基于区块链的可审计隐私保护机密交易

盖珂珂[1],陈思源[2],祝烈煌[1]
(1. 北京理工大学网络空间安全学院,北京 100081;2. 北京理工大学计算机学院,北京 100081)

摘 要: 隐私保护和交易数据审计是区块链系统相关方较为冲突的两个需求. 比特币采用未花费交易输出(Unspent Transaction Output,UTXO)的方式,保证用户能迅速查询到每笔交易的资金来源和去向,具有天然的可溯源性,确保了资金不会被"双花". 然而,由于每笔交易的交易金额及交易双方的地址都公开存储于账本,用户的交易行为变得公开可追溯,导致用户面临隐私风险. 对交易数据进行加密是一种简单有效的隐私保护手段,但也给交易的验证和审计带来不便. 本文提出了一种可审计的隐私保护机密交易方案,利用Pederson承诺实现交易合理性的公开可验证而不泄露交易的具体金额;支持交易发起方独立发起交易而无需经过接收方许可,与其他需要交易双方进行通信的机密交易方案相比,更符合实际情况同时节约了通信开销;引入陷门机制,账本和监管方外其他用户无法辨认交易发起方身份,保护了用户身份隐私;实现了多种审计功能,并根据监管方和私人审计者给出不同的审计方式;本文给出了一种新的范围证明方法,在适用于大数时较Prcash具有一定优势:对于256位大数的范围证明生成时间与Prcash基本相同,对于512位大数的范围证明生成时间节省29.78%,对于1 024位大数的范围证明生成时间节省56.86%.

关键词: 可审计;零知识证明;Pederson承诺;同态加密;范围证明
基金项目: 国家重点研发计划(No.2021YFB2701300)
中图分类号: TP309.2 **文献标识码:** A **文章编号:** 0372-2112(2025)02-0460-14
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20221020

Blockchain-Based Privacy-Preserving Auditable Confidential Transaction Scheme

GAI Ke-ke[1], CHEN Si-yuan[2], ZHU Lie-huang[1]
(1. *School of Cyberspace Science and Technology*, *Beijing Institute of Technology*, *Beijing* 100081, *China*;
2. *School of Computer Science and Technology*, *Beijing Institute of Technology*, *Beijing* 100081, *China*)

Abstract: Most current blockchain systems can hardly concurrently satisfy requirements of privacy protection and transaction data auditing. Bitcoin adopts the method of unspent transaction output (UTXO) to ensure that users can quickly query the source as well as fund destinations of each transaction, in order to avoid double spending threats. However, the users' behaviors, deemed to be privacy, maybe traced by adversaries, since transactions with addresses are stored in the ledger publicly. Even though encryption-based solutions are widely adopted, it often causes restrictions to transaction verifications and auditing. In this paper, we propose an auditable privacy-preserving confidential transaction scheme, which uses Pederson commitment to realize the public verifiability of the transaction rationality without disclosing the specific amount of the transaction. Our scheme supports the initiator of the transaction to initiate the transaction independently without permissions from the receiver, which saves the communication cost comparing with other confidential transaction schemes. By introducing the trapdoor mechanism, the identity of the transaction initiator cannot be recognized by other users outside the ledger and the supervisor, so as to protect users' privacy. It has realized a variety of audit functions, and different audit methods have been developed according to regulators and private auditors. This paper presents a new range proof method, which has advantages over Prcash when applied to large numbers. The generation time of range proof for 512 bit large numbers is shortened 29.78%, and the generation time of range proof for 1 024 bit large numbers is reduced 56.86%.

Key words: auditable; zero-knowledge proof; Pederson commitments; homomorphic encryption; range proof
Foundation Item(s): National Key Research and Development Program (No.2021YFB2701300)

收稿日期:2022-09-05;修回日期:2024-07-13;责任编辑:王天慧

1 引言

近年来,受比特币[1]及相关区块链技术发展的影响,世界各国尤其是世界主要经济体纷纷将区块链作为国家发展战略,各国央行在区块链上发行法定货币的想法也越来越受市场欢迎[2~9]. 作为区块链实现的法定货币可以为社会提供多种好处,包括降低高昂的现金生产处理成本,以匿名交易替代信用卡支付等非匿名数字支付以改善用户隐私,以密码学技术保证银行不能篡改用户数据以提升用户信任度等. 金融业、制造业等相关行业也尝试引入区块链以改善传统的支付体系[10~15].

但是,以比特币为蓝本的电子支付系统存在诸多隐私缺陷. 由于交易记录存储在区块链中并支持公开访问,交易金额公开可见,重要交易(如大额交易)可能与现实事件产生关联,暴露真实交易内容. 此外,通过追溯某一账号的交易记录,可以分析账号所有者的个人行为等敏感知识(例如,购物习惯可以反映用户收入情况和生活爱好),并映射推理个人真实身份. 对于金融领域,交易记录不仅是重要敏感的信息,同时也是宝贵的数据资源,交易记录直接公开于区块链上会给企业带来直接或间接损失,必须采取措施防止非授权用户查阅[16]. 因此,区块链系统难以替代传统的支付体系,须针对原有区块链系统的交易隐私加以改进.

具体地,交易隐私主要聚焦两个问题:(1)匿名性,即隐藏交易中发送方和接收方的身份;(2)机密性,即隐藏交易金额. 匿名性的实现方法通常采用椭圆曲线签名算法将公钥地址作为用户的交易账号以替代用户真实姓名. 机密性的实现可以采用密码算法对交易数据进行加密,只有拥有解密密钥的用户才能查询交易数据. 但是,加解密高额的计算开销与密钥托管等问题难以在大规模支付系统中应用.

一方面,区块链系统替代传统的支付体系须加强隐私保护;另一方面,保护交易隐私又会引入新的问题,即监管和审计问题[17]. 数字货币因匿名特性难以被监管,常被用于洗钱等经济犯罪,影响了数字货币的健康发展. 在此基础上,进一步保护交易隐私进一步增加了监管难度. 因此,如何在保护交易隐私的同时,实现监管,成为研究区块链在支付体系应用中的重点和难点问题.

