计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2024,33(9):105−113 [doi: 10.15888/j.cnki.csa.009608]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于同态加密的跨链交易数据隐私保护①

赵文静, 边根庆

(西安建筑科技大学 信息与控制工程学院, 西安 710399)
通信作者: 赵文静, E-mail: 1273037094@qq.com

摘　要: 为了解决区块链跨链交易数据隐私问题, 本文提出了一种基于同态加密的隐私保护方案. 该方案改进了同态加密算法以支持浮点数运算, 同时保留了原算法加法同态特性, 并支持任意次数的加法运算, 以实现对跨链交易金额的隐私保护. 为了防止同态加密的私钥管理不当或丢失对交易安全性构成威胁, 引入了基于 Shamir 秘密共享的私钥共享机制. 该机制通过增加 ECDSA 数字签名对私钥份额进行验证, 防止不可信节点发送恶意的值来恢复私钥, 同时考虑节点掉线或离开后私钥份额的动态更新, 从而防止节点串谋. 经过安全性分析和实验验证, 结果表明所提出的方案能有效保护跨链场景下的交易隐私.

关键词: 同态加密; 跨链; 交易隐私; 秘密共享; 中继链

引用格式: 赵文静,边根庆.基于同态加密的跨链交易数据隐私保护.计算机系统应用,2024,33(9):105–113. http://www.c-s-a.org.cn/1003-3254/9608.html

Privacy Protection Based on Homomorphic Encryption for Cross-chain Transaction Data

ZHAO Wen-Jing, BIAN Gen-Qing

(School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710399, China)

Abstract: To protect data privacy in blockchain cross-chain transactions, this study proposes a privacy protection scheme based on homomorphic encryption. The scheme improves the homomorphic encryption algorithm to support floating-point operations while retaining the additive homomorphic property of the original algorithm, and it supports any number of addition operations to realize the privacy protection of cross-chain transaction amounts. To prevent security threats to transactions posed by mismanagement or loss of the private key with homomorphic encryption, a private key sharing mechanism based on Shamir's secret sharing algorithm is introduced into the scheme. This mechanism prevents untrustworthy nodes from sending malicious values to recover the private key by adding ECDSA digital signatures to verify the private key share. It also considers the dynamic update of the private key share after a node drops or leaves to prevent node collusion. Security analysis and experimental verification show that the proposed scheme can effectively protect privacy in cross-chain transactions.

Key words: homomorphic encryption; cross-chain; transaction privacy; secret sharing; relay chain

区块链是一种去中心化、无需信任的分布式账本, 允许网络中所有节点通过共识算法和加密算法管理相同的数据[1]. 随着区块链技术的不断进步, 其已被广泛应用于商品溯源[2]、智能交通[3]、物流监管[4]等多个领域. 然而, 这些应用大多基于不同的场景和设计理念, 区块链之间的孤立性、封闭性以及链与链之间的高度异构化, 影响了数据流通和价值转移, 成为阻碍区块链技术广泛发展的瓶颈. 为了解决这一挑战, 跨链技术应运而生[5]. 这种允许不同区块链系统之间进行信息互通的技术, 为多个区块链平台之间的信息流动提供了可

① 基金项目: 陕西省重点研发计划 (2023-YBGY-021); 陕西省自然科学基础研究计划 (2021JLM-16)
收稿时间: 2024-03-02; 修改时间: 2024-04-01; 采用时间: 2024-04-19; csa 在线出版时间: 2024-07-24
CNKI 网络首发时间: 2024-07-25

能性, 使得多样化的服务得以跨平台提供[6]. 随着跨链技术的发展, 不同区块链系统之间信息互通的需求日益增长, 数据共享和和互操作性已成为常规需求[7]. 例如, 车联网需要车辆间的数据共享, 医疗领域要管理患者医疗数据, 金融领域需要构建更高效透明的支付系统以提升跨境支付和结算的速度, 客户信用评估需要数据计算等[8].

然而在实际的区块链应用中, 账本的公开透明性对链上的数据安全造成了威胁. 在跨链交互过程中, 因涉及多链数据传输与交换, 导致用户个人隐私及交易数据泄露的风险更高, 如何保护跨链数据隐私正成为研究热点之一. 按照康海燕等[9]提出的分类, 区块链的数据隐私主要包括身份隐私和交易隐私两种类型. 身份隐私指区块链系统中用户匿名性问题, 即用户身份信息与区块链账户地址的关联; 交易隐私则指存储在区块链的交易记录及相关信息, 包括交易金额和交易双方信息等. 在跨链系统中, 跨链交易涉及多个不同区块链网络, 不同链系统的架构和实现数据隐私的方式可能存在差异, 也增加了隐私保护问题的复杂性和难度. 本文着重研究跨链场景中的交易隐私问题.

针对上述问题, 学术界对跨链隐私的研究主要集中在同态加密、零知识证明、代理重加密等方面. Cai 等[10]提出了一种基于 Paillier 同态加密的跨链资产交易隐私保护新方法, 在隐私保护过程中利用 Paillier 同态加密代替哈希加密, 解决了整个跨链交易过程中身份隐私信息泄露的主要问题, 但对于跨链交易金额的隐私安全性有所欠缺. Yang 等[11]提出了一种基于零知识证明算法和混币技术的跨链隐私保护协议, 该协议改变了认证机制, 利用组合生成函数来映射交易中的虚拟外部地址, 允许实现快速跨链匿名交易, 但会面临中间人攻击的问题. 薛庆水等[12]提出了一种基于条件代理重加密的跨链数据共享方案, 通过随机选择的公证人执行条件代理重加密, 实现密文由提供者解密转变为请求者解密, 同时确保密钥由提供者生成 , 但存在一定的中心化风险. Yin 等[13]提出了 Bool 网络, 该平台利用多方计算来协作创建支持的每个区块链上的账户, 并利用阈值签名方案来联合签名跨链交易. 但可信执行环境的异构性意味着如果部分可信执行环境芯片遭受侧信道攻击, 仍然存在潜在的安全风险. Li 等[14]提出了一个旨在保证交换公平的密钥交换机制, 以及通过 CP-SNARK 验证机制来确保交易得到确认而不泄露交易细节的方法. 然而, 当一个链的隐私受到侵犯时, 可能会影响到其他链的隐私.

