基于 DKIM 与零知识证明的社交恢复方案设计

黄世龙,周　迅,梁培利
(成都信息工程大学 区块链产业学院,四川 成都 610225)

摘　要:区块链技术的快速发展推动了去中心化账户管理的广泛应用,尤其在数字资产和身份管理方面,然而,传统的私钥管理模式存在着私钥丢失或泄露风险,用户一旦丢失私钥便永久失去对资产的控制权。近年来,智能合约抽象账户已成为一种新型的去中心化数字身份管理方式,不仅提高了用户数据的隐私保护,还允许用户自主掌控身份信息,提供了去信任、不可篡改的身份验证手段。智能合约账户通过智能合约与用户身份绑定,能够自动化执行身份验证和资产管理任务,赋予用户对数字资产的自主控制权。现有的许多方案使用安全多方计算技术将私钥分割成多个片段分布存储,防止了单点故障,且仅在满足一定数量的参与方协作时才能重建私钥。这种方法提高了安全性,但在实际应用中,当门限无法达成时,私钥恢复往往面临失败的风险。为了解决这个问题,文中方案结合多方计算技术和基于域名识别邮件协议的社交恢复,通过引入邮件验证和零知识证明为私钥恢复提供了备用路径。这样,即使门限条件无法满足,用户也可以通过安全的社交恢复方式重获对合约账户的访问权。该方案提高了私钥管理的灵活性和安全性,为区块链生态中的数字资产管理提供了新的思路。

关键词:区块链;智能合约;安全多方计算;域名识别邮件协议;零知识证明;社交恢复
中图分类号:TP309.3　　**文献标识码**:A　　**文章编号**:1673-629X(2025)07-0196-11
doi:10.20165/j.cnki.ISSN1673-629X.2025.0058

Design of Social Recovery Scheme Based on DKIM and Zero Knowledge Proof

HUANG Shi-long,ZHOU Xun,LIANG Pei-li
(School of Blockchain Technology,Chengdu University of Information Technology,Chengdu 610225,China)

Abstract:The rapid development of blockchain technology has promoted the widespread adoption of decentralized account management, particularly in digital asset and identity management. However,traditional private key management methods come with the risk of key loss or exposure,whereby users lose permanent control over their assets if they lose their private keys. Recently,smart contract-based abstract accounts have emerged as a new decentralized digital identity management approach. This approach not only enhances user data privacy protection but also allows users to control their identity information autonomously,providing a trustless and tamper-proof method for identity verification. Smart contract accounts bind user identities through smart contracts,enabling the automated execution of identity verification and asset management tasks and granting users autonomous control over their digital assets. Many existing solutions leverage secure multi-party computation (MPC) technology to split the private key into multiple fragments stored across various locations,thereby preventing single points of failure. Only when a certain number of participants collaborate can the private key be reconstructed. Although this method enhances security,in practical applications,private key recovery can fail when the required threshold cannot be reached. To address this issue,the proposed solution combines MPC technology with a DomainKeys Identified Mail (DKIM) for social recovery,introducing email verification and zero-knowledge proof as a fallback mechanism for private key recovery. This way,even if the threshold condition is not met,users can regain access to their contract accounts through secure social recovery. The proposed approach enhances both the flexibility and security of private key management,offering new avenues for digital asset management within the blockchain ecosystem.

Key words:blockchain;smart contract;secure multi-party computation;DomainKeys Identified Mail (DKIM);zero-knowledge proof;social recovery

收稿日期:2024-10-31　　**修回日期**:2025-03-04

作者简介:黄世龙(2000-),男,硕士研究生,研究方向为区块链技术与区块链应用;通信作者:周　迅(1994-),男,博士研究生,研究方向为分布式存储。

0　引　言

在当今数字经济的飞速发展中,区块链技术凭借其去中心化、不可篡改、透明性等独特特性,迅速成为众多新兴应用的基础架构[1]。区块链技术无论是在金融科技、供应链管理,还是在数据存储与共享、数字身份验证等领域,都展现了强大的创新潜力与广泛的应用前景。将链上智能合约作为去数字身份后,能在这些众多新兴区块链应用中获得非常好的去中心化体验[2]。然而,随着区块链技术的不断普及和应用场景的日益丰富,如何有效管理和保护用户数字身份私钥这一问题逐渐成为行业的关注焦点。

私钥在区块链生态中不仅是用户访问自己数字身份智能合约账户的唯一凭证,也是授权和执行交易的核心工具[3]。私钥的丢失或泄露将直接导致用户无法访问其数字资产或执行任何操作,更为严重的是,一旦私钥被恶意攻击者获取,用户的资产可能会面临被非法转移甚至彻底丢失的风险。因此,如何设计出一种既能确保私钥安全性,又能在私钥丢失的情况下实现有效恢复数字身份的方案,成为当今区块链技术发展中的重要挑战和研究课题。现在对于私钥管理的方案大多为自主私钥管理和中心化私钥托管[4],但这两种方案分别有自己的优势与不足:中心化私钥管理方案为普通用户提供了更简便的使用体验和较低的技术门槛,但其安全性和隐私性较弱的问题仍不可忽视;而自主私钥管理方案的关键挑战在于用户必须承担全部责任,一旦私钥丢失或泄露,往往会导致无法挽回的损失。尽管自主私钥管理在提升用户身份与资产控制权方面具有显著优势,其复杂的操作流程和较高的技术门槛对普通用户构成了较大挑战。

随着区块链和加密货币技术的不断发展,如何在便利性与安全性之间找到平衡,仍是众多私钥管理方案亟需解决的关键问题。近年来,社交恢复与安全多方计算(Multi-Party Computation,MPC)技术结合的方案成为了区块链数字身份保护方案中主流的方案。2021 年 Vitalik Buterin 在其发表的文章中认为社交恢复钱包的广泛应用能够有效应对数字身份的私钥丢失问题[5],传统数字身份方案依赖用户单独维护密钥,而社交恢复方式的优点在于避免了单点故障,并为用户提供了对数字身份和私钥额外的安全保障,尤其是在设备丢失或被黑客攻击的情况下。另外,MPC 技术也极大地提升了私钥管理的安全性与可靠性,MPC 技术在这一背景下,通过分散信任和秘密共享的方式,为社交恢复提供了更稳固的保障[6]。

Daniel Escudero 在其研究指出,MPC 可以通过将私钥分成多个秘密共享片段,分布在不同的参与方手中,确保即使某些参与者失去可信度或合作意愿,剩余的参与者仍可以共同重建私钥。这显著减少了对单一联系人或少数几个人的依赖,有效降低了信任集中所带来的安全风险[7]。著名的图灵奖获得者 Adi Shamir 在 1979 年提出了 Shamir 秘密共享(Shamir's Secret Sharing,SSS)方案,这一经典的方案通过将私钥分割成若干份,每份单独存储在不同的受信任实体或节点中,只有达到某个预设的门限时,分片才能被组合以重构出原始的私钥。Shamir 的秘密共享方案通过线性插值技术来保证即使有一部分分片丢失,仍然可以通过剩余的分片恢复出私钥,避免私钥完全丢失的风险。该方案的优势在于其简洁性和高效性,特别适用于在社交网络信任模型中分发密钥分片,允许用户通过信任的联系人来协助恢复私钥[8]。