事实上,保护交易隐私和实现监管是相对冲突的要求. 目前大部分区块链交易技术只提供了其中的一部分. 例如,使用明文身份和金额的交易处理速度快,易于监管,但不提供隐私保护. 化名的使用与比特币交易类似,改善了用户隐私,但导致监管失效. Confidential Asset[18]和 Mimblewimble[19],利用加密承诺加强隐私保护. 这种交易允许隐藏支付身份和价值,并容易混合交易,但不受监管. 姜轶涵等[17]利用 Pailliar 同态加密和 Bulletproofs[20]聚合范围证明构造了一种 ACT(Auditable Confidential Transaction)可审计的机密交易方案,提高了交易创建和验证效率,但面临加解密计算开销大及密钥托管等问题.

本文面向联盟链,针对机密交易中隐私保护和监管审计难以权衡和兼顾的问题,采用 Pederson 承诺和零知识证明等密码技术构造交易,在保护交易隐私的同时提供审计功能. 即在公开可见的账本中,用户除自身发起的交易外,无法识别其他交易的有用信息(如交易金额、交易双方身份等),通过构造陷门使监管者能识别交易用户身份并实现审计,平衡了交易隐私保护与交易监管,为受信任的权威机构在区块链上记录组织内部交易以及类似的应用场景提供一种可能的解决方案.

本文的主要贡献在于:

(1)本文给出的方案有效地保护了交易隐私,对于一般用户(非监管方)既不泄露交易双方身份,也不泄露交易金额;

(2)本文通过构造陷门在保护交易隐私的情况下提升了交易可监管性,实现了资产、收入、支出等多种审计功能;

(3)本文给出的方案使监管方能够区分银行在交易中的角色(即某一银行在交易中属于发起方、接收方或是不参与交易),进而可以实现多种平均值审计,这是其他使用 Pederson 承诺构造的机密方案所不具备的;

(4)本文提出的范围证明方法在适用于大数时生成证明所需时间具有一定优势.

2 相关工作

多数区块链支付系统以比特币为蓝本进行改造,但类比特币的区块链支付系统既面临隐私泄露风险,也缺乏相关监管手段,因此许多科研人员为实现既能提升区块链交易隐私又能支持有效权威监管的区块链支付方案进行了大量研究.

混币方案 CoinJoin[21],它允许用户相互组合交易,模糊某一具体交易从输入到输出的映射关系,一定程度上保护了交易隐私. 然而,因为交易金额是暴露的,除非所有的交易输出都相同,否则统计意义下仍然是可区分的. Confidnetial Asset[18]和 Confidential Transaction[22]对比特币交易进行了改进,它们盲化了交易中的资产和金额,参与者可以对交易进行验证,但仍然暴露了交易关系图,并且不支持私人审议,审计师需要访问所有交易才能确保交易完整性. Corda[23]和 Digital Asset Holding 方案中的 Global Synchronization Log(GSL)[24]是依赖可信第三方分布式分类账,在 Corda 中,公证

员验证交易并维护参与者的隐私，而GSL对其分类账进行分段，仅全局存储值的散列，并限制对细粒度交易数据的访问，但两种方案都不支持私人审计.

简明非交互的零知识证明ZK-snarks[25]是ZCash[26]的基础，它提供了很强的机密性，然而，它的生成非常“昂贵”，创建Zcash交易需要几分钟时间，并且需要下载整个分类账，虽然这对于执行不频繁交易的单个客户端是可行的，但在银行中介系统中使用ZK-snarks的开销是难以令人接受的，也难以在资源受限的移动设备上运行. ZK-snarks还需要饱受争议的可信设置，引入金融机构不愿接受的工程复杂度和加密难度[27].

Solidus[28]提出了一种用于公有链上的机密交易协议，通过链上交易进行资产转移结算. Solidus的运营框架基于真实世界的金融机构：少数银行维护着大量用户账户. 在这个框架内，Solidus隐藏了交易金额和交易关系图，引入了可公开验证的不经意RAM机器，实现公开可验证性. 虽然这种结构还提供私人审计，但Solidus只能通过将系统中使用的所有密钥透露给审计师并解密交易来支持审计.

Prcash[18]提出了一种新颖的监管机制，使用基于承诺的Mimblewimble作为解决方案的出发点，采用一种零知识证明结构为交易增加了监管功能，同时具备匿名性和较高的效率，并通过修改交易创建协议来改善Mimblewimble的隐私. Prcash利用Pederson承诺构造的交易方案可以在不泄露交易金额的情况下进行线性运算，保护交易隐私并支持一定的审计功能，但Prcash引入了用户间的交互，增加了通信开销. ZkLedger[29]在提供私人审计的同时实现了与Solidus类似的性能. Zkledger不需要可信设置，只依赖于广泛使用的密码学假设. 它使用了一种新型的分类账结构，使用承诺隐藏交易金额并利用同态性进行线性运算，实现了交易公开可验证并防止银行隐瞒交易，参与者还可以使用滚动缓存快速生成和验证交易.

Prcash和Zkledger同其他已知Pederson承诺构建的机密方案一样引入了别的问题：由于交易金额v是在模N的交换群中参与运算的，即① 对于金额为v的交易，计算承诺时交易发起方金额设置为$-v$, 接收方金额设置为$+v$，由于$-v$在模N群运算中对应另一个正整数$N-v$，系统无法通过金额的正负区分交易发起方和接收方，需引入其他手段进行区分. 普遍的做法是采用数字签名，交易发起方使用私钥对交易进行签名，这样既区分了发起方和接收方，也避免了用户资产被其他用户盗取. 但由于签名者的公钥是公开的，交易记录也公开存储于账本中，所有人都可以对交易进行验证，这种方式无意中暴露了交易发起方的身份，削弱了机密性，部分泄露了交易关系图；②v和$v+n\cdot N\,(n=1,2,3,\cdots)$在模$N$意义下计算的承诺值相同，因此发起交易时需引入范围证明用于说明$v\in[0,N-1]$，交易发起方可以独立发起交易的方案中，交易发起方还需生成其他所有银行交易金额的范围证明，这种做法产生了很大的计算开销.

