计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
63(1): 255−272, 2026

中继链环境下一种 2 阶段跨链访问控制机制

张佩云[1] 陶 帅[2] 陈 健[2] 陈子寒[1]
[1](南京信息工程大学计算机学院、网络空间安全学院 南京 210044)
[2](南京信息工程大学软件学院 南京 210044)
(zpy@nuist.edu.cn)

A Two-Stage Cross-Chain Access Control Mechanism in Relay Chain Environment

Zhang Peiyun[1], Tao Shuai[2], Chen Jian[2], and Chen Zihan[1]
[1](*School of Computer Science*, *School of Cyber Science and Engineering*, *Nanjing University of Information Science and Technology*, *Nanjing* 210044)
[2](*School of Software*, *Nanjing University of Information Science and Technology*, *Nanjing* 210044)

Abstract With the rapid development of blockchain technology, assets such as data are no longer satisfied with circulating only on a single blockchain, so cross-chain technology has emerged. Cross-chain technology has improved the problem of blockchain data islands, but it has also brought new challenges to the transmission and sharing of data on different blockchains. On the one hand, different blockchain users usually store data in different ways, and the data verification methods of each blockchain are often not unified. Therefore, users on other blockchains may transmit false or inconsistent data, making it difficult to effectively verify the authenticity and consistency of the data, thereby affecting the reliability of cross-chain access control. On the other hand, although the current proxy re-encryption technology can effectively implement one-to-one access control, a single proxy has a centralization problem and faces the problem of high access control time overhead caused by multi-user concurrency in cross-chain scenarios. In response to the above problems, a two-stage cross-chain access control mechanism in the relay chain environment is proposed, which is designed from two stages: data verification and proxy re-encryption. In the data verification stage, to ensure the authenticity and consistency of the source of encrypted data, combined with the Fiat-Shamir heuristic and IPFS, a data verification method is designed to quickly verify the data, so that the encrypted data can be correctly received by the user, and the encrypted data is securely stored in IPFS to relieve the storage pressure on the chain. In the proxy re-encryption stage, an improved threshold proxy re-encryption method is designed to effectively deal with access control in multi-user scenarios, and the re-encryption process is optimized based on the re-encryption update algorithm. While achieving decentralized re-encryption, the time overhead of re-encryption is reduced, and the efficiency of re-encryption transaction processing for multi-user access control is effectively improved. Experimental results show that the proposed mechanism has a small overhead in data verification, and the overhead of encryption, re-encryption, and decryption processes in multi-user scenarios is better than that of the comparative methods.

Key words relay chain; proxy re-encryption; data verification; access control; data sharing

摘 要 随着区块链技术的迅速发展,数据等资产已不满足于只在单个区块链上流通,因此跨链技术应运而生。跨链技术改善了区块链数据孤岛问题,但也为数据在不同区块链上的传输与共享带来了新的挑战。一方面,不同区块链用户存储数据方式通常不同,各区块链数据验证方式往往也不统一。 因此,其他区

收稿日期: 2024-12-29; **修回日期**: 2025-07-02
基金项目: 国家自然科学基金项目(61872006); 江苏省自然科学基金项目(BK20251888)
The work was supported by the National Natural Science Foundation of China (61872006) and Jiangsu Provincial Natural Science Foundation (BK20251888).

块链上的用户可能传输虚假或者不一致的数据，导致数据难以被有效验证其真实性和一致性，从而影响跨链访问控制的可靠性。另一方面，虽然当前代理重加密技术能有效实现1对1的访问控制，但单一代理存在中心化问题，以及在跨链场景下面临多用户并发导致的访问控制时间开销大的问题。针对上述问题，提出中继链环境下一种2阶段跨链访问控制机制，分别从数据验证和代理重加密阶段2个阶段进行设计。在数据验证阶段，为确保加密数据的来源真实性和一致性，结合Fiat-Shamir启发式与星际文件系统（interplanetary file system，IPFS），设计一种数据验证方法对数据进行快速验证，使得加密数据可被用户正确接收，加密数据在IPFS中安全存储以缓解链上存储压力；在代理重加密阶段，设计一种改进的阈值代理重加密方法，以有效应对多用户场景下的访问控制，基于重加密更新算法对重加密过程进行优化，在实现去中心化重加密的同时，降低重加密的时间开销，并有效提高多用户访问控制的重加密事务处理效率。实验结果表明，所提出的机制在数据验证方面的开销较小，并在多用户场景下加密、重加密及解密过程的开销均优于对比方法。

关键词 中继链；代理重加密；数据验证；访问控制；数据共享

中图法分类号 TP309

DOI: 10.7544/issn1000-1239.202440972 **CSTR:** 32373.14.issn1000-1239.202440972

区块链的概念起源于中本聪2008年发表的论文，它是由按时间顺序排列的数据块组成的链式数据结构[1]。区块链作为一种去中心化的分布式账本，集成了密码学、分布式存储、智能合约、共识机制等技术，具有公开、透明、不可篡改、不可逆的特点[2]。因此，区块链技术在增强网络访问可靠性、降低资源运营成本、提高数据隐私和安全性等方面发挥了巨大作用，同时也在数据交易[3]、物联网[4]等领域得到广泛应用。当前区块链生态系统根据其不同的开放程度大致可分为公有链、私有链和联盟链3种类型[5]。虽然不同类型的区块链各有其独特的优势，但也因为不同的平台和协议，导致各区块链的异构性，信息无法在区块链之间传输，从而形成了各种信息孤岛，该现象给不同链上的访问控制带来了一系列的挑战。因此，单一的区块链架构也越来越不适合创新发展。跨链技术应运而生。该技术旨在解决区块链信息孤岛问题，通过区块链之间的互操作实现数据在不同的区块链之间自由流通。

跨链访问控制是实现多链生态安全互联的关键，但是其面临着新的挑战。由于不同区块链上用户存储数据方式不同、数据验证方式不统一等方面原因，其他区块链上的用户可能传输虚假或者不一致的数据[6]，导致数据的使用者无法及时获取并验证来源于其他链上的数据。因此，数据在跨链时的真实性与一致性验证是一个亟需解决的问题。同时，在不同链之间，基于对称加密的共享密钥访问控制方案不再安全[7]，而代理重加密作为常用的访问控制方案，能够实现1对1的明文隐藏性访问控制。然而，在跨链场景下，单一代理访问控制机制存在的中心化问题违背了区块链的去中心化思想，同时也难以满足跨链场景下实时的多用户并发需求。为了解决上述问题，本文提出中继链环境下一种2阶段跨链访问控制，分别从数据验证和代理重加密阶段2个阶段进行设计。本文的贡献有2点：

1)在访问控制机制的第1阶段，提出一种数据验证方法用于提高数据安全，该方法结合IPFS与Fiat-Shamir启发式，通过构建IPFS以安全存储数据，由数据拥有者基于Fiat-Shamir启发式生成零知识证明，用于声明数据真实性。在验证真实性的基础上，本文根据IPFS中的数据分片标识进行数据一致性验证，以确保进行访问控制的数据是可以被用户正确接收的，且数据验证过程交由中继链上的验证合约来实现自动化验证，而无需第三方参与。

2)在访问控制机制的第2阶段，基于数据验证，提出一种改进的可更新阈值代理重加密方法，通过中继链上的分布式的代理节点，实现去中心化的安全重加密计算；同时，设计重加密更新算法，该算法通过重加密事务的更新来优化重加密流程，实现细粒度的密钥管理与分发，并以映射方式存储用户重加密记录，降低同一用户的计算开销，进而降低多用户场景访问控制的时间开销。

1 相关工作

1.1 跨链技术

传统的跨链技术主要分为四大类：公证人机制、

哈希锁定机制、侧链/中继机制以及分布式私钥控制机制。其中，公证人机制在不同区块链间依赖可信第三方完成跨链操作，该机制实现简单并且部署成本低，但是中心化程度较高，需要对第三方公证人有很高的安全信任假设[8]。哈希锁定机制结合哈希散列和时间锁以确保跨链资产交换的原子性[9]，但无法实现资产转移。侧链机制主要通过双向锚定协议实现侧链和主链间的通信[10]，而中继机制相较于侧链机制则更加灵活。中继机制基于公证人与侧链机制形成中继链，以实现更高的可扩展性和去中心化，确保跨链事务的最终一致性。分布式私钥控制机制旨在利用分布式节点来管理区块链系统中各种资产的私钥，该机制能够将来源链上的资产映射到目标链上，从而实现跨链操作[11]。虽然该机制有去中心化治理的优势，但也存在着密钥分片通信开销大、跨链延迟高的问题。

随着跨链技术的不断发展与完善，一些研究者尝试在已有的技术基础上探索新的跨链方法以应对不同的跨链场景。Zhang 等人[12]提出了一个跨链共识协议用于跨链通信，但该协议需要多种类型节点在各自链上验证交易，增加了额外的验证开销和时延。Luo 等人[13]基于比特币闪电网络提出了一种高效的链下支付通道用于实现跨链支付，该通道结合中继机制来同步通道内的相关信息，但建立通道增加了额外的成本。Zhang 等人[14]基于公证人机制提出了一种差分隐私公证机制，该机制内置效率优先与隐私优先方案，实现不同场景下节点隐私偏好保护，但该机制缺少对公证人统一的管理和维护。Jiang 等人[15]提出了一种动态跨链数据审计模型，该模型结合伽马多重签名方法与动态 Merkle 哈希树，以在与其他区块链交互时保持数据的动态一致，但该模型需要不断更新和维护动态 Merkle 树的索引状态，带来额外的维护成本。Wang 等人[16]提出了一种基于中继的跨链资产转移模型，该模型用于跨链交易的验证与隐私保护，但只适应中低交易频率跨链交易场景。

1.2 跨链数据安全

在跨链访问控制的数据安全性方面，Liu 等人[17]提出了一种基于 2 层跨链架构的分布式并行区块链方案，以有效保证不同区域区块链的所有数据都能经过验证并且安全，但数据验证需要花费大量时间和计算开销。Hu 等人[18]提出了一种基于去中心化动态声誉值评估的跨链数据访问控制方法，该方法通过选举公证人并实时监测恶意交易活动来保证数据访问时的安全性，但是数据交由公证人处理仍存在不可靠的风险。Guo 等人[19]设计了一个支持隐私保护和审计的跨链方案 zkCross，该方案采用 3 个协议以及 2 层跨链架构实现跨链转移、交换以及审计过程中数据的安全，但是该方案没有考虑到在数据跨链前对其进行验证。Xue 等人[20]构建了一种基于双区块链协作的医联体系统安全高效的跨域认证方案，该方案在跨链交互的过程中采用匿名认证来增强安全性，以保护用户数据和隐私不被泄露，但该方案没有考虑到数据验证。Tao 等人[21]提出了一种新颖的机制 Unity，通过利用中继链的阶段性提交协议来确保跨链过程中数据读写的原子性和机密性，但该机制也没有对数据进行验证。马宇航等人[22]结合分布式密钥和属性密码设计了一个多方跨链交易方案，该方案只有当用户的属性满足访问策略时才能解密交易数据，但其没有考虑到数据验证问题。Zhang 等人[23]设计了基于区块链的跨链一致性验证方案 BeDCV，该方案通过修改 Paillier 同态加密对参与计算的数据进行加密以保证数据的安全，但需要基于 Merkle 树对数据进行审计，导致开销较大。Yang 等人[24]提出了一种方法用于在数字资产交易中保护数据隐私性，但该方法没有解决数据真实性问题。

