监管方密钥生成算法 $(rsk,rpk) \leftarrow GenRegulatorKey(pp)$, 将参数 pp 和监管方公钥 rpk 发送给 $\mathcal{A}$, 保留监管方私钥 rsk;

② 询问阶段 1: $\mathcal{A}$ 适应性地询问 O_W、O_C、$O_{\mathcal{T}}$ 预言机;

③ 挑战阶段: $\mathcal{A}$ 询问 O_R 预言机获得两个长期公钥地址 $upk_0, upk_1 \notin \mathcal{L}_{\mathcal{M}}$, 发送给 $\mathcal{CH}$; $\mathcal{CH}$ 选择 $b \xleftarrow{R} \{0,1\}$, 生成 upk_b 的一次性地址 $addr_b \leftarrow GenOTA(pp,upk_b)$、监管信息 $ct_b \leftarrow Enc(pp,upk_b,rpk)$ 和 $proof_b \leftarrow GenProof(pp,ct_b,rpk,addr_b)$, 发送给 $\mathcal{A}$;

④ 询问阶段 2: $\mathcal{A}$ 适应性地询问 O_W、O_C、$O_{\mathcal{T}}$ 预言机, 但不能使用 upk_0 和 upk_1 查询 O_C 预言机;

⑤ 猜测阶段: $\mathcal{A}$ 输出对 b 的猜测 b', 若 $b=b'$, 则实验输出结果为 1; 否则, 实验输出 0.

用 $Adv_{\mathrm{OTA}}^{\mathrm{ano}}(\lambda) = |\Pr[Expt_{\mathcal{A},\mathrm{OTA}}^{\mathrm{ano}}(\lambda)=1] - 1/2| = |\Pr[b=b'] - 1/2|$ 表示敌手 $\mathcal{A}$ 打破可监管一次性地址方案匿名性的优势.

定义 1. 对于任意的概率多项式时间 PPT 敌手 $\mathcal{A}$, 如果敌手 $\mathcal{A}$ 打破方案匿名性的优势 $Adv_{\mathrm{OTA}}^{\mathrm{ano}}(\lambda)$ 是可忽略的, 即 $Adv_{\mathrm{OTA}}^{\mathrm{ano}}(\lambda) \leqslant negl(\lambda)$, 则方案满足匿名性.

(2) 可监管性

可监管性是指对于诚实操作的交易发送方, 监管方可以恢复出一次性地址对应的接收方长期公钥.

(3) 监管可靠性

监管可靠性是指包括监管方在内的区块链系统中的所有用户都能检测到伪造的监管数据, 伪造信息无法通过监管验证. 我们通过敌手 $\mathcal{A}$ 和挑战者 $\mathcal{CH}$ 间进行的如下实验 $Expt_{\mathrm{OTA}}^{\mathrm{sup}}(\lambda)$ 来定义监管可靠性.

① 生成阶段: $\mathcal{CH}$ 执行系统公共参数生成算法 $pp \leftarrow Setup\left(1^{\lambda}\right)$ 和监管方密钥生成算法 $(rsk,rpk) \leftarrow GenRegulatorKey(pp)$, 将参数 pp 和监管方公钥 rpk 发送给 $\mathcal{A}$, 保留监管方私钥 rsk;

② 询问阶段: $\mathcal{A}$ 适应性地询问 O_W、O_C、$O_{\mathcal{T}}$ 预言机;

③ 伪造阶段: $\mathcal{A}$ 通过询问 O 和 O_C 预言机选择 upk^* 并生成对应的伪造信息: 一次性地址 $addr^*$ 以及监管信息 ct^* 和 $proof^*$, 其中 $Dec(pp,ct^*,rsk) \neq upk^*$. $\mathcal{A}$ 将伪造信息发送给挑战者;

④ 判定阶段: $\mathcal{CH}$ 运行零知识证明验证算法, 并将验证结果 $VerProof(pp,addr^*,ct^*,proof^*,rpk)$ 输出.

用 $Adv_{\mathrm{OTA}}^{\mathrm{ano}}(\lambda) = \Pr[Expt_{\mathrm{OTA}}^{\mathrm{sup}}(\lambda)=1]$ 表示敌手 $\mathcal{A}$ 打破方案监管可靠性的优势.

定义 2. 对于任意的概率多项式时间 PPT 敌手 $\mathcal{A}$, 如果敌手 $\mathcal{A}$ 打破方案监管可靠性的优势 $Adv_{\mathrm{OTA}}^{\mathrm{sup}}(\lambda)$ 是可忽略的, 即 $Adv_{\mathrm{OTA}}^{\mathrm{sup}}(\lambda) \leqslant negl(\lambda)$, 则方案满足监管可靠性.

(4) 一次性私钥的安全性

交易接收方一次性私钥的安全性是指只有真正的交易接收方可以计算出一次性地址对应的一次性私钥, 监管方和其他用户均不能求解. 我们通过敌手 $\mathcal{A}$ 和挑战者 $\mathcal{CH}$ 间进行的如下实验 $Expt_{\mathrm{OTA}}^{\mathrm{pri}}(\lambda)$ 来定义一次性私钥的安全性.

① 生成阶段: $\mathcal{CH}$ 执行系统公共参数生成算法 $pp \leftarrow Setup\left(1^{\lambda}\right)$ 和监管方密钥生成算法 $rsk,rpk \leftarrow GenRegulatorKey(pp)$, 将参数 pp 和监管方公钥 rpk 发送给 $\mathcal{A}$, 保留监管方私钥 rsk;

② 询问阶段 1: $\mathcal{A}$ 适应性地询问 O_W、O_C、$O_{\mathcal{T}}$ 预言机;

③ 挑战阶段: $\mathcal{A}$ 询问 O_R 预言机获得一个长期公钥地址 $upk \notin \mathcal{L}_{\mathcal{M}}$, 并通过 O_S 预言机查询获得 upk 的一次性地址 $addr \leftarrow GenOTA(pp,upk)$、监管信息 $ct \leftarrow Enc(pp,upk,rpk)$、交易随机数 r_{tx} 和 $proof \leftarrow GenProof(pp,ct,rpk,addr)$;

④ 询问阶段 2: $\mathcal{A}$ 适应性地询问 O_W、O_C、O_T 预言机, 但不能使用 upk 查询 O_C 预言机;

⑤ 猜测阶段: $\mathcal{A}$ 输出对一次性地址 $addr$ 私钥的猜测 a', 若私钥 $a=a'$, 则输出 1; 否则, 输出 0.

用 $Adv_{\mathrm{OTA}}^{\mathrm{pri}}(\lambda)=\Pr[Expt_{\mathrm{OTA}}^{\mathrm{pri}}(\lambda)=1]$ 表示敌手 $\mathcal{A}$ 打破方案一次性私钥安全性的优势.

定义 3. 对于任意的概率多项式时间 PPT 敌手 $\mathcal{A}$, 如果敌手 $\mathcal{A}$ 打破方案一次性私钥的安全性的优势 $Adv_{\mathrm{OTA}}^{\mathrm{pri}}(\lambda)$ 是可忽略的, 即 $Adv_{\mathrm{OTA}}^{\mathrm{pri}}(\lambda)\leqslant negl(\lambda)$, 则方案满足一次性私钥的安全性.

4.2 方案描述

(1) 系统初始化 *Setup*

输入安全参数 λ, 选择两个相同素数阶 q 的循环群 G_1 和 G_T, G 为 G_1 的生成元, e 表示双线性映射 $G_1\times G_1\to G_T$, 假设 G_1 和 G_T 中的离散对数问题是困难的, $\mathcal{H}_s$ 为 $G_1\to Z_q$ 的密码学哈希函数, 输出公共参数 $pp=(\lambda,q,G_1,G_T,G,\mathcal{H}_s)$.

(2) 用户密钥生成算法 *GenUserKey*

用户 i 的私钥仅由随机数 $d_i\in Z_q$ 构成, 公钥为 $P_i=d_iG$. 用户使用 P_i 向认证中心实名注册.

(3) 监管方密钥生成算法 *GenRegulatorKey*

监管方的私钥为 $rsk=(d_x,d_y)$, $d_x,d_y\in Z_q$, 公钥为 $rpk=(P_x,P_y,P_T)$, 其中 $P_T=d_xP_x=d_yP_y$. 监管方的密钥可通过如下方式构造:

① 监管方选择随机数 $d_x\in Z_q$, 计算 $P_y=d_xG$;

② 监管方选择随机数 $d_y\in Z_q$, 计算 $P_x=d_yG$;

③ 计算 $P_T=d_xP_x$ (因为 $d_xP_x=d_xd_yG=d_yd_xG=d_yP_y$);

④ 最终得到监管方的私钥 (d_x,d_y), 公钥 (P_x,P_y,P_T).

(4) 一次性地址生成算法 *GenOTA*

假设交易发送方为 Alice, 交易接收方为 Bob. Bob 的私钥为 $b\in Z_q$, 公钥为 $B=bG$. Alice 获取 Bob 的区块链地址 B 之后执行如下操作:

① 选择随机数 $r_{tx}\in Z_q$, 计算 $R_{tx}=r_{tx}G$;

② 计算 $t=\mathcal{H}_s(r_{tx}B)$, 为接收方 Bob 计算一次性地址 $P=tG+B$;

③ 将 R_{tx} 的值写入到交易内容中, 同时将 P 作为交易的目的地址写入交易.

(5) 加密算法 *Enc*

为了实现监管, 发送方使用监管方的公钥对接收方的区块链地址进行加密生成监管密文, Alice 执行操作如下:

① 选择随机数 $k_1\in Z_q$, 计算 $C_1=k_1P_x$;

② 选择随机数 $k_2\in Z_q$, 计算 $C_2=k_2P_y$;

③ 计算 $k=k_1+k_2$, 计算 $C_3=kP_T+B$;

④ 发送方 Alice 将撤销匿名性参数 $C_{\mathrm{revoke}}=(C_1,C_2,C_3)$ 写入交易内容中.