同态加密是一种能够在不解密数据的情况下对密文进行操作的加密技术, 从而对数据隐私起到重要的保护作用. 根据相关研究[15], 同时实现加法和乘法的全同态加密难度较大, 相比之下, 仅实现加法或乘法的其中一个性质的半同态加密算法具有更高的效率, 且适用范围更广. 目前, 常见的半同态加密算法包括 RSA、ElGamal 和 Paillier. 而 RSA 和 ElGamal 算法基于乘法同态性质, 这导致了验证交易后余额正确性方面存在一定局限性. 在区块链交易中通常需要验证交易余额, 这种验证涉及加法运算, 故选择具有加法同态属性的 Paillier 算法更为合适, 因此本文以 Paillier 算法作为跨链交易隐私保护方案的基础. 密码学算法的安全性取决于密钥管理的严谨性, 私钥的安全管理至关重要. 任何泄露或不当使用都可能导致严重的隐私数据泄露和系统安全威胁. 因此对密钥的安全管理和保护显得尤为重要.

目前主流的公证人机制、哈希锁定机制等跨链技术在实现区块链网络互操作性方面取得了进展, 但都存在局限性. 公证人机制是指引入一个或多个可信实体作为公证人, 以听取和响应链上事件. 可能产生中心化的信任问题和单点故障风险, 拥有较大的权力和控制权, 从而降低系统的公正性和可信度[16–18]. 哈希锁定机制旨在通过设置时间和解锁条件来实现跨区块链交易的安全性和原子性, 但缺乏跨链合约可能导致交易条件泄露, 存在超时和数据传输失败造成资产损失的风险, 所以它的使用场景有限[19]. 中继链则被视作是一种去中心化的公证人机制, 无需第三方介入, 通过链间通信协议和数据传输机制, 在保护用户隐私的同时, 保持交易信息的完整性和可验证性[20]. 通过利用中继链技术, 能够在不同区块链间安全地传输加密信息, 同时通过加密技术保障数据隐私[21]. 本文方案采用中继链进行跨链交互, 主要关注合约层, 通过跨链合约部署实现在链下加密. 跨链合约的主要功能是发起交易, 使用密文作为交易内容, 并通过嵌入智能合约形式构建跨链节点, 即使应用节点彼此不信任, 也能在跨链合约的控制下安全地进行数据交互[22].

基于上述分析, 本文提出了一种基于改进 Paillier 同态加密和中继链技术的跨链交易数据隐私保护方案, 旨在提高交易数据的隐私性和安全性. 主要工作如下.

(1) 对已有 Paillier 算法进行了针对浮点数的重新

设计. 提出一种支持浮点数的同态加密跨链交易数据隐私保护方案, 实现了对跨链交易金额的隐私保护.

(2) 设计了一种基于 Shamir 秘密共享的私钥共享机制. 针对改进的 Paillier 同态加密私钥管理不当或丢失问题. 该机制在实现对私钥份额抗篡改的同时, 有效防止了节点的串谋行为, 确保了同态加密私钥的完整性和安全性.

(3) 对所提出的方案进行了安全性分析和实验验证. 结果表明, 该方案能够有效保护跨链场景下的交易数据隐私.

1 预备知识

1.1 Paillier 同态加密

本文使用 Paillier 算法作为同态加密算法来设计方法. 相关的理论知识如下[23–25].

(1) 算法构成

1) 密钥生成. 随机选择两个大素数p,q, 且$p \neq q$, 满足$gcd(pq,(p-1)(q-1))=1$; 计算$n=p\cdot q$和最小公倍数$\lambda = lcm(p-1,q-1)$. 定义一个分式除法函数$L(y)=y-1/n$; 随机选择正整数$g=1+n\in Z_{n^2}^*$, 使$u=(L(g^x \bmod n^2))^{-1} \bmod n$. 存在公钥为$(n,g)$, 私钥为$(\lambda,\mu)$. 如果$p,q$长度相等, 则密钥生成步骤能够化简为:

$$g=1+n,\ \lambda=\varphi(n),\ u=\varphi(n)^{-1} \bmod n \tag{1}$$

2) 加密. 对于任意明文消息$m\in Z_n$, 选择随机数$r\in Z_{n^2}^*$, 计算密文:

$$c=g^m\cdot r^n \ (\mathrm{mod}\ n^2) \tag{2}$$

3) 解密. 输入密文$c\in Z_{n^2}^*$, 计算解密明文:

$$m=L(c^\lambda \bmod n^2)\cdot\mu \bmod n \tag{3}$$

注意解密使用$\lambda=lcm(p-1,q-1)$等价于使用p和q.

(2) 同态性

给定两个密文$c_1,c_2\in Z_{n^2}$, 其中,

$$c_1=Enc_{pk}(m_1),\ c_2=Enc_{pk}(m_2) \tag{4}$$

1) 定义密文之间的加法运算$\oplus$:

$$\begin{aligned} c_1\oplus c_2 &= c_1c_2 \bmod n^2 \\ &= (g^{m_1}.r_1^n \bmod n^2)(g^{m_2}.r_2^n \bmod n^2) \\ &= g^{m_1+m_2}\cdot(r_1r_2)^n \bmod n^2 \end{aligned} \tag{5}$$

因此$c_1\oplus c_2=Enc_{pk}(m_1+m_2 \bmod n)$.

2) 给定$a\in Z_n, c=Enc_{pk}(m)$, 定义随机数a与密文c的乘法运算$\otimes$:

$$\begin{aligned} a\otimes c &= c^a \bmod n^2 \\ &= g^{am}\cdot(r^a)^n \bmod n^2 \\ &= Enc_{pk}(a\cdot m \bmod n) \end{aligned} \tag{6}$$

因此, $a\otimes c=Enc_{pk}(a\cdot m \bmod n)$.

1.2 可验证随机函数

可验证随机函数 (verifiable random function, VRF) 是一种具有验证性质的随机数生成器, 是一个密钥相关函数, 将输入映射到一个随机的输出, 并且可以生成一个证明, 证明输出确实是由特定的输入和密钥生成的[13].