另外,Cramer 与 Nielsen 等学者提到了一种基于门限签名方案(Threshold Signature Scheme,TSS)的安全多方计算方法[9],该方法能够在保证安全性的前提下,允许多个节点协同完成签名,而无需任何单个节点拥有完整的私钥。他们的研究表明了如何在多方计算环境下,使用门限加密技术来分布式地存储和管理私钥分片,从而在关键时刻,多个分片持有者能够共同生成签名,而无需暴露任何人的完整密钥。这种方案特别适用于高安全性环境中,能够防止单点故障,并显著提高私钥管理过程中的容错能力。这两种技术都为社交恢复提供了坚实的基础,在 TSS 方案中,私钥管理更加动态和安全,签名生成不需要集中化的密钥存储;而在 SSS 方案中,私钥的安全性与隐私性通过分片机制得到保障。两者均在区块链和分布式系统中被广泛应用,为用户提供了安全和便捷的私钥恢复机制。David Evans 等人的研究表明,通过 MPC 还可以优化 Gas 费用,因为在区块链上执行的智能合约只需要各方提供计算的结果而不需传输全部数据,这降低了网络传输和存储的需求。MPC 不仅能确保系统的安全性,同时也能提升效率,进一步推动了社交恢复数字身份在 Web3 应用中的可行性[10]。

虽然社交恢复与 MPC 等新兴技术在一定程度上缓解了数字身份管理的难题,但在便利性与安全性之间仍存在一些平衡问题。社交恢复依赖于人际信任网络,这样也带来了潜在的社交信任风险,社交恢复主要依赖于用户指定的联系人来帮助恢复私钥,这一方式虽然提高了防止单点故障的能力,但仍存在社交信任风险,即如果部分联系人不再可信,恢复过程的安全性将受到威胁,从而影响恢复过程的可靠性。另外社交恢复与 MPC 技术结合的方案在某些极端情况下仍存在丢失风险:若多个参与方有单点故障问题,例如私钥分片丢失导致达不到阈值,从而无法进行签名或者无法进行恢复私钥。另外,例如 Argent Wallet Scheme 中

通过增加信任方和恢复阈值的方案在实际应用中也并不能解决阈值问题[6],并且信任方的增加会导致链上社交恢复时所消耗的 Gas 消耗增加。因此,找到适合具体场景的私钥管理方案,并在便捷性和安全性之间取得平衡,将决定这些技术的成功与否。综上,如何在提供高安全性的同时,减少对用户日常操作的干扰,仍是这一领域未来发展的关键挑战。

为此,该文提出了一种基于 DKIM 与零知识证明的新型社交恢复方案。该方案首先基于现有的 MPC 和通信计算开销较低的 SSS 技术将私钥分割并分布存储在多个独立的实体中,以确保在部分实体失效的情况下,用户仍能通过剩余实体恢复其私钥。当出现剩余实体没有达到阈值从而无法进行私钥恢复的情况,使用基于域名密钥识别邮件技术(DomainKeys Identified Mail, DKIM)结合零知识范围证明 Bulletproofs 的社交恢复机制,用户可通过可信的外部邮箱验证流程恢复账户访问权限,进一步增强了系统的安全性和可靠性。另外,该方案基于智能合约账户作为用户的数字身份,实现了数字账号可社交恢复、批量交易简单以及高安全性,在去中心化和安全性之间谋求平衡,为未来区块链多链发展环境下的数字资产保护提供新的思路和技术路径。

1 相关知识

1.1 智能合约账户

由于区块链是建立在去中心化和开放的核心理念之上的,所以对于区块链上的应用,用户需要承担更多的责任来维护自己的账号。用户使用私钥来控制和管理数字身份和数字资产,只有用私钥来进行签名,其他人才能通过验证该签名来确认这是私钥所有者的意图,最终会在区块链中全网广播与确认。由于区块链链上交易的复杂性,交互操作门槛高一直是区块链大规模采用的最大障碍之一。

智能合约账户作为区块链技术中的重要组成部分,它可以通过编写好的代码自动执行特定任务,而无需人为干预。作为数字身份与区块链结合的一个典型例子,智能合约账户与传统账户不同,具有可编程性和自治性,能够根据预设条件自动触发操作,比如交易、验证和权限管理等[1]。在智能合约账户中,操作和管理并不依赖个人,而是由智能合约代码来执行,这大大降低了区块链的交互门槛。另外在数字身份的管理中,智能合约账户提供了一种新型的、去中心化的身份验证机制。数字身份是用户在网络世界中的唯一标识,通常涉及用户的私钥、公钥及其相关的身份验证信息。在传统身份系统中,身份认证通常依赖中心化的实体或组织(如政府、银行等)。而通过智能合约账户,用户可以基于区块链技术进行自我主权身份管理,确保数据隐私和安全性。它允许用户完全控制自己的身份信息,并且能够通过智能合约自动验证和管理身份,无需依赖第三方的授权或管理[3]。此外,智能合约账户还可以通过多种机制来增强数字身份的安全性和恢复能力,特别是在用户的私钥丢失或损坏的情况下,MPC 和社交恢复等机制可以通过可信联系人或第三方验证来恢复身份[5]。这种设计使智能合约账户不仅能够提升去中心化应用中的操作效率,还为用户提供了更高的安全性和隐私保护。

1.2 安全多方计算与 Shamir 秘密共享

MPC 是一种分布式计算协议,是一种分布式计算技术,旨在让多个参与方在不泄露各自私有数据的前提下共同计算一个函数的结果[11]。其核心优势在于确保隐私和数据安全,即便各方之间缺乏完全信任。MPC 的工作原理可以简单理解为,在计算过程中,各方不直接共享原始数据,而是通过特定的密码学方法,仅共享部分数据片段。通过这些数据片段,参与方可以在计算结束时得到所需结果,但不会得知其他方的输入,混淆电路协议[12]和 GMW 协议[13]是 MPC 的两种经典实现。

Shamir 秘密共享算法是一种基于 MPC 的高效秘密分享算法,用于在一个群体中分发私密信息,除非群体中的一定数量的成员共同合作,否则无法揭示秘密[8]。SSS 具有信息论安全的特性,即使攻击者窃取了一些分享,除非他们窃取了足够数量的私钥片段,否则他们无法重构出完整私钥,与 TSS 不同的是,SSS 在控制 EOA 账户时,会直接使用完整私钥进行签名。

1.3 零知识证明与 Bulletproofs

零知识证明(Zero-Knowledge Proof,ZKP)是一种密码学技术,允许一方(证明者)向另一方(验证者)证明某一声明的真实性,而不泄露与声明相关的任何具体信息。该技术在隐私保护、区块链以及身份认证等领域中有着广泛的应用。零知识证明要求满足三个基本属性:完备性(Completeness)、可靠性(Soundness)和零知识性(Zero-Knowledge)。完备性保证如果声明正确,证明一定会通过验证;可靠性保证如果声明不正确,验证者不会被误导;而零知识性则确保验证者在验证过程中获取不到任何除声明是否正确外的其他信息。零知识证明广泛应用于区块链系统,尤其是在确保交易隐私的加密货币中,如 Zcash[14]。在此类应用中,零知识证明允许验证者在不暴露交易金额或交易双方身份的情况下,验证交易的正确性。这不仅保证了数据的隐私性,还能提高系统的安全性和透明度。