(6) 零知识证明生成算法 *GenProof*

为了防止交易发送方为逃避监管而进行欺骗, 要求交易发送方生成零知识证明信息 π_{cp}, 以表明发送方加密时使用的是监管方公钥, 且被加密内容为用于计算一次性地址的交易接收方长期公钥. 给定 $G,P,P_x,P_y,P_T,C_1,C_2,C_3\in G_1$, 交易发送方要向验证方证明:

$$PoK\left\{\left(t,k_1,k_2,k,B,B'\right):\ C_1=k_1P_x\wedge C_2=k_2P_y\wedge C_3=kP_T+B'\wedge k=k_1+k_2\wedge P=tG+B\wedge B=B'\ \right\}.$$

证明者 Alice 基于 Schnorr 协议[25]和 Fait-Shamir 启发式[21]构造非交互式零知识证明.

① 选择随机数 $r_{k_1}\in Z_q$, 计算 $Q_1=r_{k_1}P_x$;

② 选择随机数 $r_{k_2}\in Z_q$, 计算 $Q_2=r_{k_2}P_y$;

③ 计算 $r_k=r_{k_1}+r_{k_2}$;

④ 选择随机数 $r_t \in Z_q$, 计算 $Q_3 = r_k P_T - r_t G$;

⑤ 计算 $h = \mathcal{H}_s\left(G \| P \| P_x \left\| P_y \right\| P_T \| C_1 \| C_2 \| C_3 \| Q_1 \| Q_2 \| Q_3\right)$;

⑥ 计算 $w_{k_1} = r_{k_1} + h \cdot k_1$, $w_{k_2} = r_{k_2} + h \cdot k_2$ 和 $w_t = r_t + h \cdot t$;

⑦ Alice 将证明参数 $\pi_{cp} = (h, w_{k_1}, w_{k_2}, w_t)$ 写入到交易内容中.

(7) 零知识证明验证算法 *VerProof*

验证者获取到公开证明参数 $\pi_{cp} = (h, w_{k_1}, w_{k_2}, w_t)$ 后, 执行如下验证操作.

① 计算 $Q_1' = w_{k_1} P_x - hC_1$, $Q_2' = w_{k_2} P_y - hC_2$;

② 计算 $w_k = w_{k_1} + w_{k_2}$;

③ 计算 $Q_3' = w_k P_T - w_t G + h(P - C_3)$;

④ 计算 $h' = \mathcal{H}_s\left(G \| P \| P_x \left\| P_y \right\| P_T \| C_1 \| C_2 \| C_3 \| Q_1' \| Q_2' \| Q_3'\right)$;

⑤ 验证等式 $h' = h$ 是否成立, 若成立, 则验证通过, 否则, 验证失败.

若验证通过, 则说明交易发送方没有欺骗行为, 可以允许此笔交易的生成; 若验证失败, 则认为交易不合法, 拒绝记录这笔交易.

(8) 一次性地址验证算法 *VerAddr*

对于区块链上每一笔合法交易, 区块链中的用户都可以使用自己的密钥检验交易是否属于自己, 例如 Bob 计算 $P' = \mathcal{H}_s(bR_{tx})G + B$, 如果 $P = P'$, 则证明这笔交易属于 Bob.

(9) 一次性私钥计算算法 *CalOTPrivKey*

Bob 找到属于自己的交易后计算对应的一次性私钥 $d = \mathcal{H}_s(bR_{tx}) + b$.

(10) 解密算法 (一次性地址匿名性撤销算法) *Dec*

监管方用自己的私钥 $\left(d_x, d_y\right)$ 以及交易中的 $C_{\text{revoke}} = (C_1, C_2, C_3)$ 计算: $P_{\text{Recipient}} = C_3 - \left(d_x C_1 + d_y C_2\right)$, $P_{\text{Recipient}}$ 即为接收方的区块链地址.

4.3 安全性证明

(1) 交易接收方的匿名性

定理 1. 如果 DLIN 加密方案是 IND-CPA 安全的, 零知识证明 π_{cp} 满足零知识性, 则方案在随机预言机模型下满足交易接收方的匿名性.

证明: 下面用一系列游戏证明定理 1, $\Pr[Win_i]$ 表示敌手在 Game i 中获胜的优势, $negl(\lambda)$ 表示可忽略概率, λ 为安全参数.

● Game 0. 对应于定义 1 中的实验 $Expt_{\text{OTA}}^{\text{ano}}(\lambda)$, 敌手 $\mathcal{A}$ 的挑战明文为 (upk_0, upk_1), 获得的挑战密文为 $(addr_b, ct_b, proof_b)$, $\mathcal{A}$ 输出对 b 的猜测 b', 则:

$$\Pr[Win_0] = \left|\Pr[b = b'] - \frac{1}{2}\right|.$$

● Game 1. Game 1 与 Game 0 类似, 不同之处是零知识证明 π_{cp} 是由模拟器 S 模拟生成的 (如可使用文献 [26] 中的 SimSign 来代替 sign), 即 $\mathcal{CH}$ 运行 $proof_b \leftarrow S(pp, ct_b, rpk, addr_b)$. 如果敌手在 Game 0 和 Game 1 获胜的概率存在差异, 则可以利用敌手 $\mathcal{A}$ 构造一个算法来打破零知识证明的零知识性, 所以 $\Pr[Win_1]$ 满足不等式:

$$|\Pr[Win_1] - \Pr[Win_0]| \leqslant negl(\lambda).$$

● Game 2. Game 2 与 Game 1 类似, 不同之处是使用随机生成的 G_1 中的元素作为 $GenOTA(pp, pk_b)$ 的输出 $addr_b$. 我们将安全的密码学哈希函数 $\mathcal{H}_s$ 抽象为随机预言机, 若随机选取 r_{tx}, 则 $\mathcal{H}_s(r_{tx}B)$ 的输出为值域上均匀分布的随机值, 因此敌手 $\mathcal{A}$ 无法区分计算的一次性地址 $P = \mathcal{H}_s(r_{tx}B)G + B$ 和随机生成的模拟输出 $addr_b$. 所以 $\Pr[Win_2]$ 满足不等式:

$$|\Pr[Win_2] - \Pr[Win_1]| \leqslant negl(\lambda).$$

引理 1. 如果 DLIN 加密方案是 IND-CPA 安全的, 那么对于所有的 PPT 敌手 $\mathcal{A}$, $\Pr[Win_2] \leqslant negl(\lambda)$ 成立.

证明: 如果 PPT 敌手 $\mathcal{A}$ 能够以不可忽略的优势赢得 Game 2, 则可构建算法 $\mathcal{B}$ 能够以同样的优势打破 DLIN 加密方案的 IND-CPA 安全. 算法 $\mathcal{B}$ 模拟运行 Game 2 如下.

① 设置: $\mathcal{B}$ 运行 $pp \leftarrow Setup\left(1^{\lambda}\right)$、$(usk,upk) \leftarrow GenUserKey(pp)$ 和 $(rsk,rpk) \leftarrow GenRegulatorKey(pp)$, 并将 pp、rpk 返回给敌手 $\mathcal{A}$;

② 挑战前查询: 敌手 $\mathcal{A}$ 可查询如下预言机:

$O_{\mathcal{R}}$: $\mathcal{B}$ 收到后, 调用 $(usk,upk) \leftarrow GenUserKey(pp)$ 生成诚实用户的公私钥对 (usk,upk), 执行 $\mathcal{L}_{\mathcal{H}} = \mathcal{L}_{\mathcal{H}} \cup \{upk\}$.

$O_{\mathcal{W}}$: $\mathcal{B}$ 收到查询后, 访问公开账本获得 $(addr,ct,proof)$ 并返回给敌手 $\mathcal{A}$.

$O_{\mathcal{C}}$: $\mathcal{B}$ 收到查询后, 执行 $\mathcal{L}_{\mathcal{H}} = \mathcal{L}_{\mathcal{H}} - \{upk\}$, $\mathcal{L}_{\mathcal{M}} = \mathcal{L}_{\mathcal{M}} \cup \{upk\}$. 返回私钥 usk 给敌手 $\mathcal{A}$.

$O_{\mathcal{T}}$: $\mathcal{B}$ 收到查询后, 运行 $GenOTA(pp,pk)$ 和 $GenProof(pp,ct,rpk,addr)$ 算法生成并返回该交易中 upk 对应的 $(addr,ct,proof)$.

③ 挑战: 敌手 $\mathcal{A}$ 选择 $upk_0, upk_1 \notin \mathcal{L}_{\mathcal{M}}$ 作为挑战值发送给 $\mathcal{B}$. $\mathcal{B}$ 作为敌手运行 DLIN 加密方案的 IND-CPA 实验 $Expt_{\mathrm{DLIN}}^{\mathrm{CPA}}$, 将 upk_0, upk_1 转发给 $Expt_{\mathrm{DLIN}}^{\mathrm{CPA}}$ 并获得返回值 ct_b. $\mathcal{B}$ 运行 $proof_b \leftarrow GenProof(pp,ct,rpk,addr)$, 将 $(addr_b, ct_b, proof_b)$ 返回给敌手 $\mathcal{A}$.

④ 挑战后查询: 敌手 $\mathcal{A}$ 可继续查询预言机 O、$O_{\mathcal{W}}$、$O_{\mathcal{C}}$、$O_{\mathcal{T}}$, 但敌手 $\mathcal{A}$ 不能使用 pk_0 和 pk_1 查询 $O_{\mathcal{C}}$ 预言机.

⑤ 猜测: $\mathcal{A}$ 输出 b 的猜测值 b'. $\mathcal{B}$ 将猜测值 b' 转发给实验 $Expt_{\mathrm{DLIN}}^{\mathrm{CPA}}$.

由此可见, $\mathcal{A}$ 在 Game 2 中的观察同 $\mathcal{A}$ 在 $\mathcal{B}$ 中的观察完全等价分布, 所以 $\mathcal{A}$ 在 Game 2 中成功的概率与 $\mathcal{A}$ 在 $\mathcal{B}$ 中成功的概率相同. $\mathcal{A}$、$\mathcal{B}$ 均为 PPT 算法, 故敌手 $\mathcal{A}$ 赢得 Game 2 的优势和 $\mathcal{B}$ 赢得 $Expt_{\mathrm{DLIN}}^{\mathrm{CPA}}$ 的优势相同. 因为 DLIN 加密方案是 IND-CPA 安全的, 所以 $\mathcal{B}$ 赢得 $Expt_{\mathrm{DLIN}}^{\mathrm{CPA}}$ 的优势是可忽略的. 由此可以得出 $Pr[Win_2] \leqslant negl(\lambda)$.