上述方案以跨链场景为基础，数据的安全性大多仅依靠加密方式进行保障，没有考虑到对数据本身的验证，尤其是数据的真实性与数据前后的一致性，在跨链场景下缺乏一定可靠性。

1.3 基于代理重加密的访问控制

在传统区块链场景下基于代理重加密的访问控制研究方面，Wu 等人[25]提出了一种具有可验证重加密的同态代理重加密方案，该方案具有容错和错误检测能力，但结合同态签名的代理重加密方案计算开销较大。Li 等人[26]设计了一种查询隐私保护方案。该方案首先使用同态加密进行数据聚合，其次使用代理重加密实现数据共享，然而该方案需要多次加密，在应对多用户时复杂度较高。Yeh 等人[27]提出了一种增强的代理重新加密方案，以保护数据隐私，但在批量处理多用户数据时计算开销较大。Li 等人[28]设计了一种代理重加密方案，该方案基于属性的可验证密文访问控制机制，实现了密文完整性检查，但该方案引入属性后也增加了用户的计算开销。Pei 等人[29]提出了一种基于身份哈希绑定的代理重加密数据共享方法，但该方案引入的身份哈希使得代理重加密仅限于 1 对 1 的访问控制，难以有效应对多用户场景。以上方案均采用单一代理的方式实现代理

重加密，存在着中心化的安全风险。

在实际应用场景下，单个代理与区块链去中心化的特征不相符。为解决该问题，Raghav 等人[30]提出了一种能够主动更新代理份额的阈值代理重加密方案，实现基于云的安全访问控制，但该方案需要不断更新份额，导致额外的份额管理问题以及计算开销。Zhao 等人[31]设计了一种基于属性的阈值代理重加密方案，利用多个代理进行转换，该方案实现了量子安全性，但需要不断更新重加密密钥，因此，与多个用户交互时开销较大。Wu 等人[32]设计了一种基于身份的阈值代理重加密方案以隐藏委托人的私钥信息，并通过分割重加密密钥来分散代理权力，但身份加密和代理重加密相结合增加了计算开销。以上方案在多用户场景下的访问控制仍面临着较高重加密开销的问题。

基于代理重加密的访问控制相关工作分析如表 1 所示。从表 1 中可知，目前基于代理重加密的访问控制相关工作都存在一定局限性，主要表现为缺少有效的数据验证、代理中心化以及多用户重加密开销大。为解决上述问题，本文提出一种 2 阶段访问控制机制。首先，在第 1 阶段，提出一种基于 Fiat-Shamir 启发式和 IPFS 的数据验证方法，实现数据的真实性与一致性验证。其次，在第 2 阶段，提出一种改进的阈值代理重加密方法，在实现代理去中心化的同时，通过重加密更新算法来优化重加密过程，从而减少 1 对多访问控制时多用户重加密时间开销大的问题。

2 预备知识

2.1 双线性映射

本节介绍双线性配对定义，用于 3.3 节代理重加密阶段中数据拥有者对数据的加密计算以及代理节点对重加密子密文的计算。

对于某个大素数 p，G_1，G_2 是 2 个 p 阶的循环群。定义 1 个映射 $e: G_1 \times G_1 \rightarrow G_2$。若其满足以下 3 种属性，则称映射 e 是双线性映射[33]：

1）双线性。$\forall P, Q \in G_1$，$\forall x, z \in \mathbb{Z}_p$，存在如式（1）所示的关系。

$$e(P^x, Q^z) = e(P, Q)^{xz} \text{。} \tag{1}$$

2）非退化。映射 e 是非退化的，即不是恒等于 1。$\exists P, Q \in G_1$ 使得 $e(P, Q) \neq 1_{G_2}$。

3）可计算。$\forall P, Q \in G_1$，$e(P, Q)$ 是可计算的。

2.2 决策双线性 Diffie-Hellman 假设

本节介绍决策双线性 Diffie-Hellman 假设，用于 5.2 节的安全性分析，目的是证明本文代理重加密方法基于该假设满足明文攻击下的不可区分性。

给定 2 个阶为素数 q 的乘法群 G 和 G_T，群的生成元为 g，确定 1 个双线性映射 $e: G \times G \rightarrow G_T$。决策双线性 Diffie-Hellman（decisional bilinear Diffie-Hellman, DBDH）假设可以表示为[34]：对于随机选择的 $a, b, c \in \mathbb{Z}_p$，输出元组为 (g, g^a, g^b, g^c, T)，其中 $T \in G_T$。对手 A 的任务是区分出真实的 DBDH 元组 $(g, g^a, g^b, g^c, e(g, g)^{abc})$ 和随机元组 (g, g^a, g^b, g^c, T)，其优势可以定义如式（2）所示。

$$ADV_A = \left| Pr[A(g, g^a, g^b, g^c, e(g,g)^{abc}) = 1] - Pr[A(g, g^a, g^b, g^c, T) = 1] \right| \text{。} \tag{2}$$

如果在概率多项式时间内没有对手 A 以一种优势 ADV_A 解决 DBDH 假设，且该优势 ADV_A 不能小到忽略不计，则称该假设在群 G 和 G_T 中是成立的。

2.3 Fiat-Shamir 启发式

本节介绍 Fiat-Shamir 启发式，用于 3.2 节数据验证阶段。数据拥有者基于该启发式，对其持有数据的真实性进行证明与验证。

Table 1 Analysis of Related Work on Access Control Based on Proxy Re-encryption

表 1 基于代理重加密的访问控制相关工作分析

来源	数据验证局限性	访问控制局限性
文献 [25]	仅签名验证数据正确性	代理中心化，仅 1 对 1 访问控制
文献 [26]	仅签名验证数据完整性	代理中心化，仅 1 对 1 访问控制，且多种加密增加重加密计算开销
文献 [27]	未考虑数据的验证	代理中心化，仅 1 对 1 访问控制
文献 [28]	仅哈希值验证数据完整性	代理中心化，仅 1 对 1 访问控制，每次重加密需要额外计算用户属性
文献 [29]	仅智能合约验证数据一致性	代理中心化，重加密时需绑定用户身份，使得访问控制仅限 1 对 1
文献 [30]	未考虑数据的验证	定时更换重加密密钥份额，导致份额管理困难及开销大
文献 [31]	未考虑数据的验证	定时更换重加密密钥，且面对多用户时重加密时间开销大
文献 [32]	仅签名验证数据正确性	重加密时需绑定用户身份，且面对多用户时重加密时间开销大

Fiat-Shamir 启发式是零知识证明技术的一种，可以将交互式证明转换为非交互式证明。与电路形式的 Groth16[35], Plonk[36] 等零知识简洁非交互式知识证明不同，Fiat-Shamir 启发式基于代数运算（如哈希运算、模运算等）实现零知识证明，并通过随机预言机消除证明者和验证者之间的多轮交互，从而将交互式证明转换为非交互式证明[37]。因此，证明者可以向验证者声明某种数据（如身份、签名、交易金额等）的真实性，而无需透露该数据。

由于本文没有涉及验证数据逻辑关系的判断，因此，没有采用基于电路设计的零知识证明技术，而是采用基于代数运算的 Fiat-Shamir 方法并结合 IPFS 进行数据验证，从而实现对数据真实性与一致性的快速验证，以减少计算耗费。

2.4 代理重加密

本节介绍代理重加密技术，用于 3.3 节的代理重加密阶段，并在该阶段优化代理重加密技术以实现跨链场景下 1 对多的访问控制机制。

代理重加密是一种可以对密文进行安全转换的密码学技术，常用在访问控制机制中。该技术通常包含 3 种角色：数据拥有者、数据代理方以及数据使用者[38]。首先，需要数据拥有者基于其私钥和数据使用者的公钥生成一个重加密密钥用于重加密转换。其次，通过半可信数据代理方在不知道主明文的情况下，将密文从一个密钥编码转换为由另一个密钥编码的新密文，即：将加密数据从一个密钥域（数据拥有者的域）转换为另一个密钥域（数据使用者的域），从而实现 1 对 1 的访问控制。当数据拥有者需要分享加密数据给其他域的用户，但不希望将数据解密后再重新加密时，代理重加密允许数据拥有者委托代理将数据在加密状态下转发给其他域的用户，以简化加解密的繁琐流程并避免明文暴露的风险。

3 所提出的访问控制机制

本文提出的 2 阶段访问控制机制涉及的概念符号与含义如表 2 所示。

3.1 系统架构及模型框架设计

本文提出的 2 阶段跨链访问控制机制中，数据拥有者首先基于 Fiat-Shamir 启发式生成零知识证明；其次，对中继链上收集的证明以及 IPFS 中的数据分片标识分别进行验证；最后，验证完成后，通过重加密更新算法对重加密事务进行更新，代理节点需要根据分发的密钥进行各自的重加密密文计算。本文的系统架构设计如图 1 所示。

Table 2 Symbolic Representation and Meaning of Concepts

表 2 概念符号表示与含义

符号	描述
C_M	公钥加密后的密文元数据
D_i	第 i 个重加密子密文
F	重加密密文
G	乘法循环群
$\mathcal{H}$	随机预言机使用的安全哈希函数
J	组合后的密文
M	明文元数据
K_x	数组 χ 中关于当前数据使用者的键
R	重加密密钥
V_x	键 K_x 对应的映射值
W_i	第 i 个双线性映射计算的部分密文
k_i	第 i 个重加密子密钥
y	生成零知识证明选取的挑战
Π	公钥
Σ	私钥

图 1 中各个角色及其在访问控制过程中的职责分别为：

1）数据拥有者（data owner, DO）。DO 是交易的发起者，拥有个人数据的所有权，在跨链过程中负责提供证明、加密数据、数据分片标识、重加密凭证。值得注意的是，本文将完整的密文数据存储在 IPFS 中，DO 发送的是密文元数据，元数据包含的是对称加密密钥、数据内容标识符、数据哈希值等关键信息，以有效减少传输数据的开销和区块链存储压力。

2）IPFS。IPFS 是分布式的文件系统，主要通过各存储节点对数据进行分布式存储，将数据存储到 IPFS 中可以有效减轻链上存储压力。当 1 个大于 256 KB 的数据被添加到 IPFS 网络时，数据会被自动切分成多个分片，并按照一定规则保存到各个存储节点上。该切分过程基于分块算法和默克尔有向无环图实现。此外，每个分片会经过一定的运算得到其相应的分片标识，各存储节点和上传该数据的用户均可查看。由于对数据进行分片式存储，因此 IPFS 的数据上传和读取速度较快。

3）中继链。中继链是实现跨链的桥梁，为不同区块链上的用户提供跨链服务。中继链上设有分布式代理节点，能够与源链和目标链进行交互。代理节点负责处理跨链请求、收集并广播数据等。此外，每个代理节点都是 1 个服务器，多个代理服务器集群能高效处理如智能合约调用、重加密计算等操作，