3 整体架构

3.1 系统组成

本系统主要由监管方、银行、账本组成，具体结构如图1所示.

(1)监管方. 监管方用Auditor表示，能与银行交互，询问银行必要的私有数据并进行审计，如询问某银行某一时刻的资产情况，并与账本数据进行比对验证，实现对系统交易的监管.

(2)银行. 银行是交易的主体，用Bank表示. 银行Bank_i的资产用asset_i表示. 银行Bank_i拥有密钥对pk_i和sk_i.

(3)账本. 账本用Ledger表示，银行将交易请求提交到账本，账本验证后将交易打包上链并全局广播，此时银行即可认为交易被许可，随后完成转账(一般为链下行为，不具体讨论). 账本可采用常用的联盟链共识机制，如实用拜占庭容错(Practical Byzantine Fault Tolerance，PBFT)等，在全局范围内对所有有效交易进行排序. 账本可以由可信第三方维护，也可以由监管方自行维护. 本文假定账本诚实且无差错地记录交易，不考虑纠错的情况.

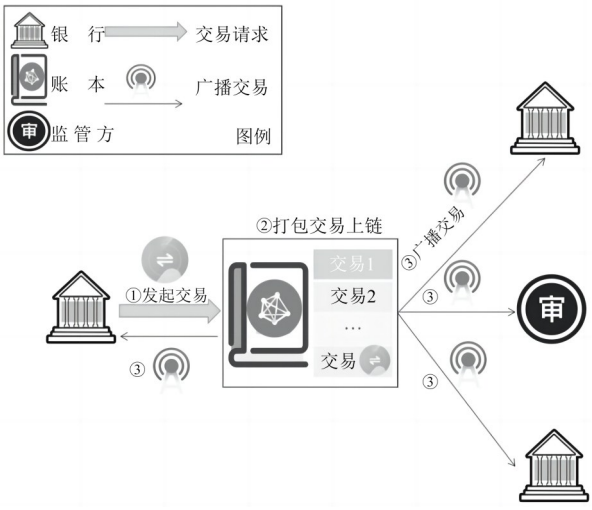

图1 系统示意图

该系统主要有以下操作：

(1)交易. 交易由银行发起，本文的交易特指银行转账，交易只能实现资产的转移，不能凭空创造或者销毁资产，银行发起的一笔交易可以有多个接收方，为了简洁，假设交易只有一个发起方和一个接收方. Bank_i

向 Bank_j 转账金额 v,则 Bank_i 的资产 asset_i 减少 v,Bank_j 的资产 asset_j 增加 v.

(2)审计. 审计由监管方发起,对银行交易进行审查和监督,监管方可以询问银行经营交易情况,如 t_0 时刻的资产情况或 t_1~t_2 时段银行的收支状况,银行对询问回复结果并给出证明数据,监管方将银行答复和账本记录进行对比,判断银行给出答案是否正确. 为了保护交易隐私,本方案中银行在接受审计时无须向监管方透露具体交易的交易金额.

3.2 密码组件

离散对数难题(Discrete Logarithm Problem, DLP). 给定 p 阶有限循环群 G,G 的一个生成元 g 和群中一个元素 $y\in G$,求解 $x\left(0\leqslant x\leqslant p-2\right)$,使 $g^x\equiv y(\bmod p)$,即求解 x 成功的概率为

$$\text{Succ}_{\text{DLP}}=\Pr[x\leftarrow(g,y)] \tag{1}$$

则对于任意多项式时间算法,Succ_{DLP} 可忽略不计.

零知识证明. 零知识证明这一概念是由 Goldwasser[30]等人于20世纪80年代初提出的. 零知识证明是一种协议,这种协议的一方称为证明者(Prover),它试图使被称为验证者(Verifier)的另一方相信某个论断是正确的,却不向验证者泄露任何有用的信息. 由 Goldwasser 等人提出的零知识证明是交互式的,也就是说,在证明者和验证者之间必须进行交互,才能实现零知识性,因而将这种零知识证明称为交互式零知识证明. 基于 public-coin 的零知识证明可以通过 Fiat-Shamir 协议转化为非交互式零知识证明.

Pederson 承诺. 给定 p 阶有限循环群 G,Pederson 承诺用于隐藏 $v\in\left[0,p-1\right]$,Pederson 承诺通常分为以下3个步骤.

(1)初始化(Setup)阶段. 选择大素数 p 生成的循环 G,选择其中两个生成元“g”,h,公开 p,g,h.

(2)承诺(Commitment)阶段. 承诺方随机选择随机数 r 盲化因子,计算承诺 comm 发送给接收者,其中:

$\text{comm}=g^v h^r(\bmod p)$,记作 $\text{comm}(v,r)$.

(3)打开(Open)阶段. 承诺方发送 (v,r) 给接收者,接收者验证:

$$g^v h^r\overset{?}{\equiv}\text{comm}(v,r)(\bmod p) \tag{2}$$

其中“$\overset{?}{\equiv}$”表示计算并判断等式两端是否相等.

利用 Pederson 承诺的同态性可以在不暴露交易具体金额的情况下实现交易线性关系的验证.

例如在不泄露 v_1、v_2、v_3 的情况下验证 $v_1+v_2=v_3$:

设

$$\begin{aligned}\text{comm}_1&=\text{comm}(v_1,r_1)\\ \text{comm}_2&=\text{comm}(v_2,r_2)\\ v_3&=v_1+v_2\end{aligned} \tag{3}$$

则

$$\begin{aligned}\text{comm}_1\cdot\text{comm}_2&=\text{comm}(v_1+v_2,r_1+r_2)\\ &=\text{comm}(v_3,r_1+r_2)\end{aligned} \tag{4}$$

要证明 $v_1+v_2=v_3$,只需验证 $r_1+r_2=r_3$.