(1) 参数生成

合约中的参数生成阶段, 一个 VRF 系统需要生成一对公私钥. 合约中的公钥可以被外部验证者获取合约中使用$VRF.GenerateKeyPair()$生成一对公私钥:

$$VRF.GenerateKeyPair() \rightarrow (publicKey,\ privateKey) \tag{7}$$

(2) 输入与输出

对于合约中的输入 $input$, 通过 VRF 生成函数生成随机数和相应的证明. 其中, 生成函数接受输入和私钥, 返回随机数和证明:

$$VRF.Verify(publicKey,\ input, randomValue,\ proof) \rightarrow 0/1 \tag{8}$$

(3) 验证

外部验证者通过合约中的公钥、输入、生成的随机数和证明来验证随机数的有效性:

$$VRF.Verify(publicKey,\ input, randomValue,\ proof) \rightarrow 0/1 \tag{9}$$

1.3 Shamir 秘密共享

Shamir 秘密共享方案是一种安全的密钥分发方法. 其允许分享者将一个秘密分割成多个私钥份额分配给参与者. 阈值参数为(t,n), 其中t表示恢复秘密信息所需的份额, n表示参与者总数[26]. 具体步骤如下.

(1) 选择一个大质数p, 且$s<p$, 然后选择一个随机数a_0, 并生成$t-1$个不同的随机数$a_1,a_2,\cdots,a_{t-1}$, 用于定义一个$t-1$阶多项式:

$$f(x)=a_0+a_1x+a_2x^2+\cdots+a_{t-1}x^{t-1} \tag{10}$$

(2) 通过在多项式$f(x)$上选择n个点, 得到n个份额. 每个参与者i收到一个份额$(i,f(i))$, 其中i是参与者的编号.

(3) 在原始数据恢复过程中, 获得t个参与者保存的份

额. 通过拉格朗日插值算法恢复数据. 常数项即为秘密.

$$s=\sum_{j=1}^{t} f(j) \prod_{i\neq j,i=1}^{t} \frac{k-i}{j-i} \tag{11}$$

2 方案设计

在不同区块链系统之间进行交互时, 确保敏感数据不被泄露至关重要. 以金融领域为例, 当银行 A 需要从银行 B 中跨链获取客户的存款余额以了解客户在两个银行的总存款额时, 如何在不暴露敏感数据的情况下有效交互显得尤为重要, 因此本文提出了一种支持浮点数的 Paillier 同态加密隐私保护方案.

2.1 Paillier 算法的改进

Paillier 同态加密的加密过程通过模运算得到密文. 因此, 只能处理整数明文和密文, 而无法对浮点数进行加密. 为了符合银行交易中浮点数运算的特性, 本文提出了一种改进的 Paillier 同态加密算法, 保留了原算法的加法同态性质, 并支持浮点数的加法运算. 与传统的模运算不同, 本文方案支持使用任意次数的加法运算, 进一步提高了加密算法的灵活性和适用性.

对全局的公钥设置为 $PK_{\text{all}}=\{pk,g,h,r,L_t,M_t\}$, 生成步骤如下.

(1) 生成密钥过程

1) 将明文和密文划分为明文集合 m_1–m_k 和密文集合 c_1–c_k, 用加密算法得到 c_1,c_2, 其余的 c 通过运算得到.

2) 使用原 Paillier 算法生成公钥 $pk(n,g)$ 和私钥 $sk(\lambda,u)$.

3) 随机生成浮点数 g,h 作为明文 m 加密的底数. 为了方便大数测试, 设置 $r<0.00000000001$.

(2) 明文加密过程

1) 随机选择正整数 L_t, 确保 L_t 取值范围大于 100:

$$M_t\left(M_t=(p+1)\times\frac{L_t}{p}\right) \tag{12}$$

其中, p 为大于 2 的正整数.

2) 根据 VRF 生成干扰因子 t_i. t_i 是范围为 (L_t,M_t) 的可验证正整数.

3) 用 pk 对干扰因子 t_i 进行 Paillier 加密, 得到:

$$L_{c_i}=EncPaillier(t_i) \tag{13}$$

4) 获取浮点数 g,h 和 r 以及干扰因子 t_i, 得到:

$$R_{c_i}=g^{m_i\times r}h^{t_i} \tag{14}$$

5) 对明文 m_i 进行加密, 得到密文 c_i:

$$c_i=(L_{c_i},R_{c_i}) \tag{15}$$

(3) 密文解密过程

1) 用 sk 对 L_{c_i} 解密得到 t_i, 并用 VRF 对 t_i 进行验证:

$$\begin{aligned} &DecPaillier\left(L_{c_i}\right)\\ &=L\left(t_i^{\lambda} \bmod n^2\right)\cdot\mu\\ &=L\left(\left(g^{t_i}r^n \bmod n^2\right)^{\lambda} \bmod n^2\right)\cdot\lambda^{-1}\\ &=\lambda\cdot t_i\cdot\lambda^{-1}=t_i \end{aligned} \tag{16}$$

2) 使用浮点数 g,h 和 r 以及干扰因子 t_i, 对密文 c_i 中的 R_{c_i} 解密得到明文 m_i:

$$\begin{aligned} DecPaillier(R_{c_i})&=DecPaillier(g^{m_i\times r}h^{t_i})\\ &=\frac{\log_g\frac{R_{c_i}}{h^{t_i}}}{r}=m_i \end{aligned} \tag{17}$$

(4) 加法同态过程

1) Paillier 具备加法同态, 因此有:

$$EncPaillier(t_1)EncPaillier(t_2)=EncPaillier(t_1+t_2) \tag{18}$$

$$EncPaillier(t_1)EncPaillier(-t_2)=EncPaillier(t_1-t_2) \tag{19}$$

2) $R_c=g^{m\times r}h^t$ 是可以运算的, 所以得到:

$$g^{m_1\times r}h^{t_1}\times g^{m_2\times r}h^{t_2}=g^{(m_1+m_2)\times r}h^{t_1+t_2} \tag{20}$$

$$g^{m_1\times r}h^{t_1}\div g^{m_2\times r}h^{t_2}=g^{(m_1-m_2)\times r}h^{t_1-t_2} \tag{21}$$