Bulletproofs 是于 2018 年提出的一种高效的非交互式零知识证明(Non-Interactive Zero-Knowledge

Proof,NIZK)[15],旨在优化区块链中加密货币交易的隐私保护和性能。Bulletproofs 通过缩短证明的长度和减少验证时的计算量,极大地提高了区块链交易的隐私性和扩展性。其最大的优势在于,与传统零知识证明技术相比,它不需要可信设置(Trusted Setup)阶段,这意味着系统在生成初始参数时不依赖于第三方,减少了潜在的安全风险。Bulletproofs 基于椭圆曲线密码学和内积证明(Inner Product Proofs),用于生成简短的零知识证明。内积证明是 Bulletproofs 的核心部分,它通过递归证明方法,将证明的计算复杂度从线性缩减到对数级别。相比于 zk-SNARKs,Bulletproofs 无需生成大的公共参数,且证明的大小与参与方的数量呈对数增长,而不是线性增长,因此 Bulletproofs 在资源受限的系统中(如区块链)有显著的优势[16]。本方案社交恢复中的核心是基于 Bulletproofs 算法中的佩德森承诺,在方案分析一节中将会做详细介绍。

1.4　社交恢复与域名识别邮件协议

社交恢复(Social Recovery)是一种用于数字身份管理和私钥保护的机制,通过分布式的社交联系人或信任实体,用户可以在私钥丢失的情况下恢复对账户的控制权。该方案的核心在于减少单点故障的风险,利用多个信任方分散管理私钥或身份验证数据。用户在设置社交恢复时,选择多个可信联系人或设备,私钥被分割并存储在这些信任方中。当用户丢失设备或私钥时,至少一部分信任方需要参与验证,帮助用户恢复访问权限。与传统的密钥备份不同,社交恢复方案具有更强的容错能力和灵活性,特别适用于去中心化应用和区块链环境中[5]。基于域名识别的邮件协议(DomainKeys Identified Mail,DKIM)是一种邮件认证协议[17],用于验证发送方是否拥有发送域名的控制权,邮件服务器利用公私钥加密技术确保发件方的身份和邮件内容的完整性。DKIM 通过为发件域名创建数字签名,附加在邮件头部,使得收件方能够验证邮件是否来自授权的发件服务器,并检测邮件在传输过程中是否被篡改。在社交恢复机制中,DKIM 可以用于保证身份验证信息的邮件安全。用户在无法恢复私钥时,可以通过发送包含身份验证信息的邮件给智能合约账户,通过 DKIM 验证邮件的真实性和发件人的合法性。结合零知识证明,该过程可以确保即使在没有完全暴露用户数据的情况下,也能够完成身份验证[18-19]。

2　方案设计

2.1　基于 MPC-SSS 的数字身份模型

在本方案中,数字身份的设计基于以太坊的账户模型,具体划分为外部拥有账户(External Owned Account,EOA)和智能合约账户(Smart Contract Account,SCA)。EOA 是以太坊中最基本的账户类型,依赖于公私钥对来管理用户与区块链的交互。通过生成密钥对,EOA 的所有者能够直接签署交易、访问数字资产以及管理与数字身份相关的操作。EOA 的控制权完全由私钥掌握,因此用户对账户及其关联的资产拥有绝对的控制权。而 SCA 则是一种更加灵活的账户形式,通过智能合约的部署,允许用户在链上定义账户的行为和逻辑规则。这些规则可以包括自动执行的交易、设定交易的条件限制以及资金支出限额等,从而提高了账户的可定制性和自动化操作的能力[20]。

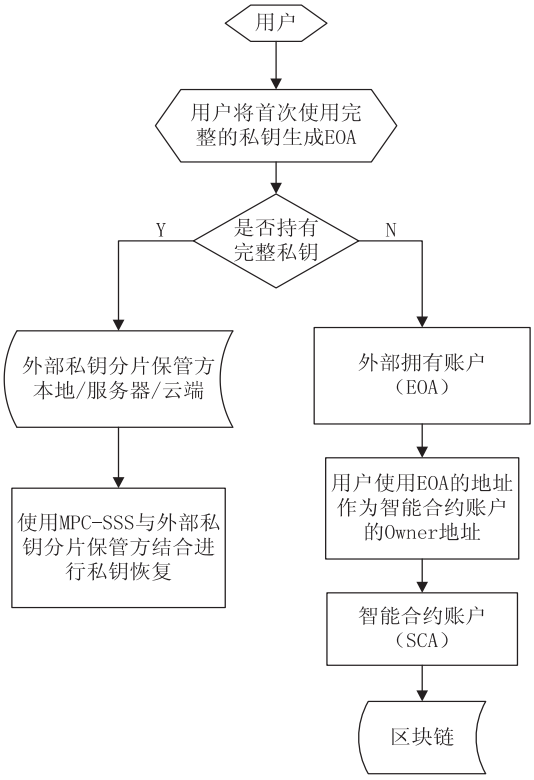


图 1　基于 MPC-SSS 的数字身份模型

在本方案的数字身份模型中(见图 1),用户私钥的管理采用了多方安全计算秘密共享方案(MPC-SSS),这一技术通过将私钥分片存储于多个位置以提升私钥的安全性和可恢复性。具体而言,私钥将被分割为若干份,分别保存在用户本地、服务器以及其他云服务中,从而避免了单点故障带来的安全隐患。该设计与门限签名方案(Threshold Signature Scheme,TSS)有所不同,后者要求多个密钥分片持有者共同协作以完成交易签名。而在 MPC-SSS 方案中,用户依然拥有完整的私钥,仅当完整私钥丢失时,才会启用私钥分片的托管机制,通过收集足够数量的分片来恢复用户的私钥。这一设计确保了用户在通常情况下能够完全掌控其私钥和数字身份,同时也为意外丢失私钥提供了可靠的恢复机制。

在具体应用中,用户通过完整私钥生成 EOA,并将该 EOA 的地址作为其智能合约账户(SCA)的所有者(Owner)地址。SCA 将作为用户的数字身份账户,负责与区块链进行交互。EOA 的主要功能在于生成初始身份标识并作为管理账户的中枢,而 SCA 则通过智能合约进一步扩展了用户账户的灵活性,允许用户在合约中预设交互逻辑,自动执行某些操作并强化账户的安全与控制机制。因此,用户不仅能够通过 EOA 直接控制其资产,还可以通过 SCA 以更加复杂和自动化的方式管理其数字身份。