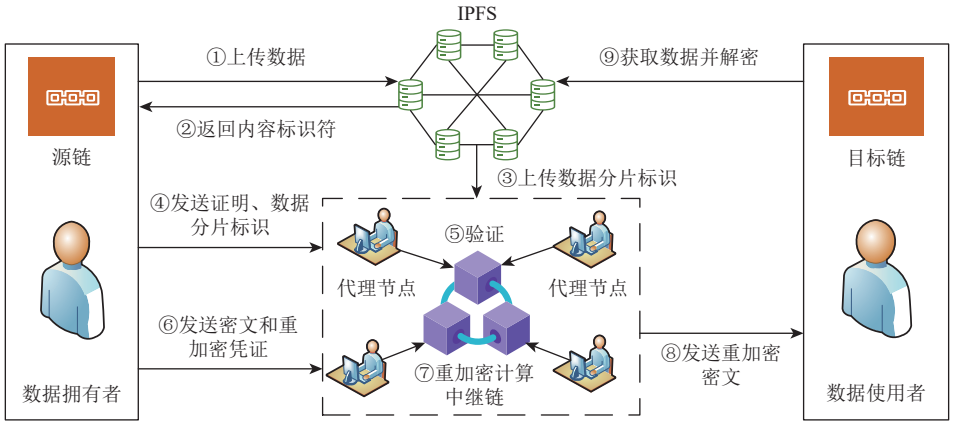


Fig. 1 System architecture in our paper

图 1 本文系统架构

通过接入跨链路由进行跨链业务的发送。

4)数据使用者(data user, DU)。DU 是数据的使用者,在跨链过程中接收数据并使用私钥解密元数据以获取内容标识符和对称密钥,最后根据内容标识符进行数据寻址和原始数据解密。

在图 1 中,步骤①~④用于访问控制的数据验证,步骤⑤~⑧是通过对数据进行阈值重加密转换,本文提出的阈值代理重加密方法具有可更新性。基于图 1,本文给出模型框架图,如图 2 所示。

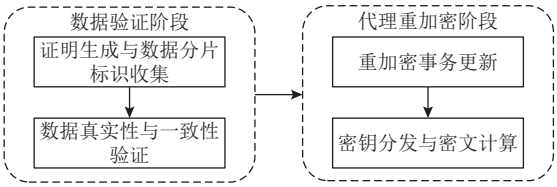


Fig. 2 Model framework

图 2 模型框架

在图 2 中,只有当 DO 的数据被验证真实且一致后,才能提供重加密计算服务,即前一阶段是后一阶段的基础和必备条件。其中,数据验证阶段主要包含证明生成与数据分片标识收集及数据真实性与一致性验证的功能,第 2 个代理重加密阶段主要包含重加密事务更新、密钥分发与密文计算的功能。第 1 阶段用于确保 DO 存储数据可靠且未被篡改,为跨链访问控制提供数据安全的基础,第 2 阶段用于降低访问控制时间开销。

3.2 数据验证阶段

在进行跨链访问控制时需要进行系统初始化。首先通过选择一个安全参数 λ,定义双线性映射 $e: G_1\times G_1\to G_2$,其中 G_1, G_2 是素数阶 q 的乘法循环群。其次,选择一个生成元 $g\in G_1$,然后计算 $T=e(g, g)$。选择 1 个大素数 p 以及 2 个模为 p 的群的生成元 r,定义 2 个哈希函数: H_1: $\{0, 1\}^*\to G_1$, H_2: $G_2\to G_1$,由函数输出公共参数 $P_G=(G_1, G_2, g, r, e, T, p, q, H_1, H_2)$。最后进行密钥生成,根据公共参数 P_G 生成用户公私钥对。源链 DO 以及目标链 DU 的公私钥对分别表示为 (g^a, a) 和 (g^b, b), $a, b\in\mathbb{Z}_p$。

在访问控制机制的数据验证阶段,本文结合 IPFS 以及 Fiat-Shamir 零知识证明技术,以 IPFS 为存储中心安全存储数据,通过收集 IPFS 的数据分片标识以验证数据一致性,通过 DO 基于 Fiat-Shamir 启发式生成关于 IPFS 内容标识符的非交互式零知识证明,以声明并验证数据的真实性。数据验证阶段模型图如图 3 所示。

图 3 中步骤①~④为证明生成与数据分片标识收集,步骤⑤~⑦为数据真实性与一致性验证。具体步骤解释如下:

步骤①. 源链上的 DO 向 IPFS 中安全地存储数据,该数据是完整的,并且是经过所选取的对称密钥加密后的密文形式。

步骤②. 当数据上传到 IPFS 后,DO 将获得由其返回的内容标识符 δ。该内容标识符是基于文件内容生成的且是唯一的,不仅能指向数据内容,还能指向数据在 IPFS 中的存储地址。

步骤③. DO 存储的数据在 IPFS 中被划分为多个数据分片,每个数据分片在 IPFS 内同样有唯一标识 Φ。DO 收集所有的数据分片标识 Φ'。

步骤④. DO 调用证明生成合约,根据 Fiat-Shamir 启发式生成零知识证明的 4 个部分。每个部分的计算具体如式(3)~(6)所示。

$$\omega_1=r^{H(\delta)} \bmod p, \tag{3}$$

$$\omega_2=r^m \bmod p, \tag{4}$$

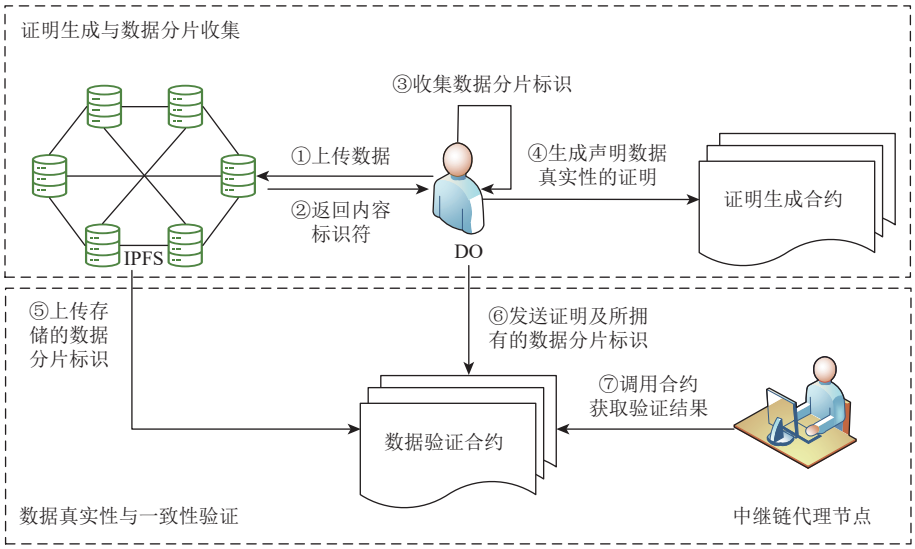


Fig. 3 Data verification phase model

图 3 数据验证阶段模型

$$y=\mathcal{H}(\omega_1,\omega_2), \tag{5}$$

$$\omega_3=m-yH(\delta) \bmod p \text{ 。} \tag{6}$$

式(3)中的 r 表示群的生成元，$H(\delta)$ 为秘密消息，以哈希形式表示，在计算时转化为数值代入各式；式(4)中的 m 为随机数；式(5)中的 y 为生成的挑战，用于非交互式证明，通常由随机预言机选取安全的哈希函数计算得到。和以太坊类似，式(5)中的 $\mathcal{H}()$ 是 Keccak256[39] 函数，用于计算挑战。式(6)用于证明的验证计算。

令 $\mathcal{J}$ 为由 DO 发送到中继链进行数据验证的数据分片标识集合，如式(7)所示。

$$\mathcal{J}=\{\Phi_i|i\in\{1,2,\cdots,n\}\} \text{ 。} \tag{7}$$

式(7)中，n 表示 DO 获取的数据分片个数。

步骤⑤. IPFS 中的每个存储节点将所持有的数据分片标识 Φ' 分别上传到中继链用于数据验证，智能合约收集这些数据分片标识并记录。如果数据大小或 DO 提供的数据产生变化，则该分片标识也会变化，进而能判断数据一致性是否改变。

步骤⑥. DO 通过跨链请求发送包含零知识证明的数组 ω 和所拥有的数据分片标识集合 $\mathcal{J}$ 到中继链的数据验证合约中。数据验证合约自动执行，通过式(5)生成挑战 y'。同时，计算 ω' 并与 ω_2 进行对比，即可验证证明的正确性。ω' 的计算如式(8)所示。

$$\omega'=r^{\omega_3}\omega_1^{y'} \text{ 。} \tag{8}$$

当证明验证成功后，合约将来自 IPFS 及 DO 的分片标识进行对比分析，从而验证数据真实性和数据一致性。

步骤⑦. 中继链上的代理节点查看数据验证合约的验证结果。如果数据真实性和数据一致性通过则开启访问控制的第 2 阶段，即代理重加密阶段。

3.3 代理重加密阶段

在代理重加密阶段，本文在中继链上设置多个代理服务器进行去中心化的重加密，以将密文转化成可以由 DU 的私钥解密的格式。传统的代理重加密是 1 对 1 的访问控制方式，在区块链的匿名性环境下，对于同一份数据，即使是同一用户仍需要多次重加密，在跨链多用户访问控制场景下，反复的重加密计算极大地限制了其效率。本文在阈值代理重加密的过程中设计了一种更新算法对其进行改进，实现可更新阈值代理重加密。该算法通过改进重加密的访问控制，使其更加高效，以用于处理多用户的重加密事务。代理重加密阶段模型图如图 4 所示。