两个承诺隐藏同一个值(EL 证明). EL 证明是由 Boudot[31]在2000年提出的. 该证明用于验证两个承诺是否隐藏相同的秘密值. 设两个承诺 A,B 对应相同的承诺值 m,其中 $A=g^m h_1^s(\bmod N)$,$B=f^m h_2^r(\bmod N)$,证明承诺 A 和 B 中承诺值相等由零知识证明 EL 协议实现,记作 $\text{EL}\left(m,s,r\middle|g,h_1,f,h_2\middle|A,B\right)$,由如下步骤构成.

(1)证明者

① 选择随机数 u,v_1,v_2,计算:

$$\begin{aligned}C_1&=g^u h_1^{v_1}(\bmod N)\\ C_2&=f^u h_2^{v_2}(\bmod N)\end{aligned} \tag{5}$$

② 计算:

$$H=\text{Hash}(C_1\| C_2) \tag{6}$$

③ 计算:

$$\begin{aligned}X&=u+Hm\\ X_1&=v_1+Hs\\ X_2&=v_2+Hr\end{aligned} \tag{7}$$

④ 公布:

$$(H,X,X_1,X_2) \tag{8}$$

(2) 验证者

① 计算:

$$C_1'=g^X h_1^{X_1}A^{-H}(\bmod N) \tag{9}$$

② 计算:

$$C_2'=f^X h_2^{X_2}B^{-H}(\bmod N) \tag{10}$$

③ 计算:

$$H'=\text{Hash}(C_1'\|C_2') \tag{11}$$

④ 验证:

$$H'\overset{?}{=}H \tag{12}$$

任意验证者都能对证明者给出的证明进行验证,通过上述验证则可相信承诺 A,B 具有相同的承诺值 m.

平方数证明(SQR 证明). SQR 证明是 Boudot[31]在2000年提出的. SQR 证明被用作验证承诺中的秘密值是否为平方数. 设 $E=g^{\alpha^2}h^{r_1}(\bmod N)$,证明承诺 E 中秘密值 α^2 由零知识证明 SQR 协议实现,记作 $\text{SQR}\left(\alpha,r_1\middle|g,h\middle|E\right)$,由如下步骤构成:

(1)证明者

① 选择随机数 r_2 并计算:

$$F=g^{\alpha}h^{r_2}(\bmod N) \tag{13}$$

② 计算:

$$\begin{aligned}r_3&=r_1-r_2\alpha\\ E'&=F^{\alpha}h^{r_3}(\bmod N)\end{aligned} \tag{14}$$

③ 执行 $\mathrm{EL}\left(\alpha, r_2, r_3 \middle| g, h, F, h \middle| F, E'\right)$

(a) 选择随机数 u, v_1, v_2，计算：

$$\begin{aligned} C_1 &= g^u h_1^{v_1} (\mathrm{mod}\ N) \\ C_2 &= f^u h_2^{v_2} (\mathrm{mod}\ N) \end{aligned} \tag{15}$$

(b) 计算：

$$H = \mathrm{Hash}(C_1 \| C_2) \tag{16}$$

(c) 计算：

$$\begin{aligned} X &= u + Hm \\ X_1 &= v_1 + Hs \\ X_2 &= v_2 + Hr \end{aligned} \tag{17}$$

④ 公布：

$$(H, X, X_1, X_2, F, E') \tag{18}$$

(2)验证者

① 计算：

$$C_1' = g^X h^{X_1} F^{-H} (\mathrm{mod}\ N) \tag{19}$$

② 计算：

$$C_2' = F^X h^{X_2} E'^{-H} (\mathrm{mod}\ N) \tag{20}$$

③ 验证：

$$\mathrm{Hash}(C_1' \| C_2') \overset{?}{=} H \tag{21}$$

范围证明. 范围证明是为了零知识地证明整数 $v \in [a, b]$，其中 a, b 为整数，$a < b$. 零知识范围证明一般由以下三部分组成.

初始化(Setup)阶段. 输入安全参数 λ，概率多项式算法 set 生成公共参数 pp，记作：

$$\mathrm{pp} \overset{s}{\leftarrow} \mathrm{sec}(\lambda) \tag{22}$$

证明(Prove)阶段. 输入公共参数 pp，秘密值 v，范围 $[a, b]$，概率多项式算法 Prove 生成证明 π，记作：

$$\pi \overset{\$}{\leftarrow} \mathrm{prove}(\mathrm{pp}, v, a, b) \tag{23}$$

验证(Verify)阶段. 输入公共参数 pp，证明 π，概率多项式算法 Ver，返回 d，$d \in \{0, 1\}$，其中 1 代表接受 π，验证通过；0 代表拒绝，验证不通过. 记作：

$$d \overset{\$}{\leftarrow} \mathrm{Ver}(\mathrm{pp}, \pi) \quad d \in \{0, 1\} \tag{24}$$

3.3 安全目标及威胁模型

(1)安全目标

本文的安全目标是隐藏具体交易金额、隐藏交易方身份以及隐藏交易关系图，同时有助于监管方审计银行的交易行为. 具体而言：①方案允许银行以隐藏金额的方式发起转移交易；②交易公开存储于账本中，除监管方外其他银行无法辨认交易各方的具体身份；③隐藏交易关系图，即交易不仅隐藏交易金额的去向，对交易金额的来源也是隐藏的.

在实现安全目标的同时需要支持审计功能，监管方可以对银行发起审计，询问银行交易相关内容，如“Bank_i 在 t_0 时刻拥有多少资产”或“Bank_j 在 t_1~t_2 时段的收支状况”，银行给出答案和相应的证明，监管方利用账本中的交易记录进行计算和验证，判断银行给出答案是否正确. 为保护交易隐私，本方案的设计确保监管方在没有询问银行的情况下无法自行从账本中得出想要的结果.