2.2 DKIM 与零知识证明结合的社交恢复

在 2.1 所述的智能合约账户模型中,私钥的管理主要依赖于 MPC-SSS 方案。虽然该方案在正常情况下能够提供极高的安全性和私钥恢复机制,但在极端情况下,如私钥分片持有者故障或分片丢失时,MPC 方案可能无法完成恢复操作,导致用户失去对其账户的控制。尽管 MPC 在分布式密钥管理中具有显著优势,但该方案在处理某些特殊情况时存在局限性,此时,用户将无法通过常规途径恢复其账户,可能导致永久性损失。社交恢复机制是允许用户依靠其社交圈或受信任的第三方进行账户恢复操作的一种机制。然而,传统的社交恢复机制中,如何确保恢复过程的安全性与隐私性是主要挑战。未经授权的恢复者可能尝试通过社交恢复获得对用户账户的控制,进而窃取用户的身份和资产。

为了解决这个问题,该文提出了一种基于 DKIM 和零知识证明的创新社交恢复机制,之后会将这种机制简称为 ZK-DKIM(Zero-Knowledge Proof's DKIM)。DKIM 是一种邮件认证协议,将其作为验证机制在区块链环境中的引入,可以为去中心化的账户恢复提供新的信任模型和安全保障。通过 DKIM,方案将传统社交恢复流程中的人为信任转移到邮件服务器的身份验证上,实现了对邮件来源和真实性的可靠确认。具体而言,DKIM 允许发送邮件的服务器在邮件头部附加加密签名,接收端服务器则通过查询发送方的 DNS 记录获取其公钥,以验证签名的合法性。这种机制确保了恢复请求的邮件确实来自用户预定的邮件服务器,而非恶意第三方,从而有效防范伪造攻击和中间人攻击,保障恢复过程的完整性。DKIM 在区块链中的应用不仅提升了账户恢复流程的安全性,还极大地增强了方案的去中心化特性。通过将信任建立在邮件服务器的签名机制上,方案不再依赖具体的守护人或社交联系人,而是利用邮件服务商提供的公开可信的基础设施作为恢复验证的信任方。这一机制有效降低了恢复流程中的人为风险,特别是避免了因社交联系人不可靠、社交网络脆弱或易受攻击等问题而导致的安全隐患。此外,DKIM 的公钥验证通过 DNS 记录实现全球分布式查验,与区块链去中心化的架构特点高度契合,使得验证过程能够在区块链的分布式网络中稳定、广泛地执行。

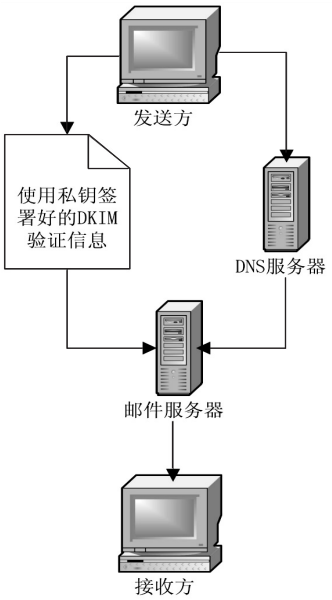


图 2 域名识别邮件协议的流程

在私钥恢复流程中引入 DKIM(见图 2),不仅为区块链用户提供了无须依赖特定社交联系人的恢复路径,还简化了去信任操作。用户只需在预定的邮件服务商上设定恢复者,无需为每个联系人单独建立信任验证,这使得恢复流程更加高效、安全。此外,DKIM 的签名验证和防伪机制进一步保障了身份验证的私密性和唯一性,从根本上防止了恢复过程中常见的邮件伪造、重放攻击等安全问题,显著提高了恢复方案在区块链应用场景中的实用性和适用性。

Hildebrandt[21] 提出了一种将灵魂绑定代币(SoulBound Tokens,SBTs)用于区块链账户的未来应用,并探讨了其潜力在社交恢复中的使用。然而,SBTs 的社交恢复机制存在不可转让性与账户绑定的局限性,在恢复操作过程中可能会导致用户隐私数据的泄露风险。尽管 SBTs 在确保账户唯一性方面具有优势,但其账户绑定特性可能限制了系统的灵活性,特别是在需要动态调整恢复条件或变更身份验证参数时。因此,SBTs 的社交恢复应用在灵活性和隐私保护方面仍需进一步优化。ZK-DKIM 方案通过零知识证明和同态验证,用户在失去私钥的情况下,可通过安全的社交恢复邮件验证流程重新获取账户控制权,并利用动态生成的新随机数和验证值实现灵活调整。该机制既确保了恢复过程的隐私保护,又增强了系统的可扩展性,从而在无需依赖代币绑定的情况下提供了更加高效和去中心化的社交恢复方案。

另外,Pedin 等人[22] 提出了一种基于智能合约的

社交恢复钱包管理方案,旨在为数字资产用户提供去中心化的私钥恢复机制。该方案通过智能合约协调社交网络中的可信联系人(如家人、朋友)来协助用户恢复账户访问权限,在私钥丢失的情况下为用户提供了一种便捷的恢复途径。然而,这种方式在信任和安全性上存在一些明显的局限。由于依赖用户的社交联系人网络,其安全性依赖于这些联系人的可信程度;如果这些联系人面临安全风险,恢复流程可能受到威胁。此外,方案的恢复过程存在一定的信任分散性问题,过度依赖社交联系人还可能导致用户隐私和安全性受到威胁,尤其当社交联系人数量有限或不可靠时,方案的恢复效率和可靠性难以保证。

相比之下,本方案避免了对特定社交联系人的依赖,使私钥恢复过程更具安全性和去信任化特性。DKIM 验证邮件的真实性,显著降低了中间人攻击和身份伪造的风险,而 Bulletproofs 零知识证明则在恢复过程中保护了用户隐私,即便链上数据被拦截,也不会泄露任何敏感信息。通过这种多层次防护,当前方案不仅不依赖主观的社交信任关系,还降低了社交网络联系人的潜在安全隐患,从而在隐私保护和抗攻击性方面展现了更高的安全性。此外,方案在灵活性上也优于 Pedin 等人的设计。通过动态生成新的验证值和随机数,用户可以在私钥丢失时通过安全的邮件验证流程快速恢复账户,而无需依赖固定的社交联系人网络。这一设计不仅确保了恢复流程的安全性与隐私性,还在不同场景下提供了更灵活的恢复路径,使方案在安全性和去中心化程度上比传统的社交恢复方式具有显著提升。

2.3　方案基本流程

接下来将结合 2.1 基于 MPC-SSS 的数字身份模型和 2.2 基于 DKIM 与零知识证明的社交恢复机制来简要描述本方案的总体流程(见图 3)。基本逻辑步骤如下:

步骤 1:首先判断用户是否拥有完整私钥。若拥有完整私钥,用户使用私钥控制 EOA,EOA 的地址将作为用于管理智能合约账户的 Owner 地址,建立社交恢复所用到的验证值,智能合约账户可签署签名交易并广播到 Ethereum 或者其他区块链网络中;若没有拥有完整私钥,则通过 MPC-SSS 结合规定数量的私钥分片托管方协助恢复私钥。