图 4 中步骤①~③为重加密事务更新，步骤④~⑧为密钥分发与密文计算。具体步骤解释如下：

步骤①. DO 首先用自己的公钥对明文元数据 M 进行加密，如式(9)所示。

$$C_M=(g^{\sigma},MT^{u}) \text{ 。} \tag{9}$$

式(9)中，$\sigma=a\times u$ 且 u 为随机系数。元数据 M 包含数据哈希值、内容标识符以及对称密钥等原数据的关键信息。其次，在数据加密后，DO 生成重加密密钥。通过 DO 的私钥和目标链上 DU 的公钥计算，获得重加密密钥 R，如式(10)所示。

$$R=g^{1/a}g^{bv}=g^{1/a+\rho} \text{ 。} \tag{10}$$

式(10)中，$\rho=b\times v$ 且 v 为随机数，$v\in\mathbb{Z}_p$。令 $O=g^{v}$ 作为公开随机参数。

最后，DO 根据 Shamir 秘密共享[40] 生成 1 个多项

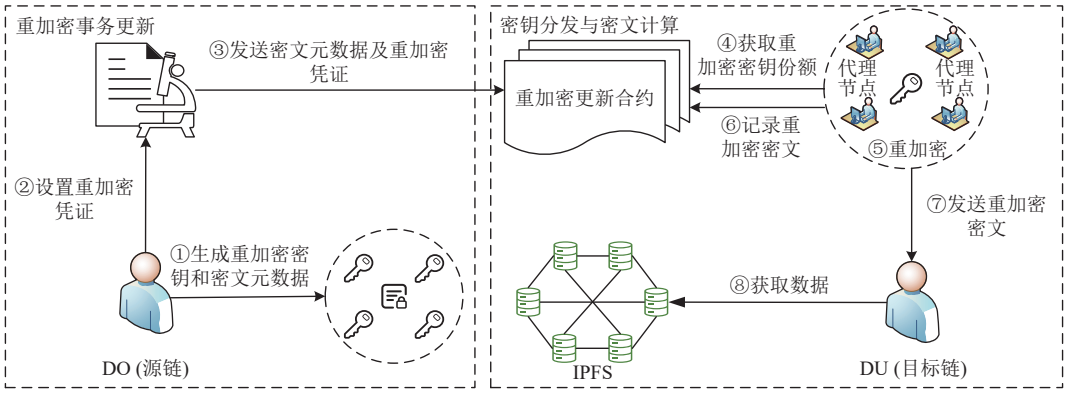


Fig. 4 Proxy re-encryption phase model

图 4 代理重加密阶段模型

式, 如式(11)所示。

$$f(x)=s+a_1x+a_2x^2+\cdots+a_{t-1}x^{t-1}\ 。\tag{11}$$

式(11)中, $s=R$。完整的重加密密钥通过有限域内的运算, 被分割成 N 个份额。其中, 第 i 个重加密子密钥 k_i 的计算如式(12)所示。

$$k_i=g_i^{f(x)}\bmod q,\ i\in\{1,2,\cdots,N\}\ 。\tag{12}$$

步骤②. DO 设置关于 DU 的重加密凭证 γ, 如式(13)所示。

$$\gamma=\{\eta,\langle Address,ID,\mathbb{P}_{\mathrm{DU}}\rangle\}\ 。\tag{13}$$

式(13)中, η 表示重加密子密钥的集合, 如式(14)所示。

$$\eta=\{k_i|i\in\{1,2,\cdots,N\}\}\ 。\tag{14}$$

步骤③. DO 将密文元数据 C_M 和重加密凭证 γ 通过跨链请求发送到中继链上重加密以更新合约所在的地址, 然后合约自动化执行更新算法。

为确保更新事务的执行, 首先, 需要初始化一个键值对映射数组 χ, 用于存储 DU 重加密记录, 数组大小由 DU 的数量决定。数组 χ 中关于当前 DU 的键 K_χ 以及其对应的映射值 V_χ 的构造, 如式(15)所示。

$$\begin{aligned}K_\chi&=H(Address+ID+\Pi_{\mathrm{DU}}),\\V_\chi&=C_M\ 。\end{aligned}\tag{15}$$

在步骤③中进行“Update”和“Reject”这 2 种事务状态的判断。其中, “Update”表示开启更新事务, 而“Reject”则表示更新事务失败。如果触发更新事务进行重加密事务更新, 记录新的 K_χ 和 C_M, 同时广播重加密子密钥和密文给代理节点, 以处理新的重加密事务。

步骤④. 如果开启更新事务, 则中继链上各个代理节点获取由 DO 生成的重加密子密钥份额并根据事务更新结果进行重加密密文计算。由于代理节点无法获得完整的重加密密钥, 从而保证了重加密密钥的安全性。

步骤⑤. 各代理节点根据所拥有的份额进行重加密密文转换。令第 i 个代理节点持有的基于双线性映射计算的部分密文记为 W_i, 如式(16)所示。

$$W_i=e(k_i^{\lambda_i(x)},g^\sigma)=e(g^{f(x_i)\lambda_i(x)},g^\sigma)=T^{\sigma\times f(x_i)\times\lambda_i(x)}\ 。\tag{16}$$

基于式(16), 第 i 个代理节点当前所持有的重加密子密文为 D_i, 如式(17)所示。

$$D_i=(MT^u,T^{\sigma\times f(x_i)\times\lambda_i(x)})\ 。\tag{17}$$

步骤⑥. 当且仅当 n 个代理节点中有 t 个节点正确完成重加密计算时, 令 $p(x)$ 为组合密文所需的插值多项式, 如式(18)所示。

$$p(x)=\sum_{i=1}^{t}p(x_i)\lambda_i(x)\ 。\tag{18}$$

其中 $\lambda_i(x)=\prod_{j=1,i\neq j}^{t}\dfrac{x-x_j}{x_i-x_j}$。组合后的密文 J 如式(19)所示。

$$J=\prod_{i=1}^{t}W_i=\prod_{i=1}^{t}T^{\sigma\times f(x_i)\times\lambda_i(x)}=T^{\sigma\left(\frac{1}{a}+\rho\right)}\ 。\tag{19}$$

基于上述密文计算, 令 F 为重加密密文, 如式(20)所示。

$$F=\left(g^\sigma,MT^u,T^{\frac{\sigma}{a}+\sigma\rho}\right)\ 。\tag{20}$$

收集到重加密密文的代理节点将重加密密文 F 替换为合约中的 C_M, 并标识该 DU 的重加密已完成, 以用于下一次重加密的更新。

步骤⑦. 重加密完成后, 由收集到重加密密文的代理节点将重加密密文 F 发送到目标链上, 并由 DU 获取 F。

步骤⑧. 目标链上的 DU 收到重加密密文 F 后, 用私钥解密元数据密文得到明文元数据 M。根据元数据 M 中的内容标识符和对称密钥, DU 去 IPFS 中寻址下载原数据密文, 并用对称密钥解密获得明文数据。

4 算法设计

基于第 3 节的框架与模型，设计证明生成、数据验证和重加密更新 3 个算法。为进一步描述算法之间的执行及逻辑关系，本文设计的算法流程图如图 5 所示。

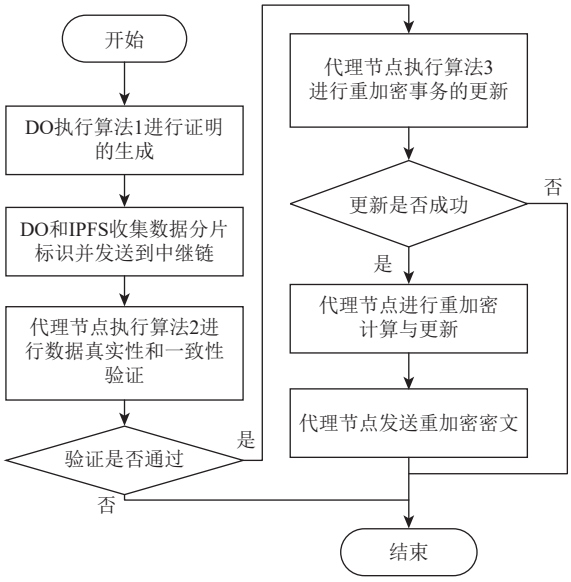


Fig. 5 Flowchart of algorithm

图 5 算法流程图

图 5 中，算法 1 是证明生成（由 DO 调用并执行），用于声明数据的真实性。算法 2 是数据验证（由代理节点调用并执行），用于验证数据的真实性和一致性。算法 1 和算法 2 均属于数据验证阶段。算法 3 是重加密更新算法，属于代理重加密阶段（由代理节点调用并执行），用于对重加密过程的优化。

算法 1. 证明生成。

输入：IPFS 中的内容标识符 δ；

输出：关于存储内容标识符 δ 的证明数组 ω。

① 初始化数组 ω；
② if $\delta \neq$ NULL then
③ 选择 1 个大素数 p 和 1 个生成元 r；
④ 选择 1 个随机数 m；
⑤ 计算 $H(\delta)$；/* $H(\delta)$ 为 δ 的哈希值 */
⑥ $\omega_1 \leftarrow r^{H(\delta)} \bmod p$；/* 通过式（3）计算*/
⑦ $\omega_2 \leftarrow r^m \bmod p$；/*通过式（4）计算*/
⑧ $y \leftarrow \mathcal{H}(\omega_1, \omega_2)$；/*通过式（5）计算*/
⑨ $\omega_3 \leftarrow m - yH(\delta) \bmod p$；/*通过式（6）计算*/
⑩ $\omega \leftarrow [\omega_1, \omega_2, \omega_3]$；
⑪ return ω；
⑫ else
⑬ return NULL；
⑭ end if

算法 1 主要描述如下：

1）算法 1 将 DO 的数据标识符 δ 作为输入，判断其是否为空（行①）。如果 δ 不为空（行②），则需要先选择 1 个大素数 p 以及生成元 r 作为公开参数（行③），然后选择 1 个用于生成证明的随机值 m（行④）。

2）通过计算数据标识符 δ 的数值哈希（行⑤），并根据式（3）~（6）（行⑥~⑨），分别计算零知识证明的 4 个部分 ω_1, ω_2, y, ω_3。然后，ω_1, ω_2, ω_3 被整合到 ω 中，以用于验证真实性并返回 ω（行⑩⑪）。

3）如果 δ 为 NULL，算法 1 返回 NULL（行⑬）。

综上所述，算法 1 中由于哈希输出长度、素数 p 及随机数 m 的位数通常都是固定的。因此，算法 1 的时间复杂度为 $O(1)$。

在算法 1 的基础上，算法 2 设计如下。

算法 2. 数据验证。

输入：证明数组 ω，数据分片标识的集合 $\mathcal{J}$；

输出：“True”和“False”。

① 初始化数组 L；
② if $\omega \neq$ NULL && $\mathcal{J} \neq$ NULL then
③ for (int $i = 1$; $i \leqslant |\mathcal{J}|$; i++)
④ 计算 $H(\Phi_i)$ 并将 $H(\Phi_i)$ 添加到 L；
⑤ end for
⑥ $y' \leftarrow \mathcal{H}(\omega_1, \omega_2)$；/*验证方通过式（5）计算挑战*/
⑦ $\omega' \leftarrow r^{\omega_4} \omega_1^{y'}$；/* 通过式（8）计算*/
⑧ if $\omega_2 == \omega'$
⑨ for (int $i = 1$; $i \leqslant |\mathcal{J}|$; i++)
⑩ 计算 $H(\Phi_i')$；/*计算 IPFS 中存储节点发送的第 i 个数据分片标识的哈希值 */
⑪ if $H(\Phi_i') \notin L$
⑫ return “False”；
⑬ end if
⑭ end for
⑮ return “True”；
⑯ else
⑰ return “False”；
⑱ end if
⑲ else
⑳ return “False”；
㉑ end if

算法 2 主要描述如下：

1)算法 2 将 DO 发送的包含零知识证明的数组 ω 和数据分片标识的集合 $\mathcal{J}$ 作为输入。首先,算法 2 初始化 1 个字节类型数组 L 用于存储分片哈希值(行①)。其次,如果 ω 和 $\mathcal{J}$ 都不为空(行②),则计算 $\mathcal{J}$ 中的每个分片哈希并写入到 L 中(行③④)。