3) 对密文 c_1 和 c_2 进行加法运算得到 c_3, 进行减法运算得到 c_4:

$$c_3=(L_{c_3},R_{c_3})=(EncPaillier(t_1+t_2),g^{(m_1+m_2)\times r}h^{t_1+t_2}) \tag{22}$$

$$c_4=(L_{c_4},R_{c_4})=(EncPaillier(t_1-t_2),g^{(m_1-m_2)\times r}h^{t_1-t_2}) \tag{23}$$

4) 对 c_3 中的 L_{c_3} 通过 Paillier 算法解密得到:

$$t_1+t_2=DecPaillier\left(L_{c_3}\right)=DecPaillier(t_1+t_2) \tag{24}$$

m_i 可以通过式 (17) 得到, 获取浮点数 g,h 和 r, 并将干扰因子 t_1+t_2 用于解密 c_3 中的 R_{c_3} 项, 得到:

$$\begin{aligned} DecPaillier\left(R_{c_3}\right)&=\frac{\left(\log_g\frac{R_{c_3}}{h^{DecPaillier\left(L_{c_3}\right)}}\right)}{r}\\ &=\frac{\log_g\frac{R_{c_1}}{h^{t_1}}+\log_g\frac{R_{c_2}}{h^{t_2}}}{r}=m_1+m_2 \end{aligned} \tag{25}$$

5) 同理, 对 c_4 中的 L_{c_4} 通过 Paillier 算法解密得到:

$$t_1 - t_2 = DecPaillier(L_{c_4}) = DecPaillier(t_1 - t_2) \quad (26)$$

并将干扰因子$t_1 - t_2$用于解密c_4中的R_{c_4}项, 得到:

$$DecPaillier\left(R_{c_4}\right) = \frac{\left(\log_g \frac{R_{c_4}}{h^{DecPaillier\left(L_{c_4}\right)}}\right)}{r}$$

$$= \frac{\log_g \frac{R_{c_1}}{h^{t_1}} - \log_g \frac{R_{c_2}}{h^{t_2}}}{r} = m_1 - m_2 \quad (27)$$

原 Paillier 算法是针对整数的, 本文方案为了支持浮点数运算, 使用浮点数g,h和r以及干扰因子作为加密的底数, 明文m_i被加密为$g^{m_i \times r} h^{t_i}$, 其中r是一个极小的小数, g,h是两个浮点数. 这一设计可以确保明文的小数部分和干扰因子被正确的加密到密文中, 从而支持浮点数运算.

在明文加密和解密过程中, 干扰因子扮演着重要角色. 干扰因子是一个随机生成的正整数, 在加密时与明文相结合, 使得同样的明文每次加密都会产生不同的密文, 增强了加密的随机性和安全性. 在解密过程中, 干扰因子用于解密密文的一部分, 同时确保只有持有正确的私钥才能解密出明文的小数部分. 干扰因子通过 VRF 生成, 保证了其随机性和可验证性, 使得外部验证者或智能合约可以验证其有效性, 从而提高了加密的安全性.

2.2 系统模型

方案涉及 3 种实体: 应用链、跨链网关和中继链. 应用链是跨链交易的主体区块链, 能够与其他应用链进行跨链交互, 执行特定的应用业务逻辑. 通过采用统一的跨链数据格式和标准化的跨链合约接入跨链交互中. 跨链网关负责在区块链网络中收集和传播跨链交易, 监控应用链和中继链上的跨链请求, 进行跨链交易转发, 支持应用链与中继链之间的消息交互. 中继链是跨链服务平台, 管理连接的应用链, 提供可信验证和可靠路由等服务, 以支持跨链交易的进行. 系统模型见图 1.

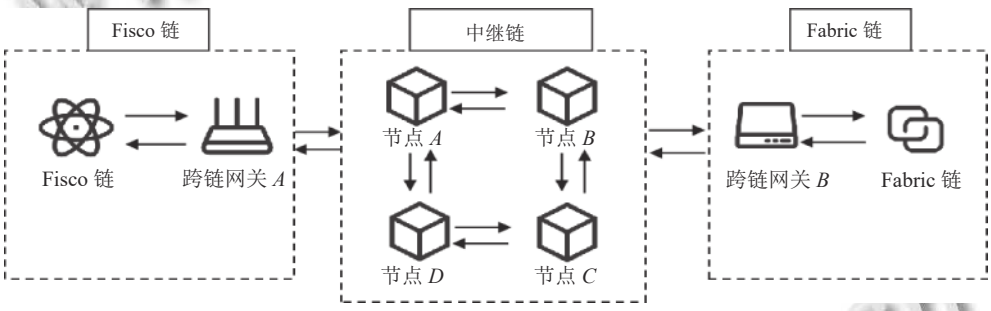

图 1 跨链系统模型

2.3 方案流程

在本文中银行 A 选择采用 Fisco 链, 银行 B 选择采用 Fabric 链. 本节给出了基于上述系统模型的隐私保护方案流程. 在第 3.2 节会比较两种区块链性能.

步骤 1: 每个银行部署自己的应用链.

步骤 2: 中继链部署.

步骤 3: 应用链跨链合约部署.

步骤 4: 启动跨链网关.

步骤 5: 在成功部署两条应用链后, 发起跨链交易请求. 以 id 为 appchain1 的 Fabric 链和 id 为 appchain2 的 Fisco 链为例, 银行 A 到银行 B 的交互过程如下.