步骤 2:区块链网络中的节点会验证交易的有效性,并将其添加到区块链上,且智能合约账户的状态将被更新。

步骤 3:MPC 是否成功恢复私钥。若没有成功恢复完整私钥,则用户通过 Mail Server 转发,向智能合约账户发送一个包含用户身份验证信息的邮件,智能合约账户收到邮件后,使用 ZK-DKIM 的技术方案 Bulletproofs 验证邮件,以确保验证结果的有效性。

步骤 4:判断邮件是否通过验证。若邮件通过验证,则是智能合约地址的所有者 Owner,更换新的 Owner 地址,并更新智能合约账户的验证信息,更换新的私钥,实现新的私钥访问智能合约账户;若邮件没有通过验证,则证明不是智能合约地址的所有者 Owner,即邮件所包含的验证值错误,可请求重发,但如果一个邮件地址发送超过一定数量的错误邮件,ZK-DKIM 会将其加入黑名单。

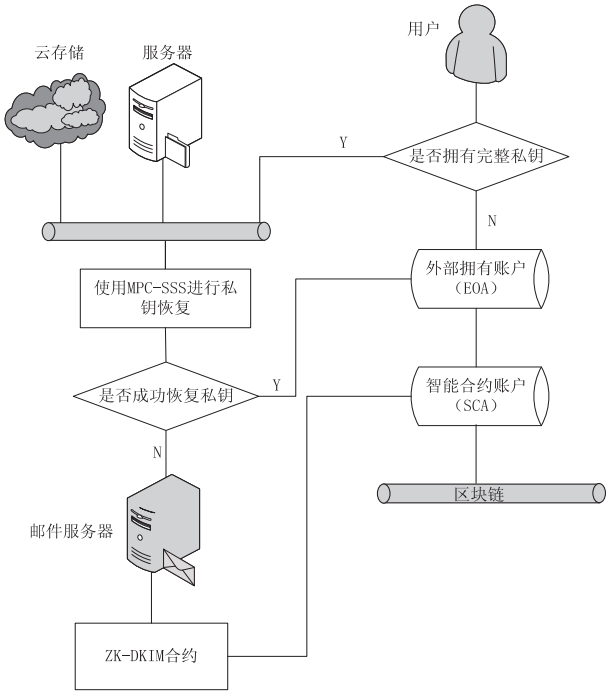


图 3　基于 DKIM 与零知识证明的社交恢复方案流程

3　方案分析

3.1　形式化证明

3.1.1　关于 Bulletproofs 社交恢复验证的形式化证明

Bulletproofs 基于 Pederson Commitments,即佩德森承诺。作为一种加密承诺方案,它最早由计算机科学家崔普和佩德森在 1991 年提出[16]。该方案允许一方(承诺者)对一个值进行承诺,而不泄露该值的具体信息,同时可以在后续阶段向其他方(验证者)揭示该值。佩德森承诺广泛应用于密码学、区块链和零知识证明等领域。

Pederson Commitments 的计算公式如式 1 所示。

$$C = g^m * h^r \bmod p \tag{1}$$

其中,C 为 Commitment,g、h 与 p 是公开的公共参数,p 为 Pederson Commitments 生成过程中的一个素数,这个素数通常是根据特定的安全性和性能要求选择。m 为需要隐藏的值,r 为随机数,g 和 h 分别为两个属于一个大素数阶群的生成元,用于计算承诺值。

Pederson Commitments 拥有离散对数问题和隐藏属性的性质,Bulletproofs 在其基础上提出了一种优化后的向量内积承诺,可以使验证方相信承诺 C 中包含一个在一定范围内的数字 v,而不用揭露 v 的值[15]。利用这个特性可以在 DKIM 的社交恢复过程中加入 Bulletproofs 的验证值进行保护,以确保邮件发送者是 SCA 的所有者。此算法证明了如式 2 的关系,g、h、G 和 V 都是公开信息和作为所保护的验证值。

$$\{(g,h \in G,n;v,r \in Z_p):$$
$$V = g^v h^r \cap v \in [0,2^n - 1]\} \tag{2}$$

3.1.2 Bulletproofs 实现零知识证明的形式化证明

准备阶段,假设在建立 SCA 时验证节点随机选择 $x,y,z \in Z_p^*$,交给用户用于零知识体系所需构造的挑战参数,将本方案用户社交恢复所用到的验证值作为 v。以下为用户操作:

(1)构造两个向量 $\boldsymbol{a}_L$ 与 $\boldsymbol{a}_R$,使得 $\boldsymbol{a}_L$ 满足:

$$< \boldsymbol{a}_L, 2^n > = v,\ \boldsymbol{a}_R = \boldsymbol{a}_L - 1^n \tag{3}$$

(2)基于佩德森承诺构造一个 $\boldsymbol{a}_L$ 与 $\boldsymbol{a}_R$ 的承诺 A。

$$A = h^{\alpha} g^{a_L} h^{a_R} \tag{4}$$

(3)随机选取盲因子 S_L, S_R。

(4)随机选取 t_1, t_2,构造关于 $t(x)$ 一次项和二次项系数的承诺。

$$T^i = g^{t_i} h^{t_i} \in G,\ i = \{1,2\} \tag{5}$$

(5)分别计算如下各值。

$$l = l(x) = \boldsymbol{a}_L - z * 1^n + s_L * x \in Z_p^n \tag{6}$$

$$r = r(x) = y^n \circ (\boldsymbol{a}_R + z * 1^n + s_R * x) + z^2 * 2^n \in Z_p^n \tag{7}$$

$$t = t(x) = < l, r > \in Z_p^n \tag{8}$$

$$\tau_x = \tau_2 * x^2 + \tau_1 + z^2 * \gamma \tag{9}$$

$$\mu = \alpha + \rho * x \in Z_p^n \tag{10}$$

(6)向验证方发送如下数据集合。

$$\{A, S, T_1, T_2, \tau_x, \mu, t(x), l(x), r(x)\}$$

验证阶段,以下为验证方操作:

(1)验证 $t(x)$,判断:

$$g^{t(x)} h^{\tau_x} ?= V^{z^2} g^{\delta(y,z)} T_1^x T_2^{x^2} \tag{11}$$

(2)根据 A 和 S 生成 $l(x), r(x)$ 的承诺 P。

$$P = AS^x g^{-z} (h`)^{zy^n + z^2 2^n} \tag{12}$$

其中:$h` = (h^1, h_2^{y^{-1}}, h_3^{y^{-2}}, \cdots, h_n^{y^{-n+1}})$。

(3)根据生成的承诺 P 判断向量 l, r 是否正确。

$$P ?= h^{\mu} g^l (h`)^r \tag{13}$$

(4)检查 l 和 r 的内积来验证 $t(x)$。

$$t(x) ?= < l, r > \tag{14}$$

(5)如果以上的判断全部正确,即证明了秘密值:

$$v \in [0,2^n - 1]$$

3.2 ZK-DKIM 实现社交恢复的改进

本设计方案考虑到 DKIM 实现社交恢复的过程,对基于 Bulletproofs 的证明过程提出一个改进的步骤。

步骤 1:将受保护的验证值 v 和随机数 r 各自分为 $\{v_1, v_2\}$ 和 $\{r_1, r_2\}$,使得:$v_1 + v_2 = v$,$r_1 + r_2 = r$。