2)在上述步骤完成后,算法 2 首先通过式(5)计算挑战 y'(行⑥),并通过式(8)计算 ω'(行⑦),并与 ω_2 进行比较以校验数据真实性。如果值相等,则验证通过,否则表示没有通过验证。其次,当 ω 得到确认之后(行⑧),算法 2 执行一致性验证步骤(行⑨⑩),通过计算每个 IPFS 存储节点发送的数据分片标识的哈希值,并查找其是否有与 L 中存储的由 DO 发送的数据分片标识的哈希值相一致的值(行⑪)。如果不一致,则返回“False”(行⑫),说明数据大小或数据内容发生了改变,验证不通过。否则,表示通过了一致性和真实性验证,返回“True”(行⑮)。

3)如果因为 ω 或者 $\mathcal{J}$ 为空导致的错误,算法 2 也会返回“False”(行⑯~㉑)。

综上所述,算法 2 中的时间复杂度受到 for 循环影响,该算法中存在 2 个并列的 for 循环,因此算法 2 时间复杂度为 $O(2|\mathcal{J}|)$。

算法 3 设计如下。

算法 3. 重加密更新。

输入:对明文元数据 M 进行公钥加密后的密文元数据 C_M 以及重加密凭证 γ;

输出:“Update”和“Reject”。

① 初始化字节类型数组 χ;
② if $\gamma \neq$ NULL && $C_M \neq$ NULL then
③ for (int $i = 1$; $i \leqslant |\eta|$; i++)
④ if k_i is not valid /* 检查第 i 个重加密子密钥的有效性 */
⑤ return “Reject”;
⑥ end if
⑦ end for
⑧ $K_\chi \leftarrow H(Address + ID + \mathrm{P_{DU}})$; /* 通过式(15)计算 */
⑨ if $K_\chi \in \chi$
⑩ 获取重加密密文 $F = (g^{\sigma}, MT^{u}, T^{\frac{\sigma}{a}+\sigma\rho})$; /*通过式(20)计算 */
⑪ else
⑫ $V_\chi \leftarrow C_M$; /* 通过式(15)计算 */
⑬ 根据 χ 中的键 K_χ 存储对应密文到映射中;
⑭ for (int $i = 1$; $i \leqslant N$; i++)
⑮ $k_i \leftarrow g_i^{f(x)} \bmod q$; /*通过式(12)计算 */
⑯ 分发 k_i 和 C_M 到节点 i; /* 分发重加密子密钥 k_i 给代理节点 i */
⑰ end for
⑱ return “Update”;
⑲ end if
⑳ 根据 χ 中的键 K_χ 将映射中的 C_M 替换为重加密密文 F;
㉑ else
㉒ return “Reject”;
㉓ end if

算法 3 主要描述如下:

1)算法 3 将密文 C_M 和重加密凭证 γ 作为输入初始化 1 个字节类型数组 χ 用于存储 DU 信息和对应的重加密密文(行①)。如果 C_M 和 γ 都不为空,则检查集合 η 中每个重加密子密钥的有效性(行②~④)。如果检查到无效的重加密子密钥,算法 3 将返回“Reject”,表示重加密事务更新不成功(行⑤)。

2)验证完重加密子密钥之后进行映射的计算,根据式(15)计算 K_χ(行⑧),并在数组 χ 中查询 K_χ 的值是否存在(行⑨)。如果存在,则说明当前 DU 已经进行过重加密计算,因此,直接输出对应的重加密密文 F(行⑩),而无需再次进行重加密计算,以减少对同一 DU 进行访问控制的开销。

3)如果在数组 χ 中没有查到 K_χ 的值,则说明需要进行重加密事务的更新。首先,将密文 C_M 记录为数组 χ 中键 K_χ 对应的映射值(行⑫⑬)。其次,依次获取集合 η 中的每个重加密子密钥并分发给中继链上的代理节点,同时,要求代理节点进行重加密密文计算与更新(行⑭~⑯)。

4)将重加密密文 F 替换到数组 χ 中记录的键 K_χ 所对应的映射中,即替换所对应的密文,表明该用户的密文已进行重加密(行⑳)。如果 C_M 或 γ 有任一为空,则表示此次重加密更新不符合条件,算法 3 返回“Reject”(行㉒)。

综上所述,算法 3 中的时间复杂度受到循环中集合 η 中重加密子密钥的个数及代理节点数量 N 的影响,因此整体时间复杂度为 $O(|\eta| + N)$。

5 安全目标及安全性分析

5.1 安全目标

本文提出的机制中的 DO 和 DU 被认为是诚实的,能够安全保护它们的私钥。IPFS 能够提供安全、

分布式的数据存储服务。此外，由于智能合约是部署在可信的环境中，因此也可以诚实地执行预定义的程序。中继链上的代理节点是半诚实的服务器，一些代理节点可能存在非法获取数据的恶意行为。基于以上的假设，本文提出了 4 个安全目标：

1）数据解密正确性。在数据正确的前提下，DU 可以使用自己的私钥正确地解密并获取数据。

2）数据真实性和一致性。DO 如果生成有效的证明，并持有有效的数据分片标识，则数据的真实性和一致性都可以被正确验证。

3）防共谋及数据隐私保护。代理节点之间的共谋或者代理节点与 DU 的共谋都无法非法破解并获取 DO 所发送的数据。同时，他人无法从 DO 生成的零知识证明中获取任何隐私信息。

4）不可篡改。DO 发送的数据或者生成的零知识证明以及中继链代理节点的证明验证结果都不可被恶意篡改。

5.2 安全性证明

定理 1. IND-CPA 安全性。本文的代理重加密方法基于 DBDH 假设的 IND-CPA 是安全的，该方法满足选择明文攻击下的不可区分性。

证明. 假设存在 1 个任意多项式时间的对手 A 可以凭借不可忽略的优势 ε 来攻破本文的代理重加密方法，那么能构造 1 个挑战者 B 能以 $\varepsilon/2$ 的优势解决 DBDH 问题。假设对手 A 可以发起密钥查询和重加密查询，则对手 A 与挑战者 B 的交互如下：

1）Steup 阶段

给定 2 个阶为素数 q 的乘法群 G 和 G_T，群的生成元为 g，从而确定 1 个双线性映射 $e: G\times G\rightarrow G_T$。对于随机选择的 $a, b, c\in\mathbb{Z}_p$，$\rho\in\{0, 1\}$，给定输入的 DBDH 元组 (g, g^a, g^b, g^c, T)，当 $T=e(g, g)^{abc}$ 时，$\rho=1$；否则 $T=T'$ 且 $T'\in G_T$ 为随机时，$\rho=0$。挑战者 B 选择安全参数 θ 进行计算，并发布全局公共参数 $P_G=(G_1, G_2, g, r, e, p, q, H_1, H_2)$。

2）Query 阶段 1

挑战者 B 对于对手 A 发起的查询做出相应回复。

①密钥对查询。当对手 A 发起密钥的查询时，挑战者 B 选取 $\alpha\in\mathbb{Z}_p$，计算公钥 $\Pi=g^{\alpha}$，私钥 $\Sigma=\alpha$ 并将公钥 Π 发送给对手 A。

②重加密密钥查询。当对手 A 发起重加密密钥的查询时，对手 A 首先选取另一个委托人 C，询问的委托人 C 的公私钥对 (g^{β}, β)。其次，挑战者 B 选取随机数 $v\in\mathbb{Z}_p$ 并进行重加密密钥计算以得到对手 A 关于委托人 C 的重加密密钥 $R=g^{1/\alpha+\beta v}$，将其分割成 N 份重加密密钥份额，并输出其中 $t-1$ 份密钥份额 $\{k_1, k_2, \cdots, k_{t-1}\}$ 发送给对手 A。

3）Challenge 阶段

对手 A 发起挑战，选取另一个委托人 C，并选择 2 条等长的明文消息 $(M_1, M_2)\in M$ 发送给挑战者 B。挑战者 B 均匀随机选择 $d\in\{0, 1\}$ 并执行关于每个重加密子密钥的密文转换，得到每份重加密密文 $D_i^*=(M_dT^u, T^{\alpha u f(i)\lambda_i(x)})$。其中，$\lambda_i(x)=\prod_{j=1, i\neq j}^{k}\frac{x-x_j}{x_i-x_j}$，$u$ 为随机系数，通过组合满足阈值数量 D_i^*，得到重加密密文 $F^*=(g^{\alpha u}, M_dT^u, T^{\alpha u\left(\frac{1}{\alpha}+\beta v\right)})$，并将 F^* 作为挑战密文发送给对手 A。

4）Query 阶段 2

对手 A 向挑战者 B 再次发出查询，查询的过程与 Query 阶段 1 相同。

5）Guess 阶段

对手 A 猜测输出 $d'\in\{0, 1\}$。如果 $d'=d$，则挑战者 B 判断 $T=e(g, g)^{abc}$。在上述交互中，对手 A 的优势定义为 $ADV_A=|Pr[d'=d]-1/2|$。根据对手 A 不可忽略的优势 ε 可知，若 $T=e(g, g)^{abc}$，则对手 A 的优势为 $ADV_A=|Pr[d'=d\,|T=e(g, g)^{abc}]|=1/2+\varepsilon$。如果 $d'\neq d$，则挑战者 B 判断 T 是否等于 T'。此时，在随机猜测的情况下，对手 A 的优势为 1/2。

综上，挑战者 B 利用对手 A 解决 DBDH 问题的优势为 $\varepsilon'=1/2(|Pr[d'=d\,|T=e(g, g)^{abc}]|+|Pr[d'=d\,|T=T']|)-1/2=1/2(1/2+\varepsilon+1/2)-1/2=\varepsilon/2$，所以挑战者 B 能以 $\varepsilon/2$ 的优势解决 DBDH 问题，这与 DBDH 假设矛盾（该假设参见 2.2 节）。因此，本文提出的代理重加密方法在基于 DBDH 假设下是 IND-CPA 安全的。 证毕.