(1) 银行 A 在 Fisco 链上通过 Paillier 算法生成的密钥, 加密用户 A 的金额信息 *amount*, 发起跨链交易 T_1 上传到 Fabric 链, 跨链网关 A 从密钥分发中心获得密钥, 并且将密钥发送给 Fabric 链的节点. 上链的基本信息包括交易哈希、交易状态、区块号、交易的返回信息、时间戳.

$$T_1 = set(Username_A, Enc_{\text{Paillier}}(amount_A)) = \{Hash, Status, BlockNumber, Receipt, Timestamp\} \quad (28)$$

同理, 银行 B 在 Fabric 链上发起跨链交易 T_2.

$$T_2 = set(Username_B, Enc_{\text{Paillier}}(amount_B)) = \{Hash, Status, BlockNumber, Receipt, Timestamp\} \quad (29)$$

(2) 银行 A 向银行 B 发起跨链交易 T_3, 即向中继链申请查看 Fisco 链的用户 B 的信息.

$$T_3 = get(BitxHub_{\text{id}} + Fisco_{\text{id}} + Fisco_{\text{broker}_{\text{address}}}, Username_B) = \{Hash, Status, BlockNumber, Receipt, Timestamp\} \quad (30)$$

(3) 银行 B 在收到银行 A 的请求消息后发起交易 T_4, 通过跨链网关向应用链的合约发送 *invokeInterchain* 方法获取用户 B 的数据. 上链的基本信息包括源链 id, 目标链 id, 交易序号, 交易类型, 调用函数名, 调

用参数, 交易状态, 是否加密.

$$T_4 = invokeInterchain(src_{\mathrm{id}}, dest_{\mathrm{id}}, index, type, callFunc, arg, txStatus, isEncrypt) = \{Hash, Status, BlockNumber, Receipt, Timestamp\} \tag{31}$$

(4) 银行 B 发起交易 T_5, 通过跨链网关 A 调用 Fabric 链合约的 *invokeReceipt* 存储数据. 上链的基本信息包括源链 id, 目标链 id, 交易序号, 交易类型, 调用的返回结果, 调用的状态.

$$T_5 = invokeReceipt(src_{\mathrm{id}}, dest_{\mathrm{id}}, index, type, result, txStatus) = \{Hash, Status, BlockNumber, Receipt, Timestamp\} \tag{32}$$

(5) 银行 A 通过 *getData* 方法发起交易 T_6, 获取用户 B 的加密数据.

$$T_6 = getData(Username_B) = \{Hash, Status, BlockNumber, Receipt, Timestamp\} \tag{33}$$

(6) 银行 A 对收到的密文进行解密, 并根据改进的 Paillier 加同态的性质, 对交易金额正确性进行验证, 即验证银行 A 的当前账户金额加上交易金额是否等于交易完成后的账户余额, 即式 (34) 是否成立.

$$Enc_{\mathrm{Paillier}}(Sum_A) = Enc_{\mathrm{Paillier}}(amount_A) + Enc_{\mathrm{Paillier}}(amount_B) \tag{34}$$

2.4 私钥共享机制

在跨链场景中, 中继链需要记录来自不同应用链的请求. 所有跨链交易对于已加入中继链跨链系统的应用链来说都是可见的. 如果应用链用户发起的跨链交易带有隐私数据, 隐私数据很容易泄露. 为防止中继链节点获取私钥并进一步获取交易详细信息, 本文提出在应用链上使用私钥对交易进行加密. 然而这种加密方式存在潜在的风险, 即私钥管理不当或丢失可能导致私钥被泄露, 从而威胁交易的隐私和安全性. 因此引入一种应用链私钥共享机制是必要的. 通过去中心化的方法共同维护同态加密的私钥, 可以有效防止私钥受到单个节点攻击的风险.

在本节中, 使用 Shamir 秘密共享方案来实现私钥共享. 考虑到节点的不可信, 有可能发送恶意私钥份额, 且节点可能因掉线或离开网络而导致私钥份额丢失. 为了防止串谋行为, 设置了更新 Shamir 秘密共享方案的机制. 通过这种方式, 即使部分节点受到攻击或处于离线状态, 仍能保证私钥的安全性和可靠性, 确保交易数据的完整性和隐私性. 应用链中的节点充当参与方, 假设应用链共有 n 个节点 AC_j $(j = 1,2,\cdots,n, j \neq n)$, 将 ID 设置为整数集合, $\{ID_1, ID_2, \cdots, ID_n\}$ 作为应用链节点的参与者编号.

步骤 1: 在系统初始化的过程中, 密钥生成中心生成私钥 $sk_{\mathrm{Paillier}} = (\lambda, \mu)$ 后给跨链网关. 跨链网关使用 Shamir 秘密共享方案将 λ 和 μ 分割成多份, 分发给不同应用链节点. 随机选择一个整数 a_0, 形成 $t-1$ 阶多项式, v 代表版本号由跨链网关存储, 其他节点无法获取.

$$\begin{cases} f_i^{\lambda}(x) = \lambda \times v + h(v) + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1} \\ h(v) = a_1 v + a_2 v^2 + a_3 v^3 \\ f_i^{\mu}(x) = \mu \times v + g(v) + b_1 x + b_2 x^2 + \cdots + b_{t-1} x^{t-1} \\ g(v) = b_1 v + b_2 v^2 + b_3 v^3 \end{cases} \tag{35}$$

其中, $f_i^{\lambda}(0) = \lambda \times v, f_i^{\mu}(0) = \mu \times v$. $f_i^{\lambda}(x), f_i^{\mu}(x)$ 和 $\lambda \times v, \mu \times v$ 需要保密, pk_{Paillier} 被公开.

为了应对可能的恶意节点尝试恢复私钥的情况, 增加了抗篡改机制. 通过验证私钥份额来确保在分发过程中私钥份额未被篡改. 具体而言, 跨链网关在分发过程中为每个私钥份额添加了一个 ECDSA 数字签名, 有助于阻止恶意节点对密钥进行篡改, 从而加强了系统对私钥管理过程的信任度.

步骤 2: 跨链网关对应用链节点 AC_j 计算私钥份额 $f_i^{\lambda}(ID_j), f_i^{\mu}(ID_j)$ 并将结果发送给 AC_j.

步骤 3: 如果发起的交易存在争议或私钥丢失, 跨链网关会执行恢复私钥 sk_{Paillier} 操作. 至少在应用链中的 k 个节点执行以下步骤.