(2)威胁模型

假设 1：监管方和账本是诚实的. 这个假设是合理的，一般情况下监管方默认是公正的，它不对银行给出的答案和证明进行篡改，忠实地进行审计和验证并给出审计结论.

假设 2：银行是非诚实的，它们可以试图隐藏资产、捏造资产或窃取其他银行的资产. 例如，银行可能否认某些自己发起的交易以掩藏非法转账行为，或是审计时虚报资产以逃避税收，或试图向系统伪造其他银行对自己的转账以窃取资产. 银行可以合盟作恶，但有假设 1 银行不能与监管方合盟.

4 具体过程

4.1 系统初始化

监管方选择大素数 p, q，计算：

$$\begin{aligned} N &= pq \\ \varphi(N) &= (p-1)(q-1) \end{aligned} \tag{25}$$

对于 mod N 的群 G，由有限交换群结构定理，可将其分解为素数幂循环群的直和，在其循环群中选择阶数较大的子群 $<g>$，构造陷门 $\{p-1 | h = g^{p-1} (\mathrm{mod}\ N)\}$，将 g, h 用于计算交易承诺，账本和监管方秘密存储 $p-1$、$q-1$.

银行 Bank_i 向监管方注册，得到公私钥对 $(\mathrm{pk}_i, \mathrm{sk}_i)$，$\mathrm{pk}_i \cdot \mathrm{sk}_i = 1 (\mathrm{mod}\ \varphi(N))$.

4.2 交易

借鉴 ZkLedger 中的交易记录方法，采用一种表格形式记录交易，其中表格的一行对应一笔交易，表格中一列对应某一银行的全部交易.

账本将交易排序后进行记账，并记录交易时间. 表 1 表示 Bank_1 向 Bank_2 转账的交易，交易金额为 v，本文允许 Bank_1 在不经过其他银行(包括 Bank_2)许可的条件下发起交易.

Bank_1 计算自己和其他所有银行的承诺：

$$\mathrm{comm}_i^t = g^{d_i^t} h^{v_i^t} (\mathrm{mod}\ N) \tag{26}$$

其中，$\boldsymbol{v}_i^t$ 代表交易 t 中 Bank_i 的交易金额，d_i^t 为盲化因子.

v_i^t：在交易 t 中，Bank_1 转出金额 v，在计算承诺时 $v_1^t = -v$；对于 Bank_2，该笔交易接收金额 v，在计算承诺时 $v_2^t = v$；对于其他银行，不参与该笔交易，在计算承诺时 $v_i^t = 0$.

d_i^t：对于交易发起方 Bank_1，$d_1^t = \mathrm{sk}_1$，对于其他银行，Bank_1 选择随机数作为 d_i^t，并使它们满足

表1　由 Bank_1 向 Bank_2 的转账交易

序号	时间	Bank_1	Bank_2	…	Bank_n
…	…	…	…	…	…
t	8-5 08:59	$\mathrm{comm}_1^t(d_1,-v)$ π_1,π_2,π_3,π_4 Token_1^t	$\mathrm{comm}_2^t(d_2,v)$ π_1,π_2,π_3,π_4 Token_2^t	…	$\mathrm{comm}_n^t(d_n,0)$ π_1,π_2,π_3,π_4 Token_n^t
…	…	…	…	…	…

$$\sum_{i=1}^{n} d_i^t = 0$$

交易采取Pederson承诺保证了交易金额在账本中是“隐藏”的. 将交易发起方金额设置为负,接收方金额设置为正,非参与方金额设置为0,除交易发起方和监管方(利用陷门),其他任意银行无法对交易金额进行区分,因此交易各方的身份也得到“隐藏”. 为了确保交易的合理性,发起交易时还需给出下列证明.

(1)交易应保持资产不变,即交易没有凭空地产生或者销毁资产,其证明记作π_1.

由于单笔交易中,交易发起方Bank_1的承诺金额为$-v$,接收方Bank_2的承诺金额为$+v$,其他银行的承诺金额为0,因此:

$$\sum_{i=1}^{n} v_i^t = 0 \tag{27}$$

选择随机数d_i时满足:

$$\sum_{i=1}^{n} d_i^t = 0 \tag{28}$$

因此:

$$\prod_{i=1}^{n} \mathrm{comm}_i^t = g^{v_i^t} h^{d_i^t} = g^{\sum_{i=1}^{n} v_i^t} h^{\sum_{i=1}^{n} d_i^t} = 1 (\mathrm{mod}\ N) \tag{29}$$

于是对于证明π_1并不需额外设置(设置为空),在验证时计算$\prod_{i=1}^{n} \mathrm{comm}_i^t$,乘积为1时则可证明交易没有凭空产生或者销毁资产.

(2)不应有无效交易,即单笔交易中交易金额不能全为0,其证明记作π_2.

为了避免大量无效交易占用账本空间、降低交易效率,应避免交易金额全为0的情况. 假设所有交易金额均为0,则

$$\mathrm{comm}_i = g^{d_i} h^0 = g^{d_i} (\mathrm{mod}\ N) \tag{30}$$

此时对于交易发起方:

$$\mathrm{comm}_i = g^{\mathrm{sk}_i} (\mathrm{mod}\ N) \tag{31}$$

由于监管方知道sk_i,它可以计算并存储g^{sk_i},依次比较comm_i和g^{sk_i},当某一组值相等,说明此时为无效交易.

于是对于证明π_2并不需额外设置(设置为空),监管方验证时执行下列算法:

For i in range n :
If $\mathrm{comm}_i \neq g^{\mathrm{sk}_i}$:
Continue
Else:
Return 1
Return 0

当返回值为1时,说明该交易的交易金额全为0,为无效交易;返回值为0时,说明该交易并非无效交易.