5.3 系统安全性分析

1）数据解密正确性。DU 收到加密的元数据后可以根据自己的私钥 b 解密并正确得到明文元数据 M，具体解密计算为：

$$M=\frac{MT^u e(g^{\sigma}, g^{\nu})^b}{T^{\sigma\left(\frac{1}{a}+\rho\right)}}=\frac{MT^u e(g, g)^{\sigma\rho}}{T^{u+\sigma\rho}}=\frac{MT^u e(g, g)^{\sigma\rho}}{e(g, g)^{u+\sigma\rho}}=\frac{MT^u}{T^u}=M\ 。$$

2）数据真实性和一致性。本文在数据验证阶段对证明进行验证的目标是验证数据的真实性。实现方式是：DO 通过基于 IPFS 存储数据后生成的内容标识符进行声明，并进行零知识证明的验证。在验证时，代理节点调用验证合约，根据 Fiat-Shamir 启发式，通过等式 $\omega_2=r^m \bmod p$ 验证证明。证明验证计算如下：

$$\omega'=r^{\omega_3}\omega_1^{y'}=r^{m-yH(\delta)}r^{H(\delta)\mathcal{H}(\omega_1, \omega_2)}\bmod p=r^m\bmod p=\omega_2。$$

如果上述计算成立,则代表结果正确,即验证了数据的真实性。除了验证数据的真实性,本文数据验证方法还能够验证数据的一致性。在数据的一致性验证方面,本文通过 IPFS 将数据以分片方式进行分布式存储,每个数据分片都有其唯一标识。用户在提供关于内容标识符的证明之外,还需要提供自己存储数据的分片标识。在中继链上验证分片标识时,只需分别计算 IPFS 中上传的分片标识哈希值与用户提供的分片标识哈希值,由于数据分片中任何内容的篡改都会导致分片标识哈希值发生变化,因此,如果双方提供的分片标识哈希值是相等的,则表明数据一致,从而反映用户没有篡改其提供的数据。

3)防共谋及数据隐私保护机制。一方面采用阈值代理重加密,基于 Shamir 秘密共享的思想,由中继链上的各个代理节点获取重加密密钥并进行单独的重加密过程,通过在有限域构造方程(如式(7)所示),任取 N 个数将 s 分成 N 份,使得每个节点只持有一部分而非完整的重加密密钥。本文机制至少需要 t 个节点的重加密子密钥才能根据拉格朗日插值公式恢复完整密钥。另一方面,由 DO 生成的零知识证明,中继链上的代理节点能获取到的信息只有 ω,根据离散对数求解困难问题,代理节点无法逆推出 $H(\delta)$ 或重构挑战 y,DO 不会泄露任何隐私信息。因此本文机制能够有效抵御共谋攻击和保护密文隐私。

4)不可篡改。DO 的零知识证明会被保存在源链上,同时 DO 的元数据密文发送并保存到中继链上。由于源链和中继链都是区块链,具有不可篡改的特点。源链和中继链数据或者交易都需要经过节点共识后才上链,同时,每一次关于密文的转换记录也会广播给链上的其他节点并记录在区块中,恶意节点无法轻易更改链上的数据,因为修改 1 个区块会导致该区块哈希变化,从而使得后续所有区块连接失效。因此,本文的机制满足不可篡改性。

6 实验评估

6.1 实验环境

本文采用 FISCO BCOS(版本 v2.9.0)作为区块链开发平台,并接入 WeCross 提供的跨链路由进行跨链业务的收集、监听与转发等操作。FISCO BCOS 区块链部署在 12th Gen Intel® Core™ i7-12700H@2.30 GHz 的中央处理器上,使用的操作系统为 Ubuntu 22.04。采用 Docker(版本 24.0.5)和 Docker-compose(版本 1.24.7)来构建和管理虚拟机环境。智能合约采用 Solidity 语言编写,数据采用 IPFS 进行存储,密码学算法采用 JPBC 库予以实现。

6.2 参数设置与对比方法

1)参数设置

本文的参数设置如表 3 所示,为测试密文数据大小对 IPFS 中存储读取的影响,在表 3 中,本文根据文献 [41] 的 IPFS 数据设置,将测试数据大小范围设置为 1~100 MB,具体分为 1 MB, 10 MB, 50 MB, 100 MB 这 4 种。其中,1 MB 数据约为 1 个传统的比特币区块大小,适用于小文件存储场景;10 MB 和 50 MB 通常用于常见的区块链交易记录或加密文件的存储场景;100 MB 通常用于大量文本或者图像音频等实用性存储场景。上述范围覆盖小型、中型及较大数据存储需求,能有效评估数据规模对 IPFS 存储读取的影响。

Table 3 Experimental Parameter Settings

表 3 实验参数设置

参数	描述	数值
Φ/MB	IPFS 测试数据大小	1, 10, 50, 100
N	重加密密钥份额总数	10, 20, 30, 40, 50
t	阈值	3, 5, 7, ⋯, 33
μ	阈值/份额之比	1/3, 1/2, 2/3
ζ	DU 的数量	2, 4, 6, ⋯, 20
T_N	跨链业务总量	5 000, 7 500, 10 000

表 3 中的重加密密钥份额总数及阈值设置参考了文献 [30],将份额总数范围设置为 {10, 20, 30, 40, 50}。在份额总数为 10 时,取阈值为 3, 5, 7;在份额总数为 20 时,取阈值为 7, 10, 13;在份额总数为 30 时,取阈值为 10, 5, 20;在份额总数为 40 时,取阈值为 13, 20, 27;在份额总数为 50 时,取阈值为 17, 25, 33。上述设置用于测试阈值/份额之比分别为 1/3, 1/2, 2/3 这 3 种情况,分别表示从低到高的不同容错情况下阈值代理重加密的时间开销。考虑到 FISCO BCOS 作为联盟链的特性,其网络环境通常由有限数量的参与方组成。因此,本文根据文献 [42],在 FISCO BCOS 初始化时用户默认数量为 4 的基准上,将 DU 的数量设定在 $\{2,4,\cdots,20\}$,以更贴近中小规模联盟链的实际应用场景,并模拟多用户环境下的访问控制机制。根据 FISCO BCSO 的性能指标和实际应用场景,根据文献 [43] 对业务数量的设置,以 10 000 为最大业务总量,将业务总量分别设置为 5 000, 7 500, 10 000,以分别测试低负载、中负载及高负载的跨链场景下访问控制的吞吐量以及时延表现。

2)对比方法

为了评估在多用户并发访问控制场景下的性能，将本文的阈值代理重加密方法与其他 3 种对比方法(PBTPRE[30], ABTPRE[31], IBTPRE[32])在加密时间、重加密时间、解密时间上进行比较。

1)PBTPRE。该方案提出了一种可变更代理份额的阈值代理重加密方法。为解决代理离开或加入网络时共享可能泄露的问题，该方案在不更改秘密的情况下定时变更代理的份额，以降低对代理的分布式共享进行攻击的可能性。

2)ABTPRE。该方案采用基于属性的阈值代理重加密，将代理重加密过程中加入单跳密钥策略属性，通过属性来主动更新代理的份额，以防止因代理的变更而导致重加密密钥份额的失效。

3)IBTPRE。该方案提出了一种基于身份加密的阈值代理重加密方法，使用身份加密替代阈值代理重加密的加密阶段，并通过全同态签名识别恶意代理。

6.3 实验结果分析

由于本文采用 IPFS 分布式地存储完整数据，因此数据的读写速率决定了获取数据的效率。为评估 IPFS 在不同大小的测试数据情况下的存储和读取时间开销性能，实验将测试数据 φ 分别设置为 1 MB, 10 MB, 50 MB, 100 MB, 实验结果如图 6 所示。

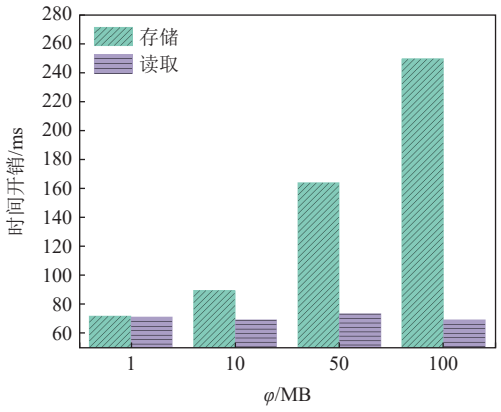


Fig. 6 Storage and reading overhead of IPFS

图 6 IPFS 的存储读取开销

图 6 表明，当测试数据文件越大，则需要更多的时间存储，但是读取数据增加的时间并不多，总体的读取时间都保持在较短的时间范围内。因此，本文用 IPFS 进行存储和读取的服务是可行并高效的。

由于本文数据的存储采用的是 IPFS+区块链混合存储方式，为了显示相较于传统链上存储的优势，本文实验对链上存储和混合存储这 2 种方式进行存储开销测试，实验结果如图 7 所示。

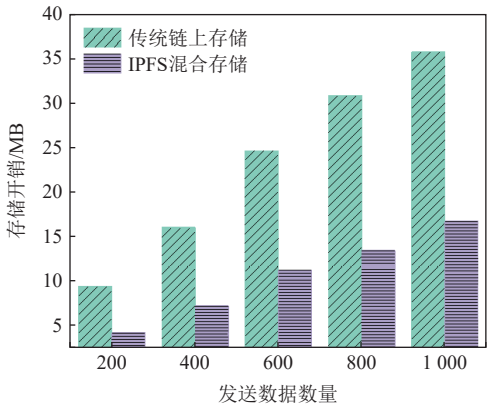


Fig. 7 Comparison of storage cost

图 7 存储开销对比

从图 7 可知，在同等数据数量下，混合存储的开销比传统的链上存储开销小很多。这是因为混合存储并不直接存储全部的密文数据，而是将关键信息进行存储，从而减少了区块链上的存储负载压力。此外，数据数量的增加，对链上存储的开销影响较大，而对混合存储的开销影响较小。混合存储相较于链上存储平均开销减少了 35%。

表 4 给出了数据验证中相关操作的时间开销和存储开销。其中，时间开销以及存储开销均为重复运行 20 次得到的结果。在表 4 中，由于 IPFS 返回的内容标识符固定为 46 B, 因此所生成的秘密消息的大小也固定。此外，从表 4 可知，数据验证主要的证明生成和验证分别仅需 0.35 ms 和 0.14 ms, 表明本文的数据验证方法是高效的。

Table 4 Time overhead and Storage Overhead for Data Verification

表 4 数据验证的时间开销和存储开销

操作	时间开销/ms	存储开销/B
内容标识符		46
数据分片标识 (单个)		46
秘密消息	0.27	111
挑战	0.04	9
证明生成	0.35	80
验证	0.15	

本文提出的 2 阶段的访问控制机制分别涉及证明生成、数据验证以及重加密更新 3 个算法。针对 3 个算法，本文相应设计了 3 种智能合约，并在以太坊提供的 Remix 编译器上进行编译，以测试部署所消耗的 Gas 数量。实验结果如图 8 所示。

由图 8 可知，证明生成合约、数据验证合约和重加密更新合约部署消耗的 Gas 数量都在可接受范围

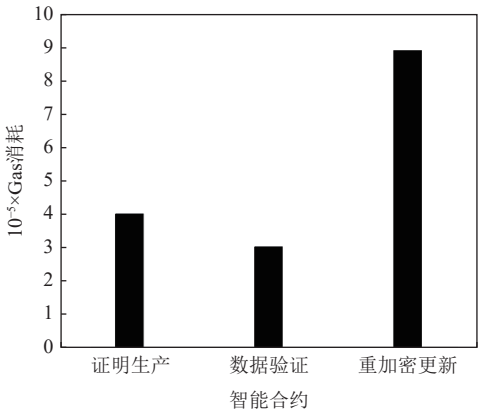


Fig. 8 Gas consumed by different contracts

图 8 不同合约消耗的 Gas

内。其中, 证明生成合约和重加密更新合约的开销较多, 是因为这 2 种合约属于事务调用, 会生成事务改变链上状态; 而重加密更新合约还涉及密钥管理、事务更新等复杂操作, 因此 Gas 消耗更多。

阈值代理重加密过程的性能与份额和阈值 2 个参数密切相关。DO 生成重加密密钥并将其分成 N 份, 由中继链代理节点收集并发布。代理节点需要根据收到的重加密密钥份额对密文进行部分重加密。当 N 个份额中的 t 个被正确计算完成后, 部分重加密的密文就可以组成新的重加密密文。

为了评估份额和阈值对于重加密过程中的时间开销影响, 本文分别设置总份额为 10, 20, 30, 40, 50, 并测试当阈值/份额之比 μ 分别为 1/3, 1/2, 2/3 时在中继链上进行重加密所花费的时间, 实验结果如图 9 所示。