(1) 跨链网关收到来自应用链的参与者 AC_j 发送的 $f_i^{\lambda}(ID_j)$ 和 $f_i^{\mu}(ID_j)$ 后, 首先通过 ECDSA 数字签名算法验证每个签名的正确性. 如果发现签名失败, 表示私钥份额可能被篡改或伪造, 将该节点移出节点群, 加入作恶名单, 同时向该应用链广播此消息. 在至少收到 k 个节点正确的私钥份额后, 重建私钥值 λ 和 μ:

$$\begin{cases} \lambda' = \dfrac{\sum_{j=1}^{t} f_i^{\lambda}(j) \prod_{i \neq j, i=1}^{t} \dfrac{k-i}{j-i} - h(v)}{v} \\ \mu' = \dfrac{\sum_{j=1}^{t} f_i^{\mu}(j) \prod_{i \neq j, i=1}^{t} \dfrac{k-i}{j-i} - g(v)}{v} \end{cases} \tag{36}$$

跨链网关由式 (36) 通过对私钥份额进行加权和, 重建同态加密私钥 $sk' = (\lambda', \mu')$, 这与标准的 Shamir 秘密共享方案有所不同.

(2) 在动态加入的情况下, 当新的参与者加入时,

无需修改现有参与者的私钥份额, 且秘密多项式$f_i^{\lambda}(x)$和$f_i^{\mu}(x)$保持不变. 选择一个新的随机点x', 计算$f_i^{\lambda}(x')$和$f_i^{\mu}(x')$生成新参与者的私钥份额, 这样可以减少对现有参与者的影响.

(3) 在动态离开的情况下, 对于参与者的离开, 若不重新计算其他私钥份额, 剩余参与者仍可以用原有的秘密多项式重新构建私钥. 但若离开的参与者私钥份额泄露, 整个私钥可能受到威胁. 此时更新版本号v. 假设原版本号为v_1, 新版本号为v_2. 对新加入的节点则重新计算版本号的$f_i^{\lambda}(x)$和$f_i^{\mu}(x)$; 对原有的节点, 将旧的版本号加上增量, 确保新旧节点之间的同步.

$$\begin{cases} f_i^{\lambda}(x)' = f_i^{\lambda}(x) + \lambda \times v_2 - \lambda \times v_1 + a_1 v_2 \\ \qquad + a_2 v_2^2 + a_3 v_2^3 - a_1 v_1 - a_2 v_1^2 - a_3 v_1^3 \\ f_i^{\mu}(x)' = f_i^{\mu}(x) + \mu \times v_2 - \mu \times v_1 + b_1 v_2 \\ \qquad + b_2 v_2^2 + b_3 v_2^3 - b_1 v_1 - b_2 v_1^2 - b_3 v_1^3 \end{cases} \tag{37}$$

3 方案分析

本文方案通过改进 Paillier 同态加密算法来支持浮点数运算, 对跨链交易金额进行加密, 保护跨链交易数据隐私. 同时基于 Shamir 秘密共享方案对改进算法的私钥进行管理, 增强了私钥的安全性. 本节对方案进行了分析和验证, 验证方案的安全性和可行性.

3.1 安全性分析

定理 1. 在改进的 Paillier 算法私钥未泄露的前提下, 攻击者不能对加密后的密文进行篡改.

证明: 改进的 Paillier 密文$(EncPaillier(t_i), g^{m_i \times r} h^{t_i})$, 解密过程中首先需要解出干扰因子$t_i$, 而$t_i$是通过原始 Paillier 算法公钥加密的, 原始 Paillier 算法的安全性基于n阶剩余类难题的判定, 在n分解未知的情况下, 破解 Paillier 加密后的密文难度相当于大整数的分解, 而大整数的分解非常困难的. 其次, 即使成功获得干扰因子t_i, 由$g^{m_i \times r} h^{t_i}$解出m_i的过程涉及离散对数难题, 而离散对数问题在椭圆曲线密码学算法中被认为难以解决. 最后, 如果篡改成功交易金额, 最终的同态加法特性验证交易金额的正确性将无法通过. 因此, 本文改进的 Paillier 算法是安全的.

定理 2. 如果应用链用户发起的跨链交易中带有隐私数据, 中继链节点不可能获取私钥并进一步获取交易详细信息.

证明: 在私钥共享机制中, 私钥首先通过 Shamir 秘密共享方案分割成多个私钥份额, 通过跨链网关分发给应用链节点AC_j, 每个私钥份额为$f_i^{\lambda}(ID_j)$, $f_i^{\mu}(ID_j)$, 这样即使部分私钥份额泄露, 也无法恢复原始私钥. 在私钥份额传输和验证过程中, 采用 ECDSA 数字签名算法对私钥份额进行签名, 以确保私钥份额在传输过程中不被篡改, ECDSA 数字签名算法的安全性基于椭圆曲线离散对数问题的难解性, 只有私钥份额持有者才能生成有效的数字签名. 在私钥重组过程中, 通过对私钥份额$f_i^{\lambda}(j)$和$f_i^{\mu}(j)$的加权和来重建密钥λ'和μ'. 若节点掉线或离开进行串谋, 重新计算私钥份额以确保新旧节点间的同步. 因此在应用链上, 私钥是安全的, 用于提交跨链交易之前进行加密和获取跨链交易之后进行解密. 由于中继链节点无法获取私钥, 因此可以确保跨链交易的隐私性.

3.2 实验验证

本文实验环境为 Intel(R) Core(TM) i5-7300HQ CPU, 内存为 16 GB 的 Windows 10 系统, 运行操作系统为 Ubuntu 20.04 LTS. 采用 Fabric 链和 Fisco 链构建底层应用链, 在腾讯云服务器进行部署. 采用 BitXHub 系统提供的跨链网关进行中继链和应用链之间通信.

本文方案评估跨链系统性能的指标包括同时执行操作的账户个数和改进 Paillier 算法的执行时间. 主要测试不同操作在跨链系统中所需的处理时间. 将原 Paillier 算法的执行时间作为基准, 对比本文方案中执行相同类型操作的 Paillier 算法所需时间. Paillier 算法的执行过程包括密钥生成、加密、密文操作和解密. 考虑到密钥生成在方案中运行次数有限, 因此不对其进行测试. 针对测试的银行数据最多为 12 位整数和 2 位小数, 根据位数和整数, 浮点数的区别, 进行了 1 000 次查询、加密和解密测试, 以评估加密, 解密和加法运算平均时间.

从图 2–图 4 可以看出, 随着整数位数增加, 整数平均加密, 解密和加法运算时间略有波动, 表明算法在处理不同位数的整数数据时能够保持稳定性. 由于本文方案改进的 Paillier 算法使用可验证随机函数 VRF 生成干扰因子, 并使用智能合约验证, 使得几乎所有阶段消耗时间都比原 Paillier 算法长, 但这种延迟时间完全在可接受范围内.