对于其他验证者(如私人审计),不知道sk_i,但注意到pk_i是公开的,且有

$$(g^{\mathrm{sk}_i})^{\mathrm{pk}_i} = g^{\mathrm{sk}_i \cdot \mathrm{pk}_i} = g (\mathrm{mod}\ N) \tag{32}$$

因此其他验证者可以通过下列算法进行验证:

For i in range n :
If $\left(\mathrm{comm}_i\right)^{\mathrm{pk}_i} \neq g$:
Continue
Else:
Return 1
Return 0

当返回值为1时,说明该交易的交易金额全为0,为无效交易;返回值为0时,说明该交易并非无效交易.

(3)当交易金额为v时,交易发起方承诺值为$-v$,由于$-v$在模N的群运算中对应另一个正整数$N-v$,在$[0,N-1]$范围内$+v$和$N-v$在模N的群运算下互为相反数,因此账本无法直接通过承诺值分辨交易发起方和接收方,必须引入证明加以区分,并保证发起方不能将自身承诺值伪造为$+v$(即将交易伪造成其他银行对自身的转账),其证明记作π_3.

为了使账本验证交易发起方身份,常用的做法是采用数字签名,如schnorr签名. 由于银行公钥是公开的,任意银行都可以公开验证交易签名,这种方式泄露了签名者(即交易发起方)的身份. 本文的安全目标之一是“隐藏”交易各方的身份,因此账本不能通过使用数字签名的方式验证转账者身份. 注意到对于交易发起方Bank_i:

$$\begin{aligned} &\mathrm{comm}_i^t \\ &= g^{\mathrm{sk}_i} h^{-v} \leftarrow n \\ &= g^{\mathrm{sk}_i} g^{-(p-1)v} (\mathrm{mod}\ N) \end{aligned} \tag{33}$$

则

$$\begin{aligned} &\mathrm{comm}_i^{\mathrm{pk}_i (q-1)} \leftarrow n \\ &= g^{\mathrm{sk}_i \cdot \mathrm{pk}_i \cdot (q-1)} g^{-(p-1) \cdot v \cdot \mathrm{pk}_i \cdot (q-1)} \leftarrow n \\ &= (g^{\mathrm{sk}_i \cdot \mathrm{pk}_i})^{(q-1)} \cdot (g^{(p-1)(q-1)})^{-v \cdot \mathrm{pk}} \\ &= g^{q-1} (\mathrm{mod}\ N) \end{aligned} \tag{34}$$

于是对于证明π_3并不需额外设置(设置为空),由于账本知道$q-1$,在验证时逐一计算$\mathrm{comm}_i^{(\mathrm{pk}_i (q-1))}$,当某一$\mathrm{comm}_i^{\mathrm{pk}_i (q-1)} = g^{q-1} (\mathrm{mod}\ N)$时,确定$\mathrm{Bank}_i$为交易发起方,$\mathrm{comm}_i$对应的承诺金额为负.

(4)交易发起方需要有足够的资产才能发起交易,

交易金额不能高于自身资产,其证明记作π_4.

以表1中的交易t为例,Bank_1在发交易t时需证明其资产大于交易金额v.

记:

$$\prod_{j=1}^{t-1}\text{comm}_i^j=\text{comm}_i^{[:t-1]} \tag{35}$$

记:

$$\prod_{j=1}^{t-1}\text{Token}_i^j=\text{Token}_i^{[:t-1]} \tag{36}$$

在发起交易t前,Bank_1将账本中所在列承诺相乘,得到:

$$\begin{aligned}\text{comm}_1^{[:t-1]}&=\prod_{j=1}^{t-1}\text{comm}_1^j\\&=g^{\sum_{j=1}^{t-1}d_1^j}h^{\sum_{j=1}^{t-1}v_1^j}(\text{mod}\,N)\end{aligned} \tag{37}$$

生成交易t时计算:

$$\begin{aligned}\text{comm}_1^{[:t]}&=\text{comm}_1^{[:t-1]}\cdot\text{comm}_1^t\\&=g^{\sum_{j=1}^{t}d_1^j}h^{\sum_{j=1}^{t}v_1^j}(\text{mod}\,N)\end{aligned} \tag{38}$$

记:$g^{\sum_{j=1}^{t}d_1^j h^{\sum_{j=1}^{t}v_1^j}}$为$g^{\sum dh^{\sum v}}$.

为了说明Bank_1的资产大于v,转化为零知识地证明$\sum v>0$. 因为$g^{\sum dh^{\sum v}}=g^{\sum dh^{N+\sum v}}(\text{mod}\,N)$,通常还需要对$\sum v\in[0,N-1]$进行范围证明. 我们发现当$N$足够大时,$[0,N-1]$已经满足银行交易所需($N>2^{40}$时,$v+N$已经大于一万亿),因此本方案默认$\sum v\in[0,N-1]$,这种设定节省了这部分范围证明的开销. 现只需证明$\sum v>0$.

具体地,对于证明任意$v>0$的方法我们通过对Tsai[32]的方案进行改进,将方案证明$v\in[a,b]$改进为只需证明$v>0$.

设承诺$c=g^r h^v(\text{mod}\,N)$,为了证明$v\geqslant 0$,证明者进行如下操作.

① 随机选择非零整数w和r',计算:

$$c'=c^{w^2}g^{r'}(\text{mod}\,N) \tag{39}$$

② 计算SQR证明:

$$\text{SQR}(w,r'|c,g|c') \tag{40}$$

③ 随机选择整数α,计算$M=\alpha^2$,满足:

$$M\leqslant w^2\cdot v \tag{41}$$

④ 计算:

$$R=w^2\cdot v-M \tag{42}$$

⑤ 选择正整数r_1和z,满足:

$$r_1+z=w^2\cdot r+r' \tag{43}$$

⑥ 生成R和M的承诺:

$$\begin{aligned}c_1'&=g^{r_1}h^M(\text{mod}\,N)\\c_2'&=g^z h^R(\text{mod}\,N)\end{aligned} \tag{44}$$

⑦ 计算SQR证明:

$$\text{SQR}(\alpha,r_1|h,g|c_1') \tag{45}$$

⑧ 公布$\text{prove}=\{c,c_1',c_2',R,\text{SQR}_1,\text{SQR}_2\}$,其中:

$$\begin{aligned}\text{SQR}_1&=\text{SQR}(w,r'|c,g|c')\\\text{SQR}_2&=\text{SQR}(\alpha,r_1|h,g|c_1')\end{aligned} \tag{46}$$

验证者对prove进行如下验证.