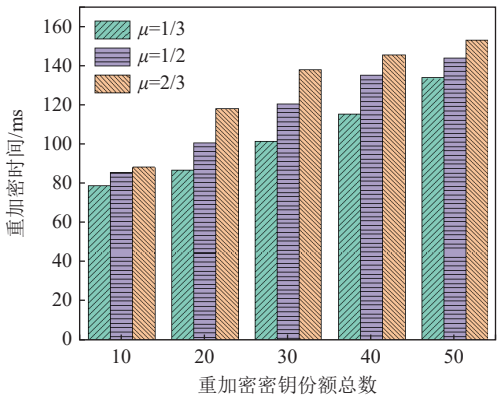


Fig. 9 Time cost of re-encryption

图 9 重加密时间开销

由图 9 可知, 重加密份额越多, 重加密所需要花费的时间就越多, 原因在于中继链上经过各个代理节点部分重加密后的密文需要组合成重加密密文。相比之下, 阈值的设置对于重加密的影响较小。

本文通过实验来比较本文方法与其他重加密方法的时间开销, 分为加密、重加密与解密 3 个步骤的对比。加密与解密实验的结果分别如图 10 和图 11 所示, 其中 DU 数量用 ζ 表示。从图 10 可知, 当 ζ 增加时, 由于 DU 相应地需要进行多次加密操作, 因此加密步骤的时间开销随着 ζ 而增加。在比较的方法中, IBTPRE 时间开销最多, ABTPRE 时间开销最少, 本文方法则均优于上述方法, 即使 ζ 增加到了 20, 加密步骤的时间开销也仅为 320 ms, 这是可以接受的。由图 11 可知, IBTPRE 的解密步骤的时间开销随着 ζ 的增加而增加, ABTPRE, PBTPRE 与本文方法在解密步骤中时间开销的变化较小, 本文方法还呈现出最低的解密时间开销, 低于 ABTPRE 与 PBTPRE。此外, 本文方法在 $\zeta=20$ 的情况下, 解密时间为 34 ms, 相较于其他方法更加高效。

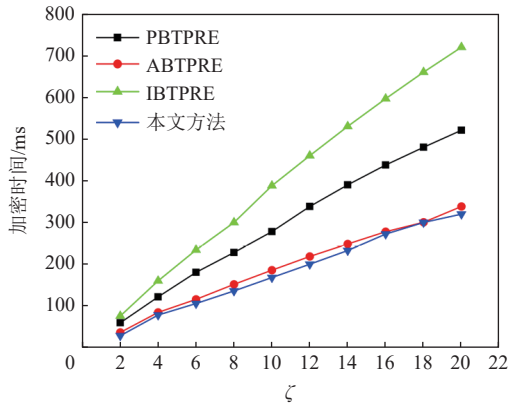


Fig. 10 Comparison of encryption time overhead

图 10 加密时间开销对比

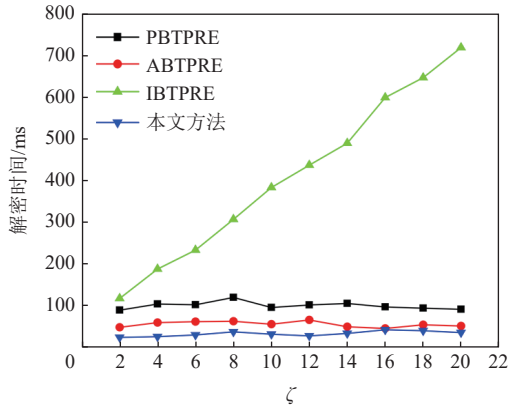


Fig. 11 Comparison of decryption time cost

图 11 解密时间开销对比

重加密步骤的对比如图 12 所示。总体上所有方法的重加密时间开销都随 ζ 的增加而增加。其中, IBTPRE 的重加密时间开销随 ζ 的增加变化最大, 而本文方法则呈现出最小的重加密时间开销的变化,

低于 ABTPRE 与 PBTPRE。当 $\zeta=20$ 时，本文方法的重加密时间开销相较于 ABTPRE 和 PBTPRE 分别降低了 34% 和 44%。尽管所有方法都需要多次重加密，但本文设计的重加密更新算法对重加密过程进行了改进，因此本文方法更具有优势。

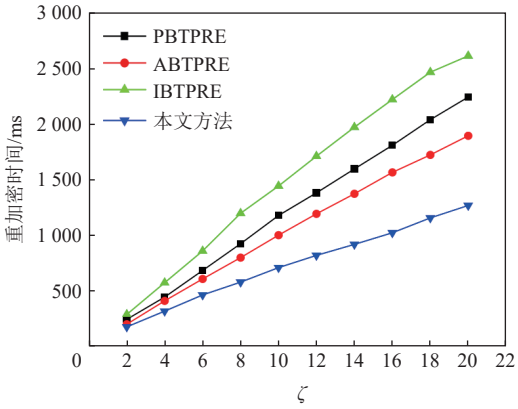


Fig. 12 Comparison of re-encryption time overhead

图 12 重加密时间开销对比

在本文的 2 阶段的访问控制机制中，使用 Java Demo SDK 对跨链业务进行吞吐量和时延测试。为了对比不同业务总量下的吞吐量与时延，本文将业务总量 T_N 分别设置为 5 000 笔、7 500 笔、10 000 笔。实验结果分别如图 13 和图 14 所示。

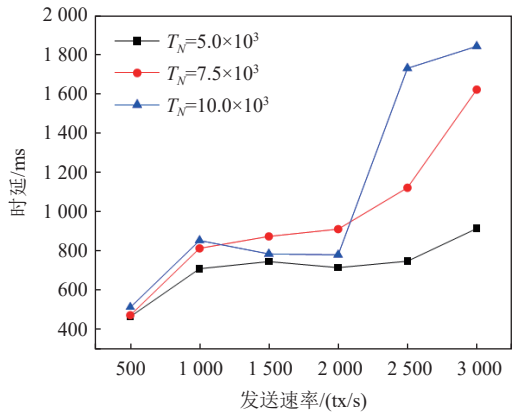


Fig. 13 Latency of cross-chain transaction

图 13 跨链业务的时延

图 13 和图 14 所反映出的时延和吞吐量随着发送速率以及总量的增加而增长。由于区块链网络状态的波动、业务处理的失败情况以及系统的处理能力变化，导致吞吐量偶有下降的情况，但总体仍呈现出上升的趋势。图 13 表明大多数情况下，在相同的业务发送速率下 T_N 越大则时延越大，这说明 T_N 是影响时延大小的因素之一。此外，发送速率的激增也会导致跨链网络的不稳定性，从而使得时延变化较

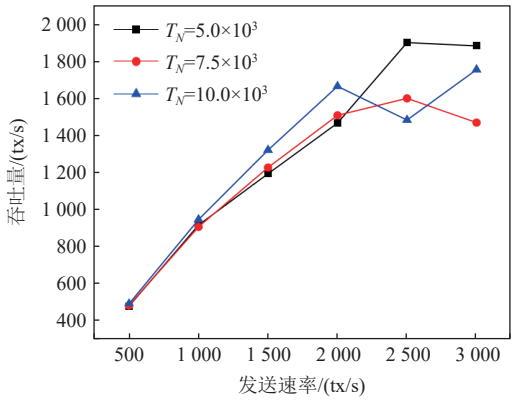


Fig. 14 Throughput of cross-chain transaction

图 14 跨链业务的吞吐量

大、波动较明显，但即使当业务总量为 10 000 笔且发送速率为 3 000 时，时延也仅约 1 841 ms。考虑跨链业务涉及多个链的状态更新和事务处理，因此该时延是可以接受的。吞吐量直接关系到访问控制机制在跨链场景下的业务处理能力，吞吐量越大，则处理业务的能力越强。图 14 表明本文的访问控制机制吞吐量同样也随着跨链业务总量和发送速率的增加而增加，但值得注意的是，T_N 的增加并没有显著提高吞吐量，而发送速率的增加则在一定程度上会相应地提高吞吐量。其中，当业务总量为 5 000 且发送速率为 2 500 时，吞吐量达到最高约为 1 903 tx/s，这说明在 T_N 不大但是发送速率较大的情况下，本文的访问控制机制处理能力相对较强，能有效提升吞吐量。

实验分析总结主要如下：

1）数据验证阶段分析。通过图 6 可知，基于 IPFS 的存储读取效率较高，且在不同数据大小的情况下读取开销均保持稳定；通过图 7 可知，本文采用的混合存储方式随着发送数据量的增长保持着较低的存储开销增长，在不同数据量的情况下存储空间占用远小于链上存储。通过图 8 以及表 3 可知，本文提出的 2 阶段访问控制机制中，数据验证阶段智能合约的设计以及具体验证的时间开销都较小，在证明的生成及验证方面耗时分别为 0.35 ms 和 0.15 ms，时间开销均较少。

2）代理重加密阶段开销分析。通过图 9 可知，所有方法的重加密时间均随着重加密密钥份额的增加而增加。由图 10~12 可知，本文提出的阈值代理重加密方法在重加密方面显著优于 IBTPRE，PBTPRE，ABTPRE 这 3 种对比方法，反映出本文在多用户场景下对重加密的改进是有效的。在加解密方面，本文方法总体时间开销小于 IBTPRE，PBTPRE，ABTPRE，但是与 ABTPRE 区别并不大。

7 结 论

本文提出了中继链环境下一种2阶段跨链访问控制机制,通过将密码学、区块链以及智能合约相结合,实现了跨链场景下安全、高效的访问控制。首先,在数据验证阶段设计一种数据验证方法用于验证跨链数据的真实性和一致性;其次,在代理重加密阶段设计一种可更新阈值代理重加密方法来改进重加密事务处理过程,以降低访问控制时间开销。通过安全分析可知,本文提出的访问控制机制可有效抵御安全威胁并实现设计目标。实验结果表明,本文机制的存储开销和在数据验证阶段的时间开销都较低,在代理重加密阶段的时间开销整体上均优于对比方法。

本文基于Fiat-Shamir启发式的数据验证方法对数据真实性与一致性进行了快速验证。未来研究可以引入基于电路设计的零知识证明技术,以验证数据的逻辑关系,从而拓展数据验证方法的应用场景。本文在代理重加密阶段降低了多用户访问控制下的时间开销,将来可以进一步研究降低加解密时间开销的方法。

作者贡献声明:张佩云和陶帅负责方案整体设计并撰写论文;陈健负责部分算法思路和实验方法;陈子寒参与论文校稿。

参 考 文 献

[1] Zhang Peiyun, Guo Weifeng, Liu Zijie, et al. Optimized blockchain sharding model based on node trust and allocation[J]. IEEE Transactions on Network and Service Management, 2023, 14(1): 2804−2816

[2] Zhang Peiyun, Liu Zijie, Zhou Mengchu, et al. A group-based block storage model with block splitting and unit encoding for consortium blockchains[J]. IEEE Transactions on Network and Service Management, 2023, 20(3): 2886−2897