步骤 2:由 $< v_1, r_1 >$ 与 $< v_2, r_2 >$ 两个数值对分别通过 Bulletproofs 所要验证的关系式 2,得到 V_1 和 V_2 两个结果,然后建立数字身份即 SCA 时预留 V_1。

步骤 3:当方案流程中的 MPC-SSS 部分无法恢复私钥时,使用新的私钥生成 EOA,签名一个新的 Owner 地址,生成新的随机数 r。

步骤 4:重新将验证值 v 和新的随机数 r 各自分为 $\{v_3, v_4\}$ 和 $\{r_3, r_4\}$,并重新按步骤 2 的方法用新的两个数值对生成 V_3 和 V_4 两个结果。

步骤 5:向 SCA 发送验证邮件,邮件包含数值对 $< v_2, r_2 >$、Bullet Proofs 的验证值、新的 Owner 地址以及用于下一次社交恢复验证的 V_3。

步骤 6:基于 Bulletproofs 算法的同态性,SCA 收到邮件后验证节点将用户透露的 $< v_2, r_2 >$ 通过关系式 2 得到 V_2,接着将 SCA 中预留的承诺 V_1 与计算得到的承诺 V_2 相乘得出新的 $V`$。

步骤 7:将 $V`$ 参与判断式 11,若最终 Bulletproofs 验证结果成功则将 Owner 地址更换为邮件中送过来的新 Owner 地址,并且更新 SCA 中的信息,同时更换 V_1 为 V_3 作为新的零知识挑战参数用于下一次验证。

4 仿真实验与测试

为验证该文提出的基于智能合约抽象账户的多方计算与社交恢复方案在不同条件下的性能和安全性,本节主要围绕恢复成功率和恢复时间两个主要指标,对比传统的密钥管理和恢复方案,进行系统的仿真实验和测试。

4.1 实验环境

在本实验过程中,在本地以太坊私有链环境中部署了社交恢复智能合约,并对其核心功能进行验证。该合约在 Ganache 环境下运行,以确保在受控的本地区块链中安全、高效地测试账户恢复功能。智能合约代码使用 Solidity 编写,通过 Hardhat 框架完成合约的编译、部署和测试,确保了开发过程的可靠性。实验中采用 Bulletproofs 算法来实现 ZK-DKIM(零知识 DKIM)验证,用于社交恢复过程中的邮件验证环节,从而保障用户在恢复账户时的隐私和安全性。

实验设备的具体配置如下:云服务器平台采用 Ubuntu 20.04 LTS 操作系统,搭载 2 核 Intel Xeon 处理器,内存 8 GB,硬盘 100 GB SSD 存储。该配置能够支持智能合约的高效交互操作,同时具备较高的网络稳定性,为方案的可行性和安全性测试提供了可靠的基础环境。该云服务器环境模拟了实际区块链环境的运

行条件,为实验的稳定性和性能评估提供了保障。

4.2　实验过程

在 ZK-DKIM 智能合约账户的创建流程中,首先通过部署 Wallet 合约来初始化账户,并配置账户的关键属性和权限控制。

Wallet 合约继承了 OpenZeppelin 的 Ownable 模块,确保在账户恢复期间仅有合法所有者可以控制账户(代码见图 4)。这一安全控制机制为账户的所有权和管理提供了基础,同时也确保了用户在恢复账户访问权限时的安全性,另外引用了 Social Recovery 合约的接口。

```
contract Wallet is Ownable, PlguinRegistry, InitialInterfaces {
    ISocialRecovery public socialRecovery;
    uint public nonce;
    uint public recoveryNonce;
    uint256 public emailHash;
    modifier checkNonce(uint256 _nonce) {
        require(nonce + 1 == _nonce, "Invalid nonce");
        _;
        nonce++;
    }
    function initialize(address _socialRecovery, address owner,
uint256 _emailHash) public {
        socialRecovery = ISocialRecovery(_socialRecovery);
        super._transferOwnership(owner);
        emailHash = _emailHash;
    }
    receive() external payable {}
function auth(
    address to,
    uint256 amount,
    uint256 _nonce,
    bytes calldata payload,
    bytes calldata sign
) internal view {
    bytes memory data = abi.encode(to, _nonce, payload, a-
mount);
    bytes32 hash = keccak256(data);
    address addr = ecrecovery(hash, sign);
    require(addr == owner(), "Auth failed: Only owner!");
}
...
}
```

图 4　智能合约账户的关键代码

创建 Social Recovery 合约(代码见图 5),合约通过构造函数接受一个 IDKIMPublicKeyOracle 地址,赋值给 oracle 变量,用以查询特定域名的 RSA 公钥。DKIM 签名使用公钥加密进行身份验证,而预言机(Oracle)则提供了 DKIM 公钥查询功能。这样一来,合约可以通过预言机实时获取签名验证所需的公钥信息,从而验证邮件发件方的真实性。

SocialRecovery 合约实现了通过邮件身份验证恢复账户的功能,核心在于利用 DKIM 协议对邮件来源进行验证,并通过零知识证明技术保证用户身份隐私。合约的构造函数接受一个 DKIM 公钥预言机地址(IDKIMPublicKeyOracle),以便在验证过程中能够实时查询到指定域名的 RSA 公钥,从而确认邮件的发件方身份。预言机作为 DKIM 验证的重要依赖,确保了合约在链上可以访问到最新的公钥信息,便于验证邮件签名的真实性。

```
contract SocialRecovery {
    using strings for *;
    IDKIMPublicKeyOracle oracle;
    constructor(address _oracle) {
        oracle = IDKIMPublicKeyOracle(_oracle);
    }
    struct Headers {
        strings.slice dkim;
        string from;
    }
    struct SigTags {
        strings.slice d;
        strings.slice i;
        strings.slice s;
        strings.slice b;
        strings.slice bh;
        strings.slice cHeader;
        strings.slice cBody;
        strings.slice aHash;
        strings.slice aKey;
        strings.slice[] h;
        uint l;
    }
    function verify(
        string memory toSign,
        string memory body,
        string memory sign,
        uint recoveryNonce,
        address newOwner,
        bool base64Encoded
    ) public view returns (bool success, string memory from)
    {
        SigTags memory sigTags;
        Headers memory headers;
        headers = parse(toSign.toSlice());
        from = parseFrom(headers.from);
        strings.slice memory dkimSig = headers.dkim;
        (sigTags, success) = parseSigTags(dkimSig.copy());
        require(success, "parse sig tags failed");
        success = verifyBodyHash(body, sigTags);
        require(success, "verify body hash failed");
        success = verifySignature(sigTags, toSign, sign);
        if (!success) {
            return (false, from);
        }
        if (!verifyBody(body, recoveryNonce, newOwner,
base64Encoded)) {
            return (false, from);
        }
        return (success, from);
}
...
}
```