① 验证:

$$\text{SQR}_1\overset{?}{=}\text{SQR}(w,r'|c,g|c') \tag{47}$$

② 验证:

$$c'\overset{?}{=}c_1'c_2' \tag{48}$$

③ 验证:

$$\text{SQR}_2\overset{?}{=}\text{SQR}(\alpha,r_1|h,g|c_1') \tag{49}$$

④ 验证R是否大于0.

通过验证后,验证者可相信承诺$c=g^r h^v(\text{mod}\,N)$中$v\geqslant 0$.

正确性:为了证明$v\geqslant 0$,可以证明$w^2\cdot v\geqslant 0$. 令$R=w^2\cdot v-M$,令$M=\alpha^2\geqslant 0$,因此只用验证$R+M=w^2\cdot v$和$R\geqslant 0$即可证明$w^2\cdot v\geqslant 0$.

如果验证①通过,则说明$c'=c^{w^2}g^{r'}$;

如果验证②通过,则说明证明者给出的计算是正确的,且有$R+M=w^2\cdot v$,因为:

$$\begin{aligned}c&=c^{w^2}g^{r'}(\text{mod}\,N)\\&=(g^r h^v)^{w^2}\cdot g^{r'}(\text{mod}\,N)\\&=(g^r h^v)^{w^2}\cdot g^{r'}(\text{mod}\,N)\\&=g^{r\cdot w^2+r'}h^{w^2\cdot v}(\text{mod}\,N)\\&=g^{r_1}h^M\cdot g^z h^R(\text{mod}\,N)n\\&=c_1c_2(\text{mod}\,N)\end{aligned} \tag{50}$$

如果验证③通过,则说明c_1'中h的指数为平方数,即$M\geqslant 0$;

如果验证④通过,则说明$R>0$,即可推出$v\geqslant 0$.

于是对于证明π_4,π_4设置为$\{c,c_1',c_2',R,\text{SQR}_1,\text{SQR}_2\}$,其中$c=\text{comm}_i^{[:t]}$.

4.3 审计

审计是交易系统的关键组成部分,监管方使用各种技术指标来衡量系统性金融风险,大部分指标是基于交易金额的线性组合,其中最重要的是求和和求平均值. 本文利用承诺的同态性能简洁地实现求和计算. 在计算平均值时,不能简单地将账本中银行所在列的承诺相乘,将乘积的承诺值“打开”除以交易的数量作为该银行的平均交易额,因为对某一银行而言,它并不参与账本中的大部分交易,在这些不相关交易中其承诺值为0,平均值的计算应将这部分交易剔除掉. 然而在其他方案中,监管方无法直接辨认承诺值是否为0,因此还需对每一笔交易额外引入交易金额是否为0的承诺,使平均值的计算变得非常复杂. 本方案的构造使

监管方能够对银行的交易类别(收入、支出、不参与)进行区分,在此基础上能简单地实现平均值的计算,这是其他方案不具备的.

首先引入审计令牌Token,由于交易由某一银行单方面发起,如果其他银行Bank_i不知晓d_i^t的信息,Bank_i将无法完成审计,因此需要将d_i"告知"Bank_i而不被其他银行知晓. 对于交易t中Bank_i对应的Token记为Token_i^j,Token_i^j的具体构造是$\mathrm{tk}_i^{t\,\mathrm{pk}_i}$:① 若$\mathrm{Bank}_i$为非交易发起方$\mathrm{tk}_i^t=g^{d_i^t}(\mathrm{mod}\,N)$;②对于交易发起方$\mathrm{Bank}_j$,选择随机数$r_t$,$\mathrm{tk}_i^t=g^{r_t}$,在表1的交易$t$中对应$\mathrm{Bank}_1$的Token为$g^{r_t\,\mathrm{pk}_1}$.

使用$g^{r_t\,\mathrm{pk}_j}$而不使用$g^{d_j^{\mathrm{pk}_j}}$(即$\mathrm{sk}_j^{\mathrm{pk}_j}$,发起方的$d_j$=$\mathrm{sk}_j$)作为交易发起方$\mathrm{Bank}_j$的Token是因为对于每个$\mathrm{Bank}_j$发起的交易,$g^{d_j^{\mathrm{pk}_j}}$是不保持不变的,这意味着如果选取$g^{d_j^{\mathrm{pk}_j}}$作为$\mathrm{Bank}_j$的Token,通过查询账本中$\mathrm{Bank}_j$所在列就能找到多个相同的Token,而这些Token所对应交易暴露了发起方就是Bank_j.

本文对常用的求和审计、平均值审计进行详细说明. 其中,求和审计包括资产审计、支出审计和收入审计. 审计的主体是审计员和银行两方,审计员向银行提出问题,银行对所提问题进行回复并给出相关"证据",审计员对银行回复内容进行验证. 审计员一般是权威的监管方,对于资产审计也支持私人审计者. 权威的监管方较私人审计者相比拥有更高的权限,他们拥有限门信息.