本文方案改进的 Paillier 算法不仅支持对整数进行同态加密, 还能直接对浮点数进行加解密和同态计算, 无需先将浮点数转换为整数再进行计算, 从而提高了运算效率. 由于不涉及模幂运算, 该算法支持任意次数的加法运算. 在进行多次浮点数的加法操作时, 算法

显示出高效性和稳定性. 根据图 5 所示的实验结果, 浮点数加密、解密和加法运算的平均运行时间并没有随着位数的增加呈现线性增长的趋势, 表明算法对于不同位数的浮点数计算效率相对稳定.

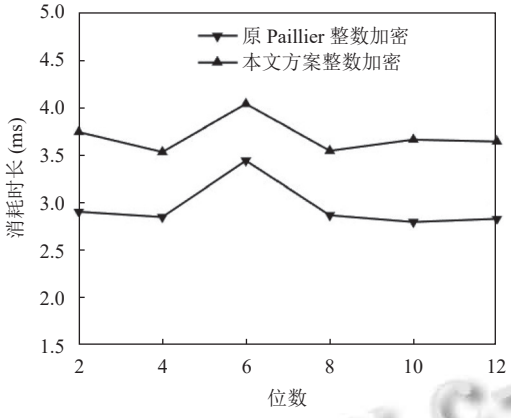

图 2 整数平均加密时间

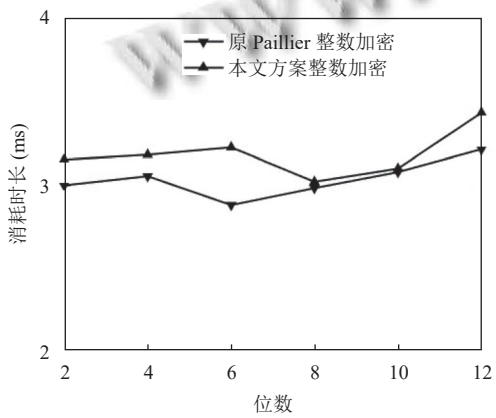

图 3 整数平均解密时间

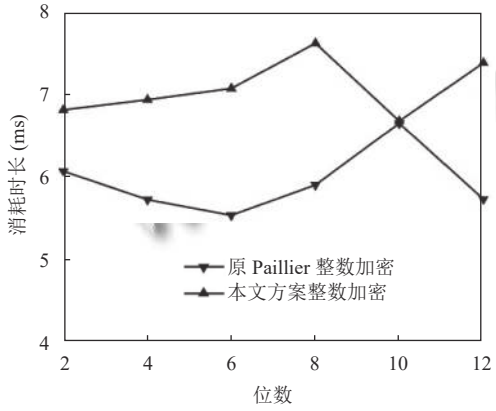

图 4 整数加法运算平均时间

本文方案采用 Fisco 链和 Fabric 链进行跨链测试. 如图 6 和图 7 所示, 随着账户个数的变化, Fisco 链上的 Paillier 算法在吞吐量方面优于 Fabric 链, 然而跨链交易的吞吐量较低且延迟较高. 随着账户个数的增加, 吞吐量下降, 延迟增加. 在跨链网络中, 跨链交易会引入一定程度的延迟, 但延迟时间完全在可接受的范围内. 总体而言, 本文方案表现出可用性和稳定性, 能够确保数据隐私的同时实现安全的跨链交互. 这使得银行可以获取必要信息而不泄露敏感数据.

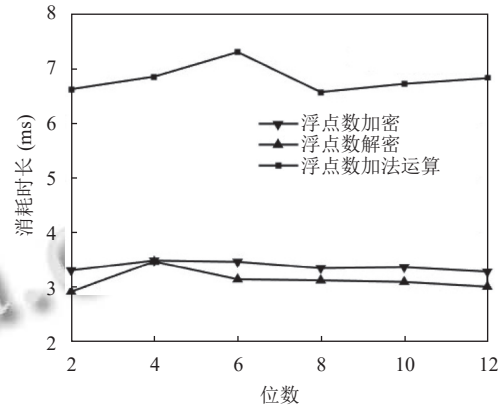

图 5 浮点数平均运行时间

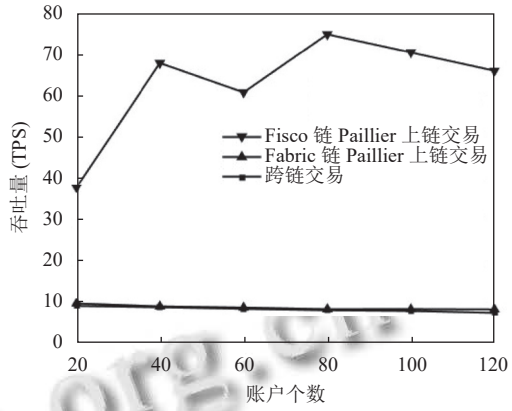

图 6 账户个数对吞吐量的影响

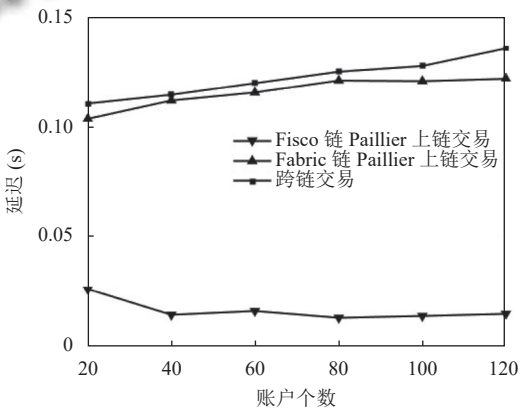

图 7 账户个数对延迟的影响

4 结论

本文提出了一个基于同态加密的跨链交易数据隐

私保护方案. 该方案采用支持浮点数加密的 Paillier 同态加密算法, 对跨链交易金额进行加密, 使跨链过程中的隐私数据以密文形式出现, 有效确保了交易数据在跨链过程中的隐私安全. 为了防止同态加密私钥管理不当或丢失的情况, 引入了基于 Shamir 秘密共享的私钥共享机制. 通过使用 ECDSA 数字签名验证私钥份额, 可以防止其在分发过程中被恶意篡改. 此外, 更新节点加入和离开的份额有助于防止节点串谋, 提升私钥管理的安全性. 安全性分析和实验验证表明, 所提出的方案有效保护了对跨链交易中的隐私数据.