图 5　社交恢复合约中的关键代码

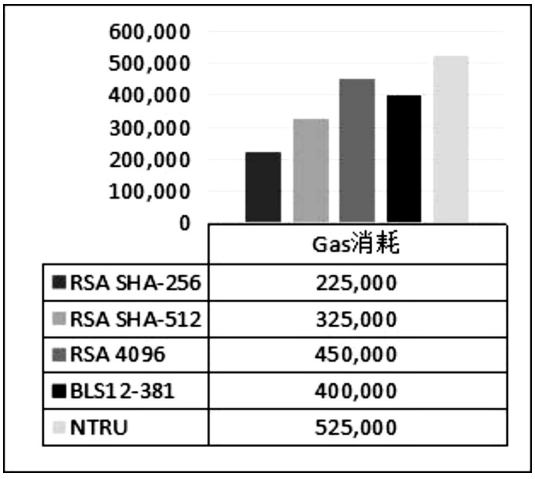
邮件验证的主要逻辑集中在 verify 函数中,该函数结合 RSA-SHA256 算法,接收待签名数据、邮件主体、恢复计数器、签名值、新的所有者地址等多个参数。

verify 函数首先解析邮件内容,提取 from 字段中的发件人地址,然后验证邮件内容的哈希值以确保内容未被篡改。接着,verifySignature 函数从 Oracle 查询所需的公钥,并对 DKIM 签名的真实性进行校验,以确认邮件确实来自声明的发件人。最后,通过 verifyBody 函数对邮件主体与恢复计数器、所有者地址的哈希值进行比对,确保内容确实与账户恢复相关联。

4.3 测试结果

为了评估该方案在以太坊环境下的实际性能,将 Gas 消耗作为主要评估指标,用以比较不同操作的资源效率。重点实验测试了 RSA-SHA256 对比其他高安全性非对称加密算法在链上所需的 Gas 消耗和对比传统的社交恢复方案(如 Argent 钱包[6])部署合约、执行合约社交恢复功能所需的 Gas 消耗。

在加密算法的 Gas 消耗方面,图 6 显示了几种不同的非对称加密算法在以太坊链上的 Gas 消耗情况,其中 RSA SHA-256 的 Gas 消耗为 225 000 Gas,相对较低。其他算法如 RSA SHA-512、RSA 4096、BLS12-381 和 NTRU 消耗明显更高。这些算法尽管在安全性或抗量子攻击能力方面有优势,但由于计算复杂度较高,Gas 消耗相应增加,因此在链上执行成本非常昂贵。特别是 RSA 4096 和 NTRU,其密钥长度和数学复杂度显著增加了资源消耗,使得这些算法难以在链上应用。



	Gas消耗
RSA SHA-256	225,000
RSA SHA-512	325,000
RSA 4096	450,000
BLS12-381	400,000
NTRU	525,000

图 6 以太坊上非对称加密算法的 Gas 消耗

RSA SHA-256 提供了可靠的非对称加密安全性,能够抵抗常见的攻击方式,同时其计算复杂度相对较低,在链上执行时不会过度增加 Gas 成本。相比之下,RSA SHA-512 尽管安全性更强,但其 Gas 消耗高出约 50%,对于大多数应用场景来说,这种额外的安全性并不总是必要。而 RSA SHA-256 的密钥长度和签名大小在链上环境中更为适中,既能够满足方案对数据加密和验证的需求,又控制了计算资源的使用。此外,RSA SHA-256 的广泛支持和兼容性也使其成为优选。它被广泛应用于数字签名、电子邮件验证等场景,技术成熟且稳定,易于集成到现有的区块链环境中。相比于较为新颖的 BLS12-381 或 NTRU,RSA SHA-256 在开发实现和实际应用中具有更多支持资源。因此,使用 RSA SHA-256 可以在不牺牲安全性的前提下,大幅度降低 Gas 消耗和链上执行成本,满足隐私保护、身份验证等社交恢复方案的需求,是一种高效而实用的选择。

在方案总体 Gas 消耗方面,传统方案通常依赖多方计算或门限签名,在链上进行大量的计算和存储操作,导致 Gas 消耗随信任方(Guardians)数量线性增长,恢复操作的费用也随之显著增加[6]。本方案将大部分计算操作移至链下的邮件服务器完成。通过零知识证明实现简洁高效的验证流程,并且链上仅需验证 DKIM 签名的有效性,因此具有恒定的 Gas 成本。而 Argent 钱包的实现也是随着信任方的数量线性扩展,实际上每增加一个信任方就需要额外的 140 000 gas,如图 7 所示。

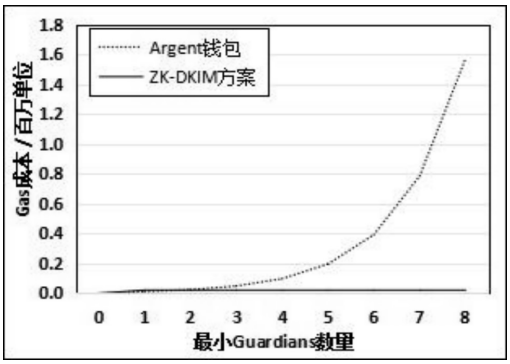


图 7 ZK-DKIM 方案与 Argent 钱包的 Gas 使用成本比较

在方案效率方面,模拟中心化 DKIM 邮件服务器和去中心化分布式验证网络,评估在不同负载和节点数量下的验证时间和效率。节点数量从低到高变化,以测量不同节点下的验证性能。节点数量从 5 个开始,逐步增加到 50 个。每秒发出 1、10、50 和 100 个请求,以模拟低到高的负载场景。分析指标为:

- 平均相应时间:从请求验证到验证完成的平均时间。
- 成功率:在一定负载下的验证通过率。
- 每秒完成的验证请求数量。

验证结果如图 8 所示,验证时间随节点数量增加而变化。中心化验证在低负载下表现较快,而分布式验证在高负载和大节点数情况下能够更有效地分担工作量。在高请求负载下,分布式方案通常比中心化方案更稳定,吞吐量更高且延迟较低。节点增加后,分布式验证的响应时间基本保持平稳,表明其具有良好的扩展性。

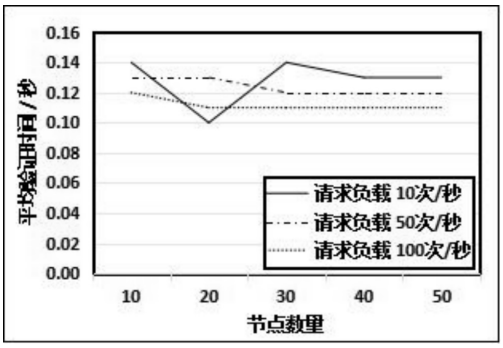


图 8　不同节点数量与请求负载下的平均验证时间

4.4　安全性分析

该方案在隐私保护、抗攻击能力和操作便捷性方面表现出色,有效防止了链上数据的泄露风险,使攻击者即使截获数据也无法获取有效信息。通过链下邮件验证和链上零知识证明的联合机制,该方案克服了传统社交恢复中的监听和信息泄露问题,大大提升了抗攻击性。与常规的 AgentWallet 方案相比(见表 1),ZK-DKIM 进一步降低了对外部信任方的依赖。用户在私钥丢失或门限条件无法达成时,可以通过邮件服务器触发身份验证,无需专门设置守护者,避免了信任方引入的潜在安全隐患,同时使方案适用于更广泛的用户群体。