综上分析可以得出, 如果 G_1 中的离散对数问题是困难的, DLIN 加密方案是 IND-CPA 安全的, 且零知识证明 π_{cp} 满足零知识性, 则敌手 $\mathcal{A}$ 在 Game 0 获胜的概率与在 Game 2 获胜的概率之间存在的差异是可忽略的, 所以 $Adv_{\mathrm{OTA}}^{\mathrm{ano}}(\lambda) \leqslant negl(\lambda)$, 故方案满足匿名性.

(2) 可监管性

定理 2. 若交易发送方是诚实的, 则监管方能够正确恢复出一次性地址对应的交易接收方的长期公钥.

证明: 因为 $C_1 = k_1 P_x$, $C_2 = k_2 P_y$, $C_3 = kP_T + B$, $k = k_1 + k_2$, $P_T = d_x P_x = d_y P_y$, 所以有:

$$\begin{aligned} P_{\mathrm{Recipient}} &= C_3 - \left(d_x C_1 + d_y C_2\right) = kP_T + B - \left(d_x k_1 P_x + d_y k_2 P_y\right) = (k_1 + k_2) P_T + B - (k_1 P_T + k_2 P_T) \\ &= (k_1 + k_2) P_T + B - (k_1 + k_2) P_T = B. \end{aligned}$$

由上述推导可知, 如果发送方诚实按照 Enc 算法对交易接收方的区块链地址进行加密, 则监管方可以通过 Dec 算法正确恢复出交易接收方的长期公钥. 因此方案满足可监管性.

(3) 监管可靠性

定理 3. 如果生成监管信息的零知识证明满足可靠性, 则方案满足监管可靠性.

证明: 方案通过非交互式零知识证明算法生成监管信息, 因此可将监管可靠性直接规约到零知识证明方案的可靠性. 针对具体的方案, 交易发送方构造虚假数据逃避监管的方式可分为如下 3 种.

① 交易发送方在加密时使用的加密公钥不是监管方的公钥, 而是 G_1 中的任意一组元素 $\left(P'_x, P'_y, P'_T\right)$, 即计算 $C'_1 = k_1 P'_x$, $C'_2 = k_2 P'_y$, $C'_3 = kP'_T + B$, 则验证者在验证时有:

$$Q'_1 = w_{k_1} P_x - hC'_1 = r_{k_1} P_x + hk_1 P_x - hk_1 P'_x.$$

因为 $P'_x \neq P_x$, 所以 $Q'_1 \neq r_{k_1} P_x$ 且 $Q'_1 \neq r_{k_1} P'_x$, 无论发送方在 $GenProof$ 算法中是设置 $Q_1 = r_{k_1} P_x$ 还是 $Q_1 = r_{k_1} P'_x$, 总有 $Q'_1 \neq Q_1$, 同理有, $Q'_2 \neq Q_2$, $Q'_3 \neq Q_3$, 由于哈希函数的抗碰撞性, 进一步可得 $h' \neq h$.

② 交易发送方在生成密文 C_3 时, 没有使用 $k = k_1 + k_2$, 而是选择随机数 $k' \neq k_1 + k_2$, 计算 $C'_3 = k'P_T + B$, 则验证者在验证时计算:

$$Q_3' = w_k P_T - w_t G + h\left(P - C_3'\right) = (r_{k_1} + hk_1 + r_{k_2} + hk_2)P_T - (r_t + ht)G + h(tG - k'P_T)$$
$$= r_{k_1}P_T + r_{k_2}P_T + hk_1P_T + hk_2P_T - r_tG - hk'P_T = (r_{k_1} + r_{k_2})P_T + h(k_1 + k_2)P_T - r_tG - hk'P_T$$
$$= r_kP_T - r_tG + h(k_1 + k_2)P_T - hk'P_T.$$

因为 $k' \neq k_1 + k_2$, 所以 $Q_3' \neq r_kP_T - r_tG$, 即 $Q_3' \neq Q_3$, 所以 $h' \neq h$.

③ 交易发送方所加密的明文不是交易接收方的区块链地址 B, 而是用随机元素 B' 代替, 即计算 $C_3' = kP_T + B'$, 则验证者在验证时计算 $P - C_3' = tG + B - kP_T - B'$, 进而得到:

$$Q_3' = w_kP_T - w_tG + h(P - C_3') = (r_{k_1} + r_{k_2})P_T - r_tG + h(B - B') = r_kP_T - r_tG + h(B - B').$$

因为 $B' \neq B$, 所以 $Q_3' \neq r_kP_T - r_tG$, 即 $Q_3' \neq Q_3$, 所以 $h' \neq h$.

综上分析, 交易发送方欺骗区块链系统用户及监管方接受零知识证明的概率是可忽略的, 虚假数据无法通过 $VerProof$ 算法的验证, 即 $Adv_{\mathrm{OTA}}^{\mathrm{sup}}(\lambda) \leqslant negl(\lambda)$, 故方案满足监管可靠性.

(4) 一次性私钥的安全性

定理 4. 如果 G_1 中的离散对数 DL 问题是困难的, 则监管方和其他用户均不能求解交易接收方一次性地址对应的一次性私钥.

证明: 假设敌手 $\mathcal{A}$ 能以不可忽略的优势赢得实验 $Expt_{\mathrm{OTA}}^{\mathrm{pri}}(\lambda)$, 则可以构建一个算法 $\mathcal{B}$ 并解决 DL 问题. 算法 $\mathcal{B}$ 模拟运行 $Expt_{\mathrm{OTA}}^{\mathrm{pri}}(\lambda)$ 如下.

① 设置: 给定算法 $\mathcal{B}$ 一个 G_1 中 DL 问题的挑战实例: 已知 $D = dG$, 求 d. 首先, $\mathcal{B}$ 输入挑战实例, 运行 $pp \leftarrow Setup\left(1^{\lambda}\right)$、$(usk, upk) \leftarrow GenUserKey(pp)$ 和 $(rsk, rpk) \leftarrow GenRegulatorKey(pp)$, 并将 pp、rpk 返回给敌手 $\mathcal{A}$;

② 询问阶段 1: $\mathcal{A}$ 适应性地询问 O_W、O_C、O_T 预言机;

③ 挑战: 敌手 $\mathcal{A}$ 查询 O_R 时, $\mathcal{B}$ 控制随机预言机 O_R 将长期公钥地址设置为 $upk = D$ 并返回给 $\mathcal{A}$. $\mathcal{A}$ 通过 O_S 预言机查询获得 upk 的一次性地址 $addr \leftarrow GenOTA(pp, upk)$、监管信息 $ct \leftarrow Enc(pp, upk, rpk)$、交易随机数 r_{tx} 和 $proof \leftarrow GenProof(pp, ct, rpk, addr)$;

④ 询问阶段 2: $\mathcal{A}$ 适应性地询问 O_W、O_C、O_T 预言机, 但不能使用 upk 查询 O_C 预言机;

⑤ 猜测: 敌手 $\mathcal{A}$ 输出 $addr$ 私钥的猜测值 a'.

假设 $\mathcal{A}$ 能够以不可忽略的优势 $\epsilon_{\mathcal{A}}$ 赢得实验 $Expt_{\mathrm{OTA}}^{\mathrm{pri}}(\lambda)$, 则算法 $\mathcal{B}$ 就可通过如下方法计算问题实例中的 d: 因为 $a' = \mathcal{H}_s(dR_{tx}) + d = \mathcal{H}_s(r_{tx}D) + d$, 所以 $d = a' - \mathcal{H}_s(r_{tx}D)$. 即算法 $\mathcal{B}$ 能够以同样的优势 $\epsilon_{\mathcal{A}}$ 解决 $D = dG$ 的 DL 问题. 因此可得 $Adv_{\mathrm{OTA}}^{\mathrm{pri}}(\lambda) \leqslant negl(\lambda)$, 故方案满足一次性私钥的安全性.

5 可监管的交易发送方身份隐私保护方案

5.1 相关定义及安全模型

5.1.1 方案定义

方案包括签名者、验证者和匿名性撤销机构 3 种角色, 分别对应着区块链交易中的交易发送方、区块链系统中的验证节点和监管方. 系统同样可以提供多个环签名的匿名性撤销机构, 由交易发送方自主选择自己信任的监管方.

可链接可撤销环签名方案 $LRRS$ 由 6 个算法构成, 描述如下.

(1) 系统建立 $pp \leftarrow LRRS\_Setup\left(1^{\lambda}\right)$: 输入安全参数 λ, 输出公共参数 pp.

(2) 密钥生成算法 $(sk, pk) \leftarrow LRRS\_GenKey(pp)$: 输入公共参数 pp, 输出私钥 sk 和公钥 pk.

(3) 环签名生成算法 $\sigma \leftarrow LRRS\_RSig\left(pp, S, sk_{\pi}, \widetilde{P_u}, m\right)$: 由签名者执行, 假设签名者的签名私钥为 sk_{π}, 其签名公钥 pk_{π} 属于集合 S, 输入公共参数 pp, 环成员的公钥集合 S, 签名私钥 sk_{π}, 匿名性撤销机构的公钥 $\widetilde{P_u}$ 和消息 m, 算法输出环签名 σ, 其中包含用于判断可链接性的密钥镜像 I 以及用于匿名性撤销的监管信息 E.

(4) 环签名验证算法 $0/1 \leftarrow LRRS\_RVer\left(pp, S, \sigma, \widetilde{P_u}, m\right)$: 由验证者执行, 输入公共参数 pp, 环成员的公钥集合 S,

环签名σ, 匿名性撤销机构的公钥$\widetilde{P_u}$和消息m, 输出 0 或 1. 1 表示验证通过, 可以进行下一步操作, 并将环签名σ中的密钥镜像I加入列表$\mathcal{L}$; 0 表示验证失败, 拒绝进行后续操作.