参考文献

1 Huang HW, Kong W, Zhou SC, *et al*. A survey of state-of-the-art on blockchains: Theories, modelings, and tools. ACM Computing Surveys, 2021, 54(2): 44. [doi: 10.1145/3441692]

2 张帅, 项伟. 基于区块链对溯源数据的多方共享系统. 计算机系统应用, 2022, 31(6): 394–399. [doi: 10.15888/j.cnki.csa.008569]

3 穆蕾, 安毅生, 肖玉坤. 基于联盟区块链的电动汽车可信充电模型. 计算机系统应用, 2023, 32(2): 119–127. [doi: 10.15888/j.cnki.csa.008984]

4 肖梦雪, 左力, 徐志锟, 等. 基于区块链的铁路电子提单系统. 计算机系统应用, 2022, 31(9): 145–151. [doi: 10.15888/j.cnki.csa.008650]

5 李芳, 李卓然, 赵赫. 区块链跨链技术进展研究. 软件学报, 2019, 30(6): 1649–1660. [doi: 10.13328/j.cnki.jos.005741]

6 何全文, 林庆新, 林晖, 等. 基于跨链的医疗数据安全共享方案. 计算机系统应用, 2023, 32(5): 97–104. [doi: 10.15888/j.cnki.csa.009087]

7 曾诗钦, 霍如, 黄韬, 等. 区块链技术研究综述: 原理、进展与应用. 通信学报, 2020, 41(1): 134–151. [doi: 10.11959/j.issn.1000-436x.2020027]

8 祝烈煌, 高峰, 沈蒙, 等. 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]

9 康海燕, 邓婕. 区块链数据隐私保护研究综述. 山东大学学报 (理学版), 2021, 56(5): 92–110. [doi: 10.6040/j.issn.1671-9352.0.2020.595]

10 Cai JY, Zhou Y, Hu TY, *et al*. PTLC: Protect the identity privacy during cross-chain asset transaction more effectively. Proceedings of the 22nd IEEE International Conference on Software Quality, Reliability, and Security Companion. Guangzhou: IEEE, 2022. 70–78. [doi: 10.1109/QRS-C57518.2022.00019]

11 Yang YH, Bai FH, Yu Z, *et al*. An anonymous and supervisory cross-chain privacy protection protocol for zero-trust IoT application. ACM Transactions on Sensor Networks, 2024, 20(2): 32. [doi: 10.1145/3583073]

12 薛庆水, 孙晨曦, 马海峰, 等. 基于条件代理重加密的跨链数据共享方案. 计算机应用研究, 2023, 40(5): 1324–1329. [doi: 10.19734/j.issn.1001-3695.2022.09.0478]

13 Yin ZY, Zhang BS, Xu JZ, *et al*. Bool network: An open, distributed, secure cross-chain notary platform. IEEE Transactions on Information Forensics and Security, 2022, 17: 3465–3478. [doi: 10.1109/TIFS.2022.3209546]

14 Li YX, Weng J, Li M, *et al*. Zerocross: A sidechain-based privacy-preserving cross-chain solution for monero. Journal of Parallel and Distributed Computing, 2022, 169: 301–316. [doi: 10.1016/j.jpdc.2022.07.008]

15 Acar A, Aksu H, Uluagac A S, *et al*. A survey on homomorphic encryption schemes: Theory and implementation. ACM Computing Surveys, 2018, 51(4): 79. [doi: 10.1145/3214303]

16 蒋楚钰, 方李西, 章宁, 等. 基于公证人组的跨链交互安全模型. 计算机应用, 2022, 42(11): 3438–3443. [doi: 10.11772/j.issn.1001-9081.2021111915]

17 叶祥翩, 刘学业, 王斌辉, 等. 面向联盟链的分布式公证人跨链模型. 应用科学学报, 2022, 40(4): 567–582. [doi: 10.3969/j.issn.0255-8297.2022.04.003]

18 郭晓涵, 姚中原, 张勇, 等. 基于改进公证人机制的联盟链跨链隐私保护方案. 计算机应用, 2023, 43(10): 3028–3037. [doi: 10.11772/j.issn.1001-9081.2022111641]

19 Herlihy M. Atomic cross-chain swaps. Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing. Egham: ACM, 2018. 245–254. [doi: 10.1145/3212734.3212736]

20 Wood G. Polkadot: Vision for a heterogeneous multi-chain framework. White Paper, 2016, 21(2327): 4662.

21 叶少杰, 汪小益, 徐才巢, 等. BitXHub: 基于侧链中继的异构区块链互操作平台. 计算机科学, 2020, 47(6): 294–302. [doi: 10.11896/jsjkx.191100055]

22 郑建辉, 林飞龙, 陈中育, 等. 基于联盟自治的区块链跨链机制. 计算机应用, 2022, 42(11): 3444–3457. [doi: 10.11772/j.issn.1001-9081.2021111922]

23 张学旺, 黎志鸿, 林金朝. 基于公平盲签名和分级加密的联盟链隐私保护方案. 通信学报, 2022, 43(8): 131–141. [doi: 10.11959/j.issn.1000-436x.2022162]

24 肖瑶, 冯勇, 李英娜, 等. 基于同态加密的区块链交易数据隐私保护方案. 密码学报, 2022, 9(6): 1053–1066. [doi: 10.13868/j.cnki.jcr.000567]

25 刁一晴, 叶阿勇, 张娇美, 等. 基于群签名和同态加密的联盟链双重隐私保护方法. 计算机研究与发展, 2022, 59(1): 172–181. [doi: 10.7544/issn1000-1239.20200576]

26 Ma ZF, Wang JY, Gai KK, *et al*. Fully homomorphic encryption-based privacy-preserving scheme for cross edge blockchain network. Journal of Systems Architecture, 2023, 134: 102782. [doi: 10.1016/j.sysarc.2022.102782]

(校对责编: 孙君艳)