表 1　安全评估项

安全评估项	ZK-DKIM	AgentWallet
零知识证明(隐私保护)	√	×
抗中间人攻击	√	√
抗身份伪造攻击	√	√
链上数据截获防护	√	×
无需额外的信任方	√	×
防监听与信息泄露	√	×
数据完整性验证	√	√

此外,该方案通过一系列安全设计和验证机制,确保了用户身份验证和私钥恢复过程的安全性,在应对中间人攻击和身份伪造等威胁方面表现出色。通过 Bulletproofs 零知识证明技术,将验证值和随机数进行拆分并多次验证,从而有效保护用户的身份信息和验证值在传输过程中不被泄露或篡改。零知识证明确保了验证信息的隐私性,保证了即便链上数据被截获,攻击者也无法还原出任何关键的身份信息或验证参数,从而保护了链上数据的完整性[23]。

另外,动态地更新验证值和随机数,确保每次验证请求均为实时生成,增强了系统的抗攻击性。这种动态验证机制结合新的私钥和 Owner 地址的生成,确保了只有拥有新私钥的用户才能通过验证,防止了身份伪造和重放攻击的可能性。在验证过程中,利用 Bulletproofs 算法的同态性特性,系统将新生成的验证值与 SCA 中预留的承诺值相结合,形成一种去中心化的、不可篡改的验证链条。这样,在每次身份验证中,智能合约能够自主检测验证信息的真实性,并在验证成功时动态更新合约中的验证参数,进一步降低了中间人和伪造身份攻击的风险。多层次、多方计算和零知识证明相结合的方式,使用户身份验证过程既安全又灵活。它在保证数据隐私性和不可篡改性的同时,避免了传统身份恢复流程中对第三方信任的依赖,为区块链环境中的去中心化身份管理提供了一种安全、高效的解决方案,能够有效抵御中间人攻击、身份伪造和数据截获等常见攻击。

5　结束语

该方案旨在提供一种数字身份的社交恢复思路,为用户在去中心化环境中更好地恢复对数字资产和身份的控制权。通过结合多方计算技术与基于域名识别的邮件协议,该恢复机制允许用户在面临私钥丢失或无法满足门限条件时,仍能通过社交验证方式重新获得对智能合约账户的访问权限。此方案不仅增强了用户对数字资产的管理灵活性,还提升了整体系统的安全性,为去中心化账户管理提供了更为可靠的解决方案。但对于社交恢复所用到的 DKIM 邮件服务器的管理仍可能存在中心化的安全问题,例如恢复请求被拦截或者服务器崩溃。后续可考虑将该机制与去中心化邮件服务相结合,例如通过联邦邮件技术,邮件验证可在多个服务器上协同完成,即便某一服务器出现问题,其他节点仍可验证,从而减小单一服务商的风险,以进一步提升安全性与抗攻击能力。

参考文献:

[1] BUTERIN V. Ethereum white paper[J]. GitHub Repository, 2013,1:22-23.

[2] MAMUN M A A, ALAM S M M, HOSSAIN M S, et al. A novel approach to blockchain-based digital identity system [C]//Proceedings of the 2020 future of information and communication conference (FICC), volume 1. San Francisco: Springer International Publishing, 2020:93-112.

[3] BUTERIN V. A next-generation smart contract and decentralized application platform[J]. White Paper, 2014, 3(37): 1-8.

[4] PAL O, ALAM B, THAKUR V, et al. Key management for blockchain technology[J]. ICT Express, 2021, 7(1): 76-80.

[5] BUTERIN V. Why we need wide adoption of social recovery wallets[J]. White Paper, 2021, 1(1): 1-10.

[6] SAITO Y, ROSE J A. Reputation-based decentralized autonomous organization for the non-profit sector: leveraging blockchain to enhance good governance [J]. Frontiers in Blockchain, 2023, 5: 1083647.

[7] ESCUDERO D. An introduction to secret-sharing-based secure multiparty computation[J]. Cryptology ePrint Archive, 2022,2022(1):1-30.
[8] SHAMIR A. How to share a secret[J]. Communications of the ACM,1979,22(11):612-613.
[9] CRAMER R,DAMGARD I,NIELSEN J B. Multiparty computation from threshold homomorphic encryption[C]//Advances in cryptology—EUROCRYPT 2001: international conference on the theory and application of cryptographic techniques. Innsbruck:Springer,2001:280-300.
[10] EVANS D,KOLESNIKOV V,ROSULEK M. A pragmatic introduction to secure multi-party computation[J]. Foundations and Trends® in Privacy and Security,2018,2(2-3):70-246.
[11] 刘 峰,杨 杰,李志斌,等. 一种基于区块链的泛用型数据隐私保护的安全多方计算协议[J]. 计算机研究与发展,2021,58(2):281-290.
[12] YAO A C. Protocols for secure computations[C]//23rd annual symposium on foundations of computer science (SFCS 1982). Chicago:IEEE,1982:160-164.
[13] SCHNEIDER T,ZOHNER M. GMW vs. Yao? Efficient secure two-party computation with low depth circuits[C]//Financial cryptography and data security: 17th international conference. Okinawa:Springer,2013:275-292.
[14] BERNSTEIN D J, HOPWOOD D, HÜLSING A, et al. SPHINCS:practical stateless hash-based signatures[C]//Annual international conference on the theory and applications of cryptographic techniques. Berlin:Springer,2015:368-397.
[15] BÜNZ B,BOOTLE J,BONEH D,et al. Bulletproofs:short proofs for confidential transactions and more[C]//2018 IEEE symposium on security and privacy (SP). San Francisco:IEEE,2018:315-334.
[16] PEDERSEN T P. Non-interactive and information-theoretic secure verifiable secret sharing[C]//Annual international cryptology conference. Berlin:Springer,1991:129-140.
[17] LEIBA B,FENTON J. DomainKeys identified mail (DKIM): using digital signatures for domain verification[C]//Fourth conference on email and anti-spam (CEAS). Mountain View:IEEE,2007.
[18] 杨 磊,张代远. 基于 DKIM 和评分管理的反邮件系统的设计[J]. 计算机技术与发展,2013,23(7):159-162.
[19] 赵 毅. 基于域密匙认证的反垃圾邮件技术[J]. 计算机科学,2006,33(4):90-91.
[20] BUTERIN V,WEISS Y,TIROSH D,et al. ERC-4337:account abstraction using alt mempool[R]. San Francisco: Ethereum Improvement Proposals,2021.
[21] HILDEBRANDT F. The future of soulbound tokens and their blockchain accounts[C]//Konferenzband zum Scientific Track der Blockchain Autumn School 2022. Mittweida: Hochschule Mittweida,2022:18-24.
[22] PEDIN IV A B,SIASI N,SAMENI M. Smart contract-based social recovery wallet management scheme for digital assets[C]//Proceedings of the 2023 ACM southeast conference. New York:ACM,2023:177-181.
[23] 周 迅. 一种基于 MPC 的可社交恢复数字身份的方法[P]. 中国:CN202410489886.4,2024-07-16.