(5) 环签名的链接性判断算法$0/1 \leftarrow LRRS_Link(\sigma,\mathcal{L})$: 由验证者执行, 输入环签名$\sigma$和列表$\mathcal{L}$, 将$\sigma$中的密钥镜像$I$与列表$\mathcal{L}$中的记录做对比, 输出 0 或 1. 0 表示$I$没有在列表$\mathcal{L}$中出现过, 环签名是“独立”的, 没有使用相同签名私钥生成的环签名存在; 1 表示I在列表$\mathcal{L}$中出现过, 环签名是“链接”的, 相同的签名私钥被使用了不止一次.

(6) 匿名性撤销算法$pk_\pi \leftarrow LRRS_Revoke\left(pp,S,\sigma,\widetilde{sk_r}\right)$: 由匿名性撤销机构 (监管方) 执行, 输入公共参数pp, 环成员的公钥集合S, 一个有效的环签名σ, 匿名性撤销机构的私钥$\widetilde{sk_r}$, 输出签名公钥pk_π.

5.1.2 安全模型

一个安全的可链接可撤销环签名方案至少应该满足可链接性、匿名性、不可伪造性和匿名可撤销性. CryptoNote[22]还要求用于支付模型的方案应满足不可诽谤性, 即防止攻击者在没有此用户私钥的情况下, 生成其私钥对应的密钥镜像.

在给出上述安全属性的形式化定义前, 我们先通过挑战者$\mathcal{CH}$提供的一系列查询预言机 (oracle) 来描述敌手$\mathcal{A}$的攻击行为. 令$\mathcal{U}=\{pk_0,pk_1,\ldots,pk_{n-1}\}$是一个公钥集合, 每对公私钥都是通过运行$LRRS_GenKey\left(1^\lambda\right)$算法生成的. 安全模型定义为敌手$\mathcal{A}$提供以下预言机.

$O_{\mathcal{R}}$: 注册预言机, 敌手$\mathcal{A}$查询此预言机注册一个新的新用户, 该预言机返回一个有效的新注册用户公钥pk_i. $O_{\mathcal{R}}$预言机用于刻画敌手可获得区块链系统中某些诚实用户的公钥pk_i.

O_S: 签名预言机, 敌手$\mathcal{A}$使用环成员公钥集合$S'\subseteq\mathcal{U}$、匿名性撤销机构的公钥$\widetilde{P_u}$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集合$\mathcal{S}_S$中. O_S预言机用于刻画敌手可通过观察公开账本获得诚实用户交易过程中生成的环签名σ'.

O_C: 俘获成员预言机, 敌手$\mathcal{A}$使用$\mathcal{U}$中某个公钥pk_i询问此预言机, 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$\mathcal{S}_C$中. O_C预言机用于刻画敌手可控制系统中某一用户的公私钥对(pk_i,sk_i).

(1) 可链接性

可链接性是指同一个私钥生成的环签名是可以被链接的, 此性质防止了交易发送方的双花行为. 我们通过敌手$\mathcal{A}$和挑战者$\mathcal{CH}$间进行的如下实验$Expt_{\mathrm{LRRS}}^{\mathrm{link}}(\lambda)$来定义可链接性.

① 生成阶段: $\mathcal{CH}$执行公共参数生成算法$pp \leftarrow LRRS_Setup\left(1^\lambda\right)$, 将生成的公共参数$pp$发送给$\mathcal{A}$;

② 询问阶段: $\mathcal{A}$询问$O_{\mathcal{R}}$预言机获得集合$\mathcal{U}=\{pk_0,pk_1,\ldots,pk_{n-1}\}$;

③ 伪造阶段: $\mathcal{A}$选择$n+1$个消息$(M_0,\ldots,M_n)$以任意公钥$pk_i\in\mathcal{U}$构造$n+1$个消息签名对$\{(M_0,\sigma_0),\ldots,(M_n,\sigma_n)\}$, 并发送给$\mathcal{CH}$;

④ 判定阶段: $\mathcal{CH}$接收签名并判断, 若$\forall i\in[0,n]$, $LRRS_RVer\left(pp,S,\sigma_i,\widetilde{P_u},M_i\right)=1$且$LRRS_LINK(\sigma_i,\mathcal{L})=0$成立, 则实验输出 1; 否则, 输出 0.

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

定义 4. 对于任意的概率多项式时间 PPT 敌手$\mathcal{A}$, 如果敌手打破方案可链接性的优势$Adv_{\mathrm{LRRS}}^{\mathrm{link}}(\lambda)$是可忽略的, 即$Adv_{\mathrm{LRRS}}^{\mathrm{link}}(\lambda)\leqslant negl(\lambda)$, 则 LRRS 方案满足可链接性.

(2) 不可诽谤性

不可诽谤性是指恶意用户无法生成另一个用户的私钥对应的密钥镜像, 此性质防止了恶意攻击者对用户财产的“窃取”行为. 我们通过敌手$\mathcal{A}$和挑战者$\mathcal{CH}$间进行的如下实验$Expt_{\mathrm{LRRS}}^{\mathrm{excu}}(\lambda)$来定义不可诽谤性:

① 生成阶段: $\mathcal{CH}$执行公共参数生成算法$pp \leftarrow LRRS_Setup\left(1^\lambda\right)$, 将生成的公共参数$pp$发送给$\mathcal{A}$;

② 询问阶段: $\mathcal{A}$询问$O_{\mathcal{R}}$预言机获得集合$\mathcal{U}=\{pk_0,pk_1,...,pk_{n-1}\}$. $\mathcal{A}$询问O_C预言机获得公钥pk_i对应的私钥$sk_i(i=1,\ldots,j-1,j+1,\ldots,n-1)$. $\mathcal{A}$继续适应性地询问O_S;

③ 伪造阶段: $\mathcal{A}$选择消息M, 公钥集合$S\subseteq\mathcal{U}\left(pk_j\in S\right)$和$\widetilde{P_u}$, 构造签名$\sigma^* \leftarrow LRRS_RSig(pp,S,sk_i,T,m)$并发送给$\mathcal{CH}$;

④ 判定阶段: $\mathcal{CH}$ 运行环签名验证算法, 若 $LRRS\_RVer\left(pp,S,\sigma^*,\widetilde{P_u},m\right)=1$ 且 $I^*=I_j$, 则实验输出 1; 否则, 输出 0.

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

定义 5. 对于任意的概率多项式时间 PPT 敌手 $\mathcal{A}$, 如果敌手打破方案不可诽谤性的优势 $Adv_{\mathrm{LRRS}}^{\mathrm{excu}}(\lambda)$ 是可忽略的, 即 $Adv_{\mathrm{LRRS}}^{\mathrm{excu}}(\lambda)\leqslant negl(\lambda)$, 则 LRRS 方案满足不可诽谤性.

(3) 不可伪造性

不可伪造性是指如果环 S 中任何人的私钥都没有泄露, 则敌手不能伪造出关于环 S 的有效环签名. 我们通过敌手 $\mathcal{A}$和 挑战者 $\mathcal{CH}$ 间进行的如下实验 $Expt_{\mathrm{LRRS}}^{\mathrm{unfo}}(\lambda)$ 来定义不可伪造性.

① 生成阶段: $\mathcal{CH}$ 执行公共参数生成算法 $pp\leftarrow LRRS\_Setup\left(1^{\lambda}\right)$, 将生成的公共参数 pp 发送给 $\mathcal{A}$;

② 询问阶段: $\mathcal{A}$ 询问 $O_{\mathcal{R}}$ 预言机获得集合 $\mathcal{U}=\{pk_0,pk_1,\ldots,pk_{n-1}\}$, 并适应性地询问 $O_{\mathcal{S}}$ 预言机获得签名;

③ 伪造阶段: $\mathcal{A}$ 选择消息 m, 公钥集合 $S\subseteq\mathcal{U}$ 和 $\widetilde{P_u}$, 构造签名 $\sigma^*\leftarrow LRRS\_RSig\left(pp,S,sk_i,\widetilde{P_u},m\right)$ 并发送给 $\mathcal{CH}$, 要求 $\mathcal{A}$ 没有使用消息 m、集合 S 和 $\widetilde{P_u}$ 查询过 $O_{\mathcal{S}}$ 预言机;

④ 判定阶段: $\mathcal{CH}$ 运行环签名验证算法, 若 $LRRS\_RVer\left(pp,S,\sigma^*,\widetilde{P_u},m\right)=1$, 则输出 1; 否则, 输出 0.

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

定义 6. 对于任意的概率多项式时间 PPT 敌手 $\mathcal{A}$, 如果敌手打破方案不可伪造性的优势 $Adv_{\mathrm{LRRS}}^{\mathrm{unfo}}(\lambda)$ 是可忽略的, 即 $Adv_{\mathrm{LRRS}}^{\mathrm{unfo}}(\lambda)\leqslant negl(\lambda)$, 则 LRRS 方案满足不可伪造性.

文献 [27] 指出, 只要方案满足可链接性和不可诽谤性, 则方案满足不可伪造性, 具体将在后文中证明.

(4) 匿名性

匿名性是指给定一个环签名, 假设环成员的个数为 n, 如果任何环成员的私钥都没有被泄露, 则敌手确定出签名者的概率最大为 $1/n$. 我们通过敌手 $\mathcal{A}$和 挑战者 $\mathcal{CH}$ 间进行的如下实验 $Expt_{\mathrm{LRRS}}^{\mathrm{anon}}(\lambda)$ 来定义匿名性.

① 生成阶段: $\mathcal{CH}$ 执行公共参数生成算法 $pp\leftarrow LRRS\_Setup\left(1^{\lambda}\right)$, 将生成的公共参数 pp 发送给 $\mathcal{A}$;

② 询问阶段: $\mathcal{A}$ 询问 $O_{\mathcal{R}}$ 预言机获得集合 $\mathcal{U}=\{pk_0,pk_1,\ldots,pk_{n-1}\}$, 并使用 $\{pk_2,\ldots,pk_{n-1}\}$ 中的公钥适应性地询问 $O_{\mathcal{S}}$ 预言机获得公开签名;

③ 挑战阶段: $\mathcal{A}$ 选择消息 m, 公钥 pk_0,pk_1 发送给 $\mathcal{CH}$; $\mathcal{CH}$ 选择 $b\xleftarrow{R}\{0,1\}$, 使用 sk_b 和公钥集合 $S\subseteq\mathcal{U}\wedge pk_b\in S$ 和 $\widetilde{P_u}$ 生成环签名 $\sigma\leftarrow LRRS\_RSig\left(pp,S,sk_b,\widetilde{P_u},m\right)$ 并发送给 $\mathcal{A}$;

④ 猜测阶段: $\mathcal{A}$ 输出对 b 的猜测 b', 若 $b=b'$, 则实验输出结果为 1; 否则, 实验输出 0.