[3] Li Zhenyu, Ding Yong, Yi Chen, et al. Blockchain data accelerated exchange method for data element circulation[J]. Journal of Computer Research and Development, 2024, 61(10): 2554−2569 (in Chinese)
(李振宇, 丁勇, 易晨, 等. 面向数据要素流通的区块链数据加速交换方法[J]. 计算机研究与发展, 2024, 61(10): 2554−2569)

[4] Shi Hongjian, Ma Ruhui, Zhang Weishan, et al. Semi-centralized federated learning framework based on blockchain[J]. Journal of Computer Research and Development, 2023, 60(11): 2567−2582 (in Chinese)
(施宏建, 马汝辉, 张卫山, 等. 基于区块链辅助的半中心化联邦学习框架[J]. 计算机研究与发展, 2023, 60(11): 2567−2582)

[5] Hu Jian, Zhu Peng, Li Juanjuan, et al. A secure medical information storage and sharing method based on multi blockchain architecture[J]. IEEE Transactions on Computational Social Systems, 2024, 11(5): 6392−6406

[6] Gupta D, Karati A, Saad W, et al. Quantum-defended blockchain-assisted data authentication protocol for Internet of vehicles[J]. IEEE Transactions on Vehicular Technology, 2022, 71(3): 3255−3266

[7] Han Xu, Tian Daxin, Zhou Jianshan, et al. Privacy-preserving proxy re-encryption with decentralized trust management for MEC-empowered VANETs[J]. IEEE Transactions on Intelligent Vehicles, 2023, 8(8): 4105−4119

[8] Chen Jing, Yang Hao, He Kun, et al. Current status and prospects of blockchain extension technology[J]. Journal of Software, 2024, 35(2): 828−851 (in Chinese)
(陈晶, 杨浩, 何琨, 等. 区块链扩展技术现状与展望[J]. 软件学报, 2024, 35(2): 828−851)

[9] Ren Yongjun, Lv Zhiying, Xiong Neal, et al. HCNCT: A cross-chain interaction scheme for the blockchain-based metaverse[J]. ACM Transactions on Multimedia Computing, Communications and Applications, 2024, 20(7): 1−23

[10] Pathak A, Anbagi I, Hamilton H. SATI: Sidechain-based access control & trust mechanism for IoT networks[J]. IEEE Transactions on Network and Service Management, 2024, 21(5): 5888−5903

[11] Xie Tianxiu, Gai Keke, Zhu Liehuang, et al. Rac-chain: An asynchronous consensus-based cross-chain approach to scalable blockchain for metaverse[J]. ACM Transactions on Multimedia Computing, Communications and Applications, 2024, 20(7): 1−24

[12] Zhang Yuchao, He Xiaofeng, Wang Xiaotian, et al. DeCross: Toward accountable and efficient cross-chain collaboration in IIoT[J]. IEEE Transactions on Industrial Informatics, 2025, 21(6): 4947−4957

[13] Luo Xinyi, Xue Kaiping, Sun Qibin, et al. CrossChannel: Efficient and scalable cross-chain transactions through cross-and-off-blockchain micropayment channel[J]. IEEE Transactions on Dependable and Secure Computing, 2025, 22(1): 649−663

[14] Zhang Kai, Tsai P, Tian Jiao, et al. DPNM: A differential private notary mechanism for privacy preservation in cross-chain transactions[J]. IEEE Transactions on Information Forensics and Security, 2025, 20: 2224−2236

[15] Jiang Jiajia, Zhang Yushu, Zhao Jiahao, et al. Towards efficient consistency auditing of dynamic data in cross-chain interaction[J/OL]. IEEE Transactions on Dependable and Secure Computing, 2025[2025-05-05]. https://ieeexplore.ieee.org/document/10854675

[16] Wang Haonan, Wang Jingyu, Liu Lixin, et al. Temporary relay: A more flexible way to cross chains[J]. Peer-to-Peer Networking and Applications, 2024, 17: 3489−3504

[17] Liu Weikang, Cao Bin, Peng Mugen, et al. Distributed and parallel

blockchain: Towards a multi-chain system with enhanced security[J]. IEEE Transactions on Dependable and Secure Computing, 2025, 22(1): 723−739

[18] Hu Xiaoxuan, Ling Yaochen, Hua Jialin, et al. A blockchain cross-chain transaction method based on decentralized dynamic reputation value assessment[J]. IEEE Transactions on Network and Service Management, 2024, 21(5): 5597−5612

[19] Guo Yihao, Xu Minghui, Cheng Xiuzhen, et al. zkCross: A novel architecture for cross-chain privacy-preserving auditing[C]//Proc of the 33rd USENIX Security Symp. Berkeley, CA: USENIX Association, 2024: 6219−6235

[20] Xue Lingyan, Huang Haiping, Xiao Fu, et al. A cross-domain authentication scheme based on cooperative blockchains functioning with revocation for medical consortiums[J]. IEEE Transactions on Network and Service Management, 2022, 19(3): 2409−2420

[21] Tao Yuechen, Li Bo, Li Baochun. On atomicity and confidentiality across blockchains under failures[J]. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(2): 766−780

[22] Ma Yuhang, Zhang Liang, Wu Xingyu, et al. Multi-party cross-chain transaction scheme based on distributed key generation and attribute-based encryption[J]. Journal of Computer Research and Development, 2023, 60(11): 2534−2544 (in Chinese)
(马宇航, 张亮, 吴星雨, 等. 基于分布式密钥生成和属性基密码的多方跨链交易方案[J]. 计算机研究与发展, 2023, 60(11): 2534−2544)

[23] Zhang Yushu, Jiang Jiajia, Dong Xuewen, et al. BeDCV: Blockchain-enabled decentralized consistency verification for cross-chain calculation[J]. IEEE Transactions on Cloud Computing, 2023, 11(3): 2273−2284

[24] Yang Fan, Qiao Yanan, Bo Junge, et al. Blockchain and digital asset transactions-based carbon emissions trading scheme for industrial Internet of Things[J]. IEEE Transactions on Industrial Informatics, 2024, 20(4): 6963−6973

[25] Wu Liqiang, Fu Shaojing, Luo Yuchuan, et al. A robust and lightweight privacy-preserving data aggregation scheme for smart grid[J]. IEEE Transactions on Dependable and Secure Computing, 2024, 21(1): 270−283

[26] Li Kunchang, Shi Runhua, Guo Wanpeng, et al. Dynamic range query privacy-preserving scheme for blockchain-enhanced smart grid based on lattice[J]. IEEE Transactions on Dependable and Secure Computing, 2024, 21(4): 1652−1664

[27] Yeh L, Hsu W, Shen C. GDPR-compliant personal health record sharing mechanism with redactable blockchain and revocable IPFS[J]. IEEE Transactions on Dependable and Secure Computing, 2024, 21(4): 3342−3356

[28] Li Xiaoxiao, Xie Yong, Wang Hui, et al. dAPRE: Efficient and reliable attribute-based proxy re-encryption using DAG for data sharing in IoT[J]. IEEE Transactions on Consumer Electronics, 2024, 70(1): 584−596

[29] Pei Hongmei, Yang Peng, Li Weihao, et al. Proxy re-encryption for secure data sharing with blockchain in Internet of medical Things[J]. Computer Networks, 2024, 245: 110373

[30] Raghav, Andola N, Katyayani V, et al. Proactive threshold-proxy re-encryption scheme for secure data sharing on cloud[J]. The Journal of Supercomputing, 2023, 79: 14117−14145

[31] Zhao Feixiang, Weng Jian, Xie Wenli, et al. HRA-secure attribute-based threshold proxy re-encryption from lattices[J]. Information Sciences, 2024, 655: 119900

[32] Wu Liqiang, Han Yiliang, Yang Xiaoyuan, et al. Identity-based threshold proxy re-encryption scheme from lattices and its applications[J]. Frontiers of Information Technology & Electronic Engineering, 2022, 23: 258−277

[33] Zhang Qingyang, Fu Yujie, Cui Jie, et al. Efficient fine-grained data sharing based on proxy re-encryption in IIoT[J]. IEEE Transactions on Dependable and Secure Computing, 2024, 21(6): 5797−5809

[34] Xue Wenyi, Yang Yang, Li Yingjiu, et al. ACB-Vote: Efficient, flexible, and privacy-preserving blockchain-based score voting with anonymously convertible ballots[J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 3720−3734

[35] Lipmaa H. Polymath: Groth16 is not the limit[C]//Proc of the 43rd Annual Int Cryptology Conf. Berlin: Springer, 2024: 170−206

[36] Liu Tianyi, Xie Tiancheng, Zhang Jiaheng, et al. Pianist: Scalable zkRollups via fully distributed zero-knowledge proofs[C]//Proc of the 46th IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2024: 1777−1793

[37] Attema T, Fehr S, Klooß M. Fiat–Shamir transformation of multi-round interactive proofs (extended version)[J]. Journal of Cryptology, 2023, 36(4): 1−45

[38] Song Wenpeng, Zhang Liang, Ma Yuhang. Auditable medical data sharing based on distributed digital identity[J]. Journal of Computer Research and Development, 2024, 61(10): 2501−2513 (in Chinese)
(宋文鹏, 张亮, 马宇航. 基于分布式数字身份的可审计医疗数据共享[J]. 计算机研究与发展, 2024, 61(10): 2501−2513)

[39] Kanth V, Hale B. Blockchain-based authenticated stego-channels and application to Ethereum[J]. IEEE Transactions on Dependable and Secure Computing, 2024, 22(1): 373−387

[40] Yang Zhimei, Peng Changgen, Zhong Chongyi, et al. Consortium blockchain private key protection scheme based on rational secret sharing and blockchain[J]. Computer Networks, 2024, 242: 110260

[41] Chen Zhuo, Zhu Liehuang, Jiang Peng. Tackling data mining risks: A tripartite covert channel merging blockchain and IPFS[J]. IEEE Transactions on Network Science and Engineering, 2025, 12(3): 1831−1848

[42] Wu Xiaohua, Yu Fangjian, Wang Jing, et al. Bpf-payment: Fair payment for cloud computing with privacy based on blockchain and homomorphic encryption[J]. Peer-to-Peer Networking and Applications, 2023, 16(5): 2649−2666

[43] Yang Yinghong, Bai Fenhua, Yu Zhou, et al. An anonymous and supervisory cross-chain privacy protection protocol for zero-trust IoT application[J]. ACM Transactions on Sensor Networks, 2023, 20(2): 1−20



Zhang Peiyun, born in 1974. PhD, professor, PhD supervisor. Senior member of CCF. Her main research interests include blockchain, cloud computing, edge computing, big data, trusted computing, service computing, and intelligent information processing.

张佩云, 1974 年生。博士，教授，博士生导师。CCF 高级会员。主要研究方向为区块链、云计算、边缘计算、大数据、可信计算、服务计算、智能信息处理。



Tao Shuai, born in 2000. Master. His main research interest includes blockchain.

陶　帅, 2000 年生。硕士。主要研究方向为区块链。



Chen Jian, born in 2000. Master. His main research interest includes blockchain.

陈　健, 2000 年生。硕士。主要研究方向为区块链。



Chen Zihan, born in 2002. Master candidate. His main research interest includes service computing.

陈子寒, 2002 年生。硕士研究生。主要研究领域为服务计算。