用 $Adv_{\mathrm{LRRS}}^{\mathrm{anon}}(\lambda)=|\Pr[Expt_{\mathrm{LRRS}}^{\mathrm{anon}}(\lambda)=1]-1/2|$ 表示敌手 $\mathcal{A}$ 打破方案匿名性的优势.

定义 7. 对于任意的概率多项式时间 PPT 敌手 $\mathcal{A}$, 如果敌手打破方案匿名性的优势 $Adv_{\mathrm{LRRS}}^{\mathrm{anon}}(\lambda)$ 是可忽略的, 即 $Adv_{\mathrm{LRRS}}^{\mathrm{anon}}(\lambda)\leqslant negl(\lambda)$, 则 LRRS 方案满足匿名性.

(5) 匿名可撤销性

匿名可撤销性是指监管方可通过有效签名恢复出签名者的真实公钥. 我们通过敌手 $\mathcal{A}$和 挑战者 $\mathcal{CH}$ 间进行的如下实验 $Expt_{\mathrm{LRRS}}^{\mathrm{revoke}}(\lambda)$ 来定义匿名可撤销性.

① 生成阶段: $\mathcal{CH}$ 执行公共参数生成算法 $pp\leftarrow LRRS\_Setup\left(1^{\lambda}\right)$, 将生成的公共参数 pp 发送给 $\mathcal{A}$;

② 询问阶段: $\mathcal{A}$ 询问 $O_{\mathcal{R}}$ 预言机获得集合 $\mathcal{U}=\{pk_0,pk_1,\ldots,pk_{n-1}\}$, $\mathcal{A}$ 选择公钥集合 $S\subseteq\mathcal{U}$ 和 $\widetilde{P_u}$, 适应性地询问 $O_{\mathcal{S}}$;

③ 伪造阶段: 所有询问结束后, $\mathcal{A}$ 输出由私钥 sk_π, 以及消息 m 构造的 $\sigma^*\leftarrow LRRS\_RSig\left(pp,S,sk_\pi,\widetilde{P_u},m\right)$ 且 $LRRS\_RVer\left(pp,S,\sigma^*,\widetilde{P_u},m\right)=1$; $\mathcal{A}$ 将 $\left(S,\sigma^*,\widetilde{P_u},m\right)$ 发送给 $\mathcal{CH}$. 要求 $\mathcal{A}$ 没有使用消息 m、集合 S 和 T 查询过 $O_{\mathcal{S}}$ 预言机;

④ 判定阶段: 使用监管方私钥 $\widetilde{sk_u}$, $\mathcal{CH}$ 运行 $LRRS\_Revoke\left(pp,S,\sigma^*,\widetilde{sk_u}\right)$ 算法, 得到输出 $pk_{\mathcal{R}}$. 若 $pk_{\mathcal{R}}\neq pk_\pi$, 则实验输出 1; 否则, 输出 0.

用 $Adv_{\mathrm{LRRS}}^{\mathrm{revoke}}(\lambda)=\Pr[Expt_{\mathrm{LRRS}}^{\mathrm{revoke}}(\lambda)=1]$ 表示敌手 $\mathcal{A}$ 打破方案匿名可撤销性的优势.

定义 8. 对于任意的概率多项式时间 PPT 敌手 $\mathcal{A}$, 如果敌手打破方案匿名可撤销性的优势 $Adv_{\mathrm{LRRS}}^{\mathrm{revoke}}(\lambda)$ 是可忽略的, 即 $Adv_{\mathrm{LRRS}}^{\mathrm{revoke}}(\lambda) \leqslant negl(\lambda)$, 则 LRRS 方案满足匿名可撤销性.

5.2 方案描述

(1) 系统建立 *LRRS_Setup*

与第 4.2 节 *Setup* 算法相同, 公共参数 $pp=(\lambda,q,G_1,G_T,G,\mathcal{H}_s)$.

(2) 密钥生成算法 *LRRS_GenKey*

① 用户 i 的私钥用 $d_i \in Z_q$ 表示, 公钥为 $P_i = d_i G$;

② 环签名的匿名性撤销机构 (监管方) u 选择 $\widetilde{d_u} \in Z_q$ 作为私钥, 并计算 $\widetilde{P_u} = \widetilde{d_u} G$ 作为公钥. 设定固定的撤销参数 $R_{\mathrm{revoke}} = rG$.

(3) 环签名生成算法 *LRRS_RSig*

假设待签名消息为 M, 签名者生成环签名的步骤如下.

① 随机选择 $n-1$ 个用户的公钥, 与自己的签名公钥进行混合得到集合;

② $S=\{P_1,P_2,\ldots,P_n\}$, 假设签名公钥为 P_π, 私钥为 d_π;

③ 计算 $E = I = e\left(R_{\mathrm{revoke}},\widetilde{P_u}\right)^{d_\pi}$;

④ 在 Z_q 中随机选择 $\{q_i | i=1,\ldots,n,i\neq\pi\}$ 和 $\{w_i | i=1,\ldots,n,i\neq\pi\}$, 计算 $L_i = q_i G + w_i P_i$ ($i\neq\pi$) 和 $R_i = e\left(R_{\mathrm{revoke}},\widetilde{P_u}\right)^{q_i} I^{w_i}$ $(i\neq\pi)$;

⑤ 选择随机数 $k\in Z_q$, 计算 $L_\pi = kG$ 和 $R_\pi = e\left(R_{\mathrm{revoke}},\widetilde{P_u}\right)^{k}$;

⑥ 计算 $c = \mathcal{H}_s\left(M\|L_1,\ldots,L_n\|R_1,\ldots,R_n\|S\|\widetilde{P_u}\|R_{\mathrm{revoke}}\|E\right)$;

⑦ 计算 $w_\pi = c - \sum w_i \bmod q (i=1,\ldots,n,i\neq\pi)$, $q_\pi = k - w_\pi d_\pi \bmod q$, 生成签名 $\sigma=(I,w_1,\ldots,w_n,q_1,\ldots,q_n)$.

(4) 环签名验证算法 *LRRS_RVer*

验证者获取环签名 σ, 消息 M 以及公开的集合 S 和 $\widetilde{P_u}$ 后, 对 σ 进行验证, 验证过程如下.

① 对于 $i=1,\ldots,n$, 计算 $L_i' = q_i G + w_i P_i$, $R_i' = e\left(R_{\mathrm{revoke}},\widetilde{P_u}\right)^{q_i} I^{w_i}$;

② 计算 $c_{ver} = \sum_{i=1}^{n} w_i \bmod q$;

③ 计算 $c' = \mathcal{H}_s\left(M\|L_1',\ldots,L_n'\|R_1',\ldots,R_n'\|S\|\widetilde{P_u}\|R_{\mathrm{revoke}}\|E\right)$;

④ 若 $c_{ver} = c'$ 成立, 则算法输出 1, 验证通过; 否则, 输出 0, 验证失败.

(5) 环签名的链接性判断 *LRRS_Link*

验证者获取环签名 σ 中的密钥镜像 I, 看其是否在历史密钥镜像列表 $\mathcal{L}$ 中出现过, 若出现过, 则 *LRRS_Link* 算法输出 1, 表示用户进行了双花行为, 要拒绝此签名; 否则, *LRRS_Link* 算法输出 0, 表示环签名是“独立”的, 可以接受此签名. 并将 I 加入历史密钥镜像列表 $\mathcal{L}$ 中.

(6) 匿名性撤销算法 *LRRS_Revoke*

私钥为 $\widetilde{d_u}$ 的监管方想要恢复出签名公钥, 则依次代入集合 S 中的公钥, 对于 $i=1,\ldots,n$, 判断 $E = e(P_i,R_{\mathrm{revoke}})^{\widetilde{d_u}}$ 是否成立, 输出使等式成立的公钥 P_π.

5.3 安全性证明

5.3.1 可链接性

定理 5. 在随机预言机模型下, 方案满足可链接性.

证明: 假设敌手 $\mathcal{A}$ 至多可以进行 q_H 次 $\mathcal{H}_s$ 次询问. 如果对于集合 S (S 中的元素个数为 n), 敌手可以生成 $n+1$ 个密钥镜像为 I_i 的有效的环签名 σ_i, 其中, 对于任意的 $i,j\in[1,n]$, $I_i\neq I_j$, 因为集合 S 的公钥个数为 n, 则对于每一个 i, 至少有一个 $I_i \neq e\left(R_{\mathrm{revoke}},\widetilde{P_u}\right)^{d_i}$.

环签名 $\sigma=(I,w_1,\ldots,w_n,q_1,\ldots,q_n)$ 能通过 *LRRS_RVer* 的验证说明有:

$$L_i' = q_i G + w_i P_i$$
$$R_i' = e\left(R_{\text{revoke}}, \widetilde{P_u}\right)^{q_i} I^{w_i}$$

令 $F = e\left(R_{\text{revoke}}, \widetilde{P_u}\right)$, 则有:

$$\log_G L_i' = q_i + w_i d_i,$$
$$\log_F R_i' = q_i + w_i \log_F I.$$

文献 [27] 中指出 $\nexists i : d_i = \log_F I$ 表示所有的 w_i 都是独立确定的, $\mathcal{H}_s$ 是一个随机预言机, 对于任意给定的 $\left(M, L_1', \ldots, L_n', R_1', \ldots, R_n', S, \widetilde{P_u}, R_{\text{revoke}}, E\right)$, $\sum_{i=1}^{n} w_i (\text{mod } q) = H_s(M \| L_1', \ldots, L_n' \| R_1', \ldots, R_n' \| S \| \tilde{P}_u \| R_{\text{revoke}} \| \text{E})$, 因此方案满足可链接性.

5.3.2 不可诽谤性

定理 6. 在随机预言机模型下, 如果 G_1 中的离散对数 DL 问题是困难的, 则方案满足不可诽谤性.

证明: 假设敌手 $\mathcal{A}$ 已知集合 S (S 中的元素个数为 n) 和 S 中 $n-1$ 个公钥对应的私钥 $d_i (i = 1, \ldots, j-1, j+1, \ldots, n)$ 和监管方公钥 $\widetilde{P_u}$, 其能以不可忽略的概率 ϵ 伪造生成代表 S 的有效环签名 σ, σ 中的密钥镜像 $I = e\left(R_{\text{revoke}}, \widetilde{P_u}\right)^{d_j}$, 则可以构建一个算法 $\mathcal{B}$ 解决 G_1 中的 DL 问题.

(1) 设置: 给定 $\mathcal{B}$ 一个 DL 问题的挑战实例: 已知 $Q, G \in G_1$, $Q = xG$, $\mathcal{B}$ 的目标是求解 x. $\mathcal{B}$ 选取成员 $j \in n$, 并设定 $P_j = Q$ 作为其公钥. $\mathcal{B}$ 输入挑战实例, 对于 $i \neq j$, $\mathcal{B}$ 随机选取 $d_i \in Z_q$, 并设置 $P_i = d_i G$. $\mathcal{B}$ 提供给敌手 $\mathcal{A}$ 参数 $pp = (\lambda, q, G_1, G_T, G, \mathcal{H}_s)$.

(2) 查询: $\mathcal{B}$ 控制随机预言机 $\mathcal{H}_s$ 和 O_C、O_S 预言机, 并假设 $\mathcal{A}$ 至多可以进行 q_H 次 $\mathcal{H}_s$ 次询问, 以及 q_{sig} 次 O_S 询问. 算法 $\mathcal{B}$ 和敌手 $\mathcal{A}$ 之间完成如下交互.

- $\mathcal{H}_s$ 询问: B 维护一个初始为空的列表 $LIST$, $LIST$ 列表的结构为 $\{\delta', \tau'\}$. 对于询问 $\delta = (M' \| L_1, \ldots, L_n \| R_1, \ldots, R_n \| S \| \widetilde{P_u} \| R_{\text{revoke}} \| E)$, B 查询其之前是否被询问过, 若之前被询问过, 则直接找到记录中对应的 τ, 并返回 τ. 若之前没有询问过, 则 B 选择 $\tau \in Z_q$, 在 $LIST$ 中添加 $\{\delta, \tau\}$, 并返回 τ.
- O_S 询问: 对于 $S' = \{P_1, P_2, \ldots, P_n\}$, $\widetilde{P_u}$, 以及消息 M', 如果询问 $\left(P_i, S', \widetilde{P_u}, M'\right)$, 且 $i \neq j$, 则 B 使用 d_i 生成环签名 $\sigma' \leftarrow LRRS\_RSig\left(pp, S', d_i \widetilde{P_u}, M'\right)$, 将 σ' 返回. 如果 $i = j$, 则 B 在无需 P_j 私钥的情况下模拟生成环签名, 操作如下.

① 计算 $E = I = e\left(P_j, \widetilde{P_u}\right)^r$;

② 对于 $\eta = 1, 2, \ldots, n$, 随机选取 $w_\eta, q_\eta \in Z_q$, 计算 $L_\eta = q_\eta G + w_\eta P_\eta$ 和 $R_\eta = e\left(R_{\text{revoke}}, \widetilde{P_u}\right)^{q_\eta} I^{w_\eta}$;

③ 计算 $c = \sum_{\eta=1}^{n} w_\eta \bmod q$;

④ 随机选取 $s \in Z_q$;

⑤ 设置 $\mathcal{H}_s\left(M' \| L_1, \ldots, L_n \| R_1, \ldots, R_n \| S' \| \widetilde{P_u} \| R_{\text{revoke}} \| E\right) = c$, 并将 $\{\delta, \tau\} = \{M' \| L_1, \ldots, L_n \| R_1, \ldots, R_n \| S' \| \widetilde{P_u} \| R_{\text{revoke}} \| E, c\}$ 添加到表 $LIST$ 中, 若 $\mathcal{H}_s\left(M' \| L_1, \ldots, L_n \| R_1, \ldots, R_n \| S' \| \widetilde{P_u} \| R_{\text{revoke}} \| E\right)$ 已经在表 $LIST$ 中记录着另一个不同的值, 则模拟错误, B 终止模拟实验. B 模拟成功的概率为 $\left(1 - \left(q_{sig} + q_H\right)/q\right)$.

⑥ B 返回 $\sigma' = (I, w_1, \ldots, w_n, q_1, \ldots, q_n)$ 作为应答.

(3) 伪造: 敌手 $\mathcal{A}$ 以 ϵ 的概率输出其所选择的 $\left(S, \widetilde{P_u}, M\right)$ 的伪造签名 $\sigma = (I, w_1, \ldots, w_n, q_1, \ldots, q_n)$. 如果伪造的签名有效, 则 $P_j \in S$, 且 $I = e\left(R_{\text{revoke}}, \widetilde{P_u}\right)^{d_j}$.

算法 $\mathcal{B}$ 解决 G_1 中的离散对数问题如下: 运行分叉引理算法 $F_D(Q)$ 两次, 以 frk 的概率得到两个有效环签名: $\sigma_0 = (I, w_1, \ldots, w_n, q_1, \ldots, q_n)$ 和 $\sigma_1 = \left(I, w_1', \ldots, w_n', q_1', \ldots, q_n'\right)$. $\mathcal{B}$ 检查 $w_j' = w_j$ 是否成立, 若成立, 则终止算法; 否则, 算法 $\mathcal{B}$ 即可输出 $x = d_j = (q_j - q_j')/(w_j' - w_j) \bmod q$, 这是因为在两个环签名中, 有 $L_j = q_j G + w_j P_j = q_j' G + w_j' P_j$ 且 $P_j = Q$, 则 $x = d_j = \log_G P_j = (q_j - q_j')/(w_j' - w_j) \bmod q$.

由分叉引理可知 $\varepsilon_{\mathcal{B}} \geqslant frk \geqslant acc^2/(q_H + q_S) - acc/2^q \geqslant \varepsilon^2/(q_H + q_S) - 1/2^q$, 因此, 若敌手 $\mathcal{A}$ 能够以不可忽略的概率 ϵ 伪造出有效的密钥镜像为 $I = e\left(R_{\text{revoke}}, \widetilde{P_u}\right)^{d_j}$ 的环签名, 则存在算法 $\mathcal{B}$ 能够以不可忽略的概率 $\epsilon_{\mathcal{B}}$ 解决 G_1 中的离

散对数问题, 这与假设相矛盾. 因此, 本方案满足不可诽谤性.

5.3.3 不可伪造性

文献 [27] 指出不可伪造性实际上就是可链接性和不可诽谤性的结合.

定理 7. 如果可链接可撤销环签名方案满足可链接性和不可诽谤性, 则方案也满足不可伪造性.

证明: 假设存在敌手 $\mathcal{A}$ 对于给定的集合 S (S 中的公钥个数为 n) 和消息 M 伪造了一个环签名 $\sigma_0=(I_0,\dots)$. 考虑由所有集合 S 中的诚实签名者生成的相同消息 M 的有效环签名: $\sigma_1,\sigma_2,\dots,\sigma_n$, 则伪造签名 σ_0 存在下面两种情况.

① $I_0\in\{I_i\}(i=1,2,\dots,n)$, 这与不可诽谤性矛盾.

② $I_0\notin\{I_i\}(i=1,2,\dots,n)$, 这与可链接性矛盾.

综上分析, 只要方案满足可链接性和不可诽谤性, 那么方案也满足不可伪造性.

5.3.4 匿名性

定理 8. 在随机预言机模型下, 如果 BDDH 问题是困难的, 则方案满足匿名性.

证明: 假设敌手 $\mathcal{A}$ 能以不可忽略的优势赢得实验 $Expt_{\mathrm{LRRS}}^{\mathrm{ano}}(\lambda)$, 则可以构建一个算法 $\mathcal{B}$ 解决 BDDH 问题. 算法 $\mathcal{B}$ 模拟运行 $Expt_{\mathrm{LRRS}}^{\mathrm{ano}}(\lambda)$ 如下:

(1) 设置: 给定算法 $\mathcal{B}$ 一个 BDDH 问题的挑战实例: 已知 $(G,P_{\pi0}=aG,\widetilde{P_u}=bG,R_{\mathrm{revoke}}=cG)$, 首先, $\mathcal{B}$ 输入挑战实例, 随机选取为 $z\in G_T$, 并设定 $P_{\pi1}=zG$. 提供给敌手 $\mathcal{A}$ 参数 $pp=(\lambda,q,G_1,G_T,G,\mathcal{H}_s)$.

(2) 查询: $\mathcal{B}$ 控制随机预言机 $\mathcal{H}_s$ 和 O_C 、 O_S 预言机, 并假设 $\mathcal{A}$ 至多可以进行 q_H 次 $\mathcal{H}_s$ 次询问, 以及 q_{sig} 次 O_S 询问. 具体查询过程同定理 6 中的证明.

(3) 挑战:

① 敌手 $\mathcal{A}$ 任意选择消息 M^*, 公钥集合 $P_{\pi0}$ 和 $P_{\pi1}$, 发送给算法 $\mathcal{B}$;

② $\mathcal{B}$ 选择 $b\overset{R}{\leftarrow}\{0,1\}$, 构造相应的环签名 (过程参见定理 6 证明);

③ $b=0$ 时, 签名公钥为 $P_{\pi0}$, 密钥镜像 $I=e\left(R_{\mathrm{revoke}},\widetilde{P_u}\right)^a=e(cG,bG)^a=e(G,G)^{abc}$;

④ $b=1$ 时, 签名公钥为 $P_{\pi1}$, 密钥镜像 $I=e\left(R_{\mathrm{revoke}},\widetilde{P_u}\right)^z=e(cG,bG)^z=e(G,G)^{zbc}$.

⑤ 猜测: 敌手 $\mathcal{A}$ 输出 b 的猜测值 b'. 假设 $\mathcal{A}$ 能够以不可忽略的优势 $\epsilon_{\mathcal{A}}$ 赢得实验, 则算法 $\mathcal{B}$ 就可利用密钥镜像 I 以同样的优势解决 $e(G,G)^{abc}$ 的 BDDH 判定问题. 由此可得 $Adv_{\mathrm{LRRS}}^{\mathrm{anon}}(\lambda)\leqslant negl(\lambda)$, 方案满足匿名性.

5.3.5 匿名可撤销性

定理 9. 如果 G_1 中的离散对数问题是困难的, 签名满足不可伪造性, 则方案满足签名者匿名可撤销性.

证明: 在实验 $Expt_{\mathrm{LRRS}}^{\mathrm{rev}}(\lambda)$ 中, 敌手 $\mathcal{A}$ 获取了某一个环成员的私钥. 如果 PPT 敌手 $\mathcal{A}$ 能够以不可忽略的优势赢得 $Expt_{\mathrm{LRRS}}^{\mathrm{rev}}(\lambda)$, 则可构建算法 $\mathcal{B}$ 能够以不可忽略的概率解决 G_1 中的离散对数问题. 算法 $\mathcal{B}$ 模拟运行 $Expt_{\mathrm{LRRS}}^{\mathrm{rev}}(\lambda)$ 如下.

(1) 设置: 给定算法 $\mathcal{B}$ 一个 G_1 中的离散对数问题实例: 已知 $(P=P_i=d_iG=aG)$, 其中 (P_i,d_i) 为第 i 个环成员的密钥对. 运行 $pp\leftarrow LRRS\_Setup\left(1^\lambda\right)$, 并将 pp 返回给敌手 $\mathcal{A}$;

(2) 查询: 敌手 $\mathcal{A}$ 可查询如下预言机:

$O_{\mathcal{R}}$: 收到查询后, 调用 $LRRS\_GenKey(pp)$ 生成诚实用户的公私钥对 sk,pk, 构建环成员公钥集合 $\mathcal{U}=(pk_0,pk_1,\dots,pk_{n-1})$;

O_S: $\mathcal{B}$ 收到查询后, 调用 $C\mathcal{H}$ 的签名预言机返回合法环签名 (m_i,σ_i).

(3) 伪造: 敌手 $\mathcal{A}$ 随机选择第 i 个环成员并伪造其签名消息 $\left(m_i,\sigma_i^*\right)$ 发送给 $\mathcal{B}$. 敌手成功选择第 i 个环成员的概率为 $1/n$.

假设敌手成功赢得实验 $Expt_{\mathrm{LRRS}}^{\mathrm{rev}}$, 因为签名满足不可伪造性, 则生成的 σ 是有效的, $\mathcal{B}$ 运行匿名性撤销算法 $P_i\leftarrow LRRS\_Revoke(pp,S,\sigma_j^*,\widetilde{P_u},\widetilde{d_u})$; 因为 σ_j^* 为合法签名, 由环签名生成过程中的 $PoK\{d_i:I=e(R_{\mathrm{revoke}},\widetilde{P_u})^{d_i}\}(M)$ 的可靠性说明 $\mathcal{A}$ 知道私钥 d_i 使得 $E_i=e(R_{\mathrm{revoke}},\tilde{P}_u)^{d_i}$ 成立. 假设 $\mathcal{A}$ 能够以不可忽略的优势 $\epsilon_{\mathcal{A}}$ 赢得实验, 则算法 $\mathcal{B}$ 能

够以 $\epsilon_{\mathcal{A}}/n$ 的概率求得 P 的离散对数 $(a=d_i)$. 这与假设相矛盾, 因此可得 $Adv_{\mathrm{LRRS}}^{\mathrm{revoke}}(\lambda)\leqslant negl(\lambda)$, 方案满足匿名可撤销性.

上述推导过程证明了对于恶意的签名者, 其无法生成一个有效的环签名, 使监管方恢复出的签名公钥是另一个用户的. 而对于环成员中的第 π 个诚实签名者, 监管方使用私钥 $\widetilde{d_u}$ 可以正确恢复出签名公钥:

$$E_\pi=e\left(R_{\mathrm{revoke}},\widetilde{P_u}\right)^{d_\pi}=e\left(P_\pi,\widetilde{P_u}\right)^{r_{\mathrm{revoke}}}=e(P_\pi,R_{\mathrm{revoke}})^{\widetilde{d_u}}.$$

综上分析, 定理 9 成立.

6 算法实现与测试

本节分别对本文设计的可监管一次性地址方案和可监管环签名方案进行了算法实现和性能测试. 测试程序使用 go 语言编写, 并调用 go 语言版本的 PBC 库[28]来实现双线性映射的运算. 实验环境配置如表 2 所示.

表 2 实验环境

名称	配置
处理器	Intel(R) Core(TM) i7-10710U 1.10 GHz
内存	1 GB
操作系统	Ubuntu 16.04
go	Version 1.6.2

在双线性映射的参数选取上, PBC 库中的 G_1, G_2 和 G_T 群的阶均为 r,r 的位长设置为 256 比特, PBC 库将配对分为 A、B、C、D、E、F、G 这 7 种类型, 本实验采用类型 A. 类型 A 的配对构造在素数域 Z_q $(q\equiv 3 \bmod 4)$ 中的椭圆曲线 $y^2=x^3+x$ 上, q 的位长设置为 512 比特, G_1 和 G_T 都是 $E(Z_q)$ 上的点群.

6.1 可监管一次性地址方案测试

对第 3.2 节中给出的可监管一次性地址方案中的算法进行了代码实现并测试了各算法执行 100 次后所得的平均时间, 测试数据如图 4 所示.

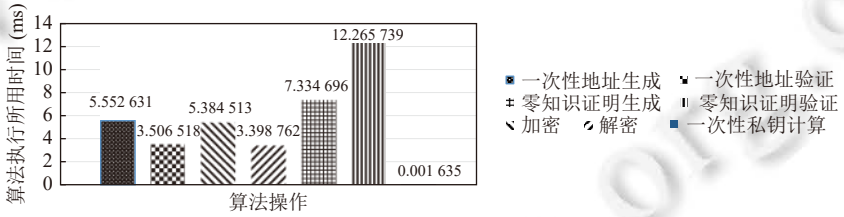


图 4 算法执行时间

实验结果表明, 除一次性私钥的计算, 其他算法执行一次的时间均为毫秒级别, 一次性私钥计算的时间更是微秒级别的. 这是因为用户执行一次性地址的验证操作时计算的 $\mathcal{H}_s(bR_{tx})$ 可以直接用于计算 $d=\mathcal{H}_s(bR_{tx})+b$, 而不需要重复计算, 所以, 一次性私钥计算的时间仅为一次数加操作的时间.

与 CryptoNote 中的一次性地址方案相比, 可监管方案要求发送方额外执行加密和零知识证明的生成操作, 验证者也增加了零知识证明的验证操作, 对于某笔交易, 这些操作次数与发送方选择的监管机构个数和接收方个数呈线性相关; 一次性地址的生成和验证以及某监管机构的解密操作次数只与接收方个数有关; 表 3 给出了在不同的操作次数下, 各算法执行所需的总时间.

在实际交易系统中, 发送方选择的监管机构个数通常为 1. 此时, 对于某笔交易, 发送方所需执行的一次性地址生成、加密和零知识证明生成操作也只与接收方的个数有关. 由表 3 可知, 当某笔交易的接收方个数为 20 时, 执行这些操作的总时间约 0.366 s, 对零知识证明和一次性地址验证的总时间约为 0.315 s, 解密总时间为 0.068 s.

表 3 不同操作次数下各算法所需时间 (ms)

操作次数	一次性地址生成	加密	零知识证明生成	零知识证明验证	一次性地址验证	解密
1	5.572 716	5.383 413	7.322 584	12.263 873	3.496 717	3.407 651
5	25.863 580	26.917 065	36.612 920	61.319 365	17.483 585	17.038 255
10	55.727 160	53.834 130	73.225 840	122.638 730	34.967 170	34.076 510
15	83.590 740	80.751 195	109.838 760	183.958 095	52.450 755	51.114 765
20	111.454 320	107.668 260	146.451 680	245.277 460	69.934 340	68.153 020

6.2 可监管环签名方案测试

依据第 4.2 节中给出的可监管环签名方案中的算法定义, 分别进行代码实现并进行了测试. 在实际交易中, 为了保证交易效率, 生成环签名的环成员个数 n 通常不超过 10 个, 且发送方也会尽可能少的设置监管方的数量 l, 故本节分别测试了 n 从 2 到 10 的情况下, l 取值 1 到 5 时的环签名生成和验证所用时间. 图 5 是 LRRS 环签名的生成时间结果, 图 6 是 LRRS 环签名的验证时间结果. 由实验结果可知, 环签名的生成和验证时间随 n 或 l 的增加呈线性增长, 当 n 为 10, 且 l 为 5 时, 环签名生成时间约 0.101 s, 验证时间约 0.102 s.

监管机构撤销环签名匿名性的时间与真正签名者的索引 *index* 有关. 假设环集合中第 1 个成员的索引为 1, 图 7 展示了签名者索引 *index* 从 1 到 10 的情况下, 撤销环签名匿名性所需的时间. 撤销匿名性时间随 *index* 增长而增长, 当 *index* 为 10 时, 所需时间约 13.96 ms.

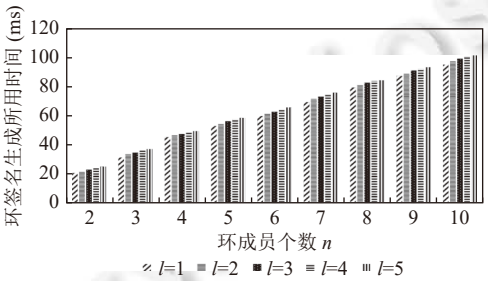


图 5 LRRS 环签名生成时间

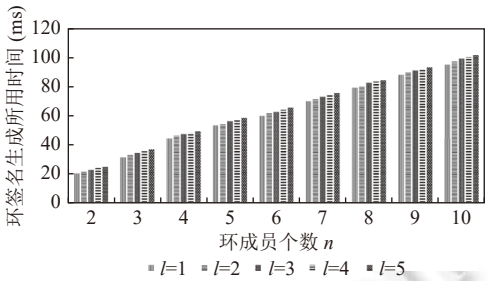


图 6 LRRS 环签名验证时间

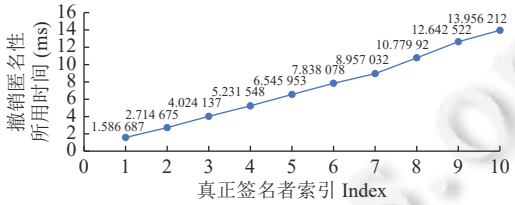


图 7 撤销 LRRS 环签名匿名性所用时间

6.3 测试结果分析

为了实现交易方身份的可监管恢复, 在算法设计层面上本文引入双线性映射运算生成监管信息以实现环签名的可撤销性, 同时为了保证双线性群上的运算安全, 使用了 DLIN 加密和基于双线性映射的密钥镜像生成算法. 这些运算在一定程度上增加了交易过程中的计算复杂度, 延长了交易时间.

我们将本方案同采用类似技术路线的门罗币单地址环签名交易方案[22]进行了比较, 比较结果如表 4 所示. 其中, n 为环签名的成员个数; $|G_1|$ 表示群 G_1 中元素的长度, $|G_T|$ 表示群 G_T 中元素的长度, $|Z_q|$ 表示有限域 Z_q 中元素的长度; exp_1 表示群 G_1 中的幂次 (标量乘法) 运算, exp_T 表示群 G_T 中的幂次运算; mul_1 表示群 G_1 中的加法运算, mul_T 表示群 G_T 中的乘法运算.

表 4 本方案同门罗币方案[22]的比较

方案	一次性地址计算时间	环签名时间	签名大小	用户地址大小	监管信息
文献[22]	mul_1	$(4n-2)exp_1+(2n-2)mul_1$	$\|G_1\|+2n\|Z_q\|$	$2\|G_1\|$	—
本方案	$2mul_1+3exp_1$	$(2n-1)exp_1+(2n-1)exp_T+(n-1)2mul_1+(n-1)mul_T$	$\|G_T\|+2n\|Z_q\|$	$\|G_1\|$	$3\|G_1\|$

通过关键运算分析可知, 监管的引入的确增加了一次性地址的计算时间和监管信息的存储开销. 但从测试数据分析可知, 在选定位长为 512 比特素数域Z_q的情况下, 方案设计的算法运算时间均为毫秒级. 在目前区块链多应用于非高频交易场景的情况下, 这一结果完全可以符合于交易性能需求.

7 总 结

本文基于门罗币的身份隐私保护技术设计了监管可靠的一次性地址方案和可链接可撤销环签名方案, 在实现基于自主混币的身份隐私保护的同时, 还满足了交易双方身份“可控匿名”的监管要求. 测试结果表明, 该方案符合于非高频交易场景下的性能需求, 可广泛应用于具有身份隐私保护和监管需求的区块链交易系统中.

References:

[1] Li JJ, Yuan Y, Wang FY. Blockchain-based digital currency: The state of the art and future trends. Acta Automatica Sinica, 2021, 47(4): 715–729 (in Chinese with English abstract). [doi: 10.16383/j.aas.c210018]

[2] 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]

[3] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. 2018. https://bitcoin.org/bitcoin.pdf

[4] Monero. About monero. 2018. https://getmonero.org/knowledge-base/about

[5] 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]

[6] Meiklejohn S, Pomarole M, Jordan G, Levchenko K, McCoy D, Voelker GM, Savage S. A fistful of bitcoins: Characterizing payments among men with no names. In: Proc. of the 2013 ACM Conf. on Internet Measurement. Barcelona: ACM, 2013. 127–140. [doi: 10.1145/2504730.2504747]

[7] Zhao C, Guan Y. A graph-based investigation of bitcoin transactions. In: Proc. of the 11th IFIP Int'l Conf. on Digital Forensics XI. Orlando: Springer, 2015. 79–95. [doi: 10.1007/978-3-319-24123-4_5]

[8] 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]

[9] Valenta L, Rowan B. Blindcoin: Blinded, accountable mixes for bitcoin. In: Proc. of the 2015 Int'l Workshops Financial Cryptography and Data Security. San Juan: Springer, 2015. 112–126. [doi: 10.1007/978-3-662-48051-9_9]

[10] Maxwell G. CoinJoin: Bitcoin privacy for the real world. 2013. https://bitcointalk.org/index.php?topic=279249.0

[11] Ruffing T, Moreno-Sanchez P, Kate A. CoinShuffle: Practical decentralized coin mixing for bitcoin. In: Proc. of the 19th European Symp. on Research in Computer Security. Wroclaw: Springer, 2014. 345–364. [doi: 10.1007/978-3-319-11212-1_20]

[12] Zhu LH, Gao F, Shen M, Li YD, Zheng BK, Mao HL, Wu Z. Survey on privacy preserving techniques for blockchain technology. Journal of Computer Research and Development, 2017, 54(10): 2170–2186 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2017.20170471]

[13] 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]

[14] Noether S, Mackenzie A, The Monero Research Lab. Ring confidential transactions. Ledger, 2016, 1: 1–18. [doi: 10.5195/ledger.2016.34]

[15] Wang Z, Fan J, Cheng L, An HZ, Zheng HB, Niu JX. Supervised anonymous authentication scheme. Ruan Jian Xue Bao/Journal of Software, 2019, 30(6): 1705−1720 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5746.htm [doi: 10.13328/j.cnki.jos.005746]

[16] Camenisch J. Specification of the identity mixer cryptographic library, version 2.3.0. Technical Report, Almaden: IBM, 2010. 1−48.

[17] Yang YT, Cai JL, Zhang XW, Yuan Z. Privacy preserving scheme in block chain with provably secure based on SM9 algorithm. Ruan Jian Xue Bao/Journal of Software, 2019, 30(6): 1692−1704 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5745.htm [doi: 10.13328/j.cnki.jos.005745]

[18] Ra G, Kim T, Lee I. VAIM: Verifiable anonymous identity management for human-centric security and privacy in the Internet of Things. IEEE Access, 2021, 9: 75945–75960. [doi: 10.1109/ACCESS.2021.3080329]

[19] Zhao XQ, Li Y. Auditable and traceable blockchain anonymous transaction scheme. Journal of Applied Sciences, 2021, 39(1): 29–41 (in Chinese with English abstract). [doi: 10.3969/j.issn.0255-8297.2021.01.003]

[20] Boneh D, Boyen X, Shacham H. Short group signatures. In: Proc. of the 24th Annual Int'l Cryptology Conf. Santa Barbara: Springer, 2004. 41–55. [doi: 10.1007/978-3-540-28628-8_3]

[21] Fiat A, Shamir A. How to prove yourself: Practical solutions to identification and signature problems. In: Proc. of the 1986 Advances in Cryptology (CRYPTO 1986). Berlin: Springer, 1986. 186–194. [doi: 10.1007/3-540-47721-7_12]

[22] Van Saberhagen N. CryptoNote v 2.0. 2013. https://decred.org/research/saberhagen2013.pdf

[23] Liu DYW, Liu JK, Mu Y, Susilo W, Wong DS. Revocable ring signature. Journal of Computer Science and Technology, 2007, 22(6): 785–794. [doi: 10.1007/s11390-007-9096-5]

[24] Bellare M, Neven G. Multi-signatures in the plain public-key model and a general forking lemma. In: Proc. of the 13th ACM Conf. on Computer and Communications Security. Alexandria: ACM, 2006. 390–399. [doi: 10.1145/1180405.1180453]

[25] Schnorr CP. Efficient signature generation by smart cards. Journal of Cryptology, 1991, 4(3): 161–174. [doi: 10.1007/BF00196725]

[26] Chase M, Lysyanskaya A. On signatures of knowledge. In: Proc. of the 26th Annual Int'l Conf. on Cryptology. Santa Barbara: Springer, 2006. 78–96. [doi: 10.1007/11818175_5]

[27] Fujisaki E, Suzuki K. Traceable ring signature. IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences, 2008, E91.A(1): 83–93. [doi: 10.1093/ietfec/e91-a.1.83]

[28] Pairing-based Cryptography for Go. 2018. https://github.com/Nik-U/pbc

附中文参考文献:

[1] 李娟娟, 袁勇, 王飞跃. 基于区块链的数字货币发展现状与展望. 自动化学报, 2021, 47(4): 715–729. [doi: 10.16383/j.aas.c210018]

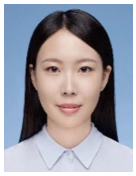
[2] 姚前, 张大伟. 区块链系统中身份管理技术研究综述. 软件学报, 2021, 32(7): 2260–2286. http://www.jos.org.cn/1000-9825/6309.htm [doi: 10.13328/j.cnki.jos.006309]

[12] 祝烈煌, 高峰, 沈蒙, 李艳东, 郑宝昆, 毛洪亮, 吴震. 区块链隐私保护研究综述. 计算机研究与发展, 2017, 54(10): 2170–2186. [doi: 10.7544/issn1000-1239.2017.20170471]

[15] 王震, 范佳, 成林, 安红章, 郑海彬, 牛俊翔. 可监管匿名认证方案. 软件学报, 2019, 30(6): 1705−1720. http://www.jos.org.cn/1000-9825/5746.htm [doi: 10.13328/j.cnki.jos.005746]

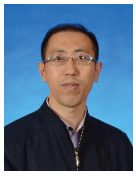
[17] 杨亚涛, 蔡居良, 张筱薇, 袁征. 基于SM9算法可证明安全的区块链隐私保护方案. 软件学报, 2019, 30(6): 1692−1704. http://www.jos.org.cn/1000-9825/5745.htm [doi: 10.13328/j.cnki.jos.005745]

[19] 赵晓琦, 李勇. 可审计且可追踪的区块链匿名交易方案. 应用科学学报, 2021, 39(1): 29–41. [doi: 10.3969/j.issn.0255-8297.2021.01.003]



宋靖文(1995－), 女, 硕士, 主要研究领域为区块链, 数字货币.

韩旭(1993－), 男, 博士, CCF 学生会员, 主要研究领域为区块链, 数字货币.



张大伟(1974－), 男, 博士, 副教授, 博士生导师, CCF 专业会员, 主要研究领域为信息安全, 区块链.

杜晔(1978－), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为云安全, 网络攻防, 信息保密技术.