(1)求和审计

(1-1)监管方审计

① 资产. 监管方询问银行Bank_i"直到交易t发生时你有多少资产",将Bank_i在交易t发生时的资产用asset_i^t表示,此时银行进行如下操作:

(a)将账本中Bank_i所在列序号t之前的承诺相乘,得到:$\mathrm{comm}_i^{[:t]}(\mathrm{mod}\,N)$(记为$c_1$)

(b)将账本中Bank_i所在列序号t之前的Token相乘,得到:

$$\prod_{j=1}^{t}\mathrm{Token}_i^j=\prod_{j=1}^{t}g^{\mathrm{tk}_i^j}{}^{**\,\mathrm{pk}_i}=\left(\prod_{j=1}^{t}g^{\mathrm{tk}_i^j}\right)^{\mathrm{pk}_i}(\mathrm{mod}\,N)\quad(51)$$

(记$\prod_{j=1}^{t}g^{\mathrm{tk}_i^j}$为$c_2$)

(c)向监管方发送:

$$(c_1,c_2,\mathrm{asset}_i^t)$$

监管方进行如下验证:

(a)验证:

$$\mathrm{comm}_i^{[:T]}(\mathrm{mod}\,N)\overset{?}{=}c_1\quad(52)$$

(b)验证:

$$\prod_{j=1}^{t}\mathrm{Token}_i^j(\mathrm{mod}\,N)\overset{?}{=}c_2^{\mathrm{pk}_i}(\mathrm{mod}\,N)\quad(53)$$

(c)验证:

$$c_2h^{\mathrm{asset}_i^t}(\mathrm{mod}\,N)\overset{?}{=}c_1\quad(54)$$

如果验证通过,监管方可相信银行Bank_i给出的资产asset_i^t是正确的.

② 支出. 监管方询问银行Bank_i"交易t_1~t_2之间你一共转出了多少资金",将Bank_i在交易t_1~t_2内转出的资金用$\mathrm{vout}_i^{[t_1:t_2]}$表示,此时银行进行如下操作:

(a)Bank_i找出交易t_1~t_2内发起的全部交易,其承诺设为

$$\mathrm{commout}_i^j(j=k_1,k_2,\cdots,k_u)\quad(55)$$

(b)计算:

$$\prod_{j=k_1}^{k_u}\mathrm{commout}_i^j(\mathrm{mod}\,N)\quad(56)$$

(c)向监管方发送$(g^{\mathrm{sk}_i},c_3,\mathrm{vout}_i^{[t_1:t_2]})$

监管方进行如下验证:

(a)利用π_3找出交易t_1~t_2内Bank_i发起的全部交易,其承诺设为

$$\mathrm{commout}_i^{j'}(j'=k_1',k_2',\cdots,k_u')\quad(57)$$

(b)验证:

$$\prod_{j'=k_1'}^{k_u'}\mathrm{commout}_i^{j'}(\mathrm{mod}\,N)\overset{?}{=}c_3\quad(58)$$

(c)验证:

$$g^{\mathrm{sk}_i}h^{\mathrm{vout}_i^{[t_1:t_2]}}(\mathrm{mod}\,N)\overset{?}{=}c_3\quad(59)$$

如果验证通过,则监管方可相信银行Bank_i给出的$\mathrm{vout}_i^{[t_1:t_2]}$是正确的.

③ 收入. 监管方询问银行Bank_i"交易t_3~t_4之间你一共接收了多少资金",将Bank_i在交易t_3~t_4之间转出的资金用$\mathrm{vin}_i^{[t_3:t_4]}$表示,此时银行进行如下操作:

(a)Bank_i在交易t_3~t_4之间接收转账的全部交易,其承诺设为:

$$\mathrm{commin}_i^{j''}(j''=k_1'',k_2'',\cdots,k_w'')\quad(60)$$

(b)计算:

$$\prod_{j''=k_1''}^{k_w''}\mathrm{commin}_i^{j''}(\mathrm{mod}\,N)\quad(61)$$

(c)计算:

$$\prod_{j''=k_1''}^{k_w''}\mathrm{Token}_i^{j''}(\mathrm{mod}\,N)\quad(62)$$

(d)向监管方发送$(c_4,c_5,\mathrm{vin}_i^{[t_3:t_4]})$

监管方进行如下验证:

(a)设交易t_3~t_4之间Bank_i的承诺为

$$\mathrm{comm}_i^j\left(j=t_3,t_3+1,\cdots,t_4\right)$$

利用sk_i解密Token_i^j得到d_i^j：

$$g^{d_i^j}=\mathrm{Token}_i^j\mathrm{sk}_i \tag{63}$$

(*b*)找出交易t_3~t_4之间内与Bank_i无关的交易：

因为与Bank_i无关的交易其$\boldsymbol{v}_i^j$为0，因此：

$$\mathrm{comm}_i^j=g^{d_i^j}h^{v_i^j}=g^{d_i^j} \tag{64}$$

逐一验证：

$$\mathrm{comm}_i^j\overset{?}{**}=g^{d_i^j}\left(j=t_3,t_3+1,\cdots,t_4\right) \tag{65}$$

满足上述等式的comm_i^j对应的交易即为与Bank_i无关的交易，将这些交易的comm_i^j记作：

$$\mathrm{commzero}_i^j\left(j=l_1,l_2,\cdots,l_x\right)$$

(*c*)找出交易t_3~t_4之间Bank_i发起的交易：

找到交易发起方的方法在π_3中已经给出，对$\mathrm{comm}_i^j\left(j=t_3,t_3+1,\cdots,t_4\right)$逐一运用$\pi_3$就能得到$\mathrm{Bank}_i$发起的交易，记作：

$$\mathrm{commout}_i^j\left(j=m_1,m_2,\cdots,m_y\right)$$

(*d*)找出交易t_3~t_4之间Bank_i接收转账的交易：

在$\mathrm{comm}_i^j\left(j=t_3,t_3+1,\cdots,t_4\right)$中除去$\mathrm{Bank}_i$不相关的交易$\mathrm{commzero}_i^j\left(j=l_1,l_2,\cdots,l_x\right)$和$\mathrm{Bank}_i$发起的交易$\mathrm{commout}_i^j\left(j=m_1,m_2,\cdots,m_y\right)$剩下的交易即为$\mathrm{Bank}_i$接收转账的交易，记为

$$\mathrm{commin}_i^j\left(j=n_1,n_2,\cdots,n_z\right)$$

(*e*)验证：

$$\prod_{j=n_1}^{n_z}\mathrm{commin}_i^j(\mathrm{mod}\,N)\overset{?}{=}c_4 \tag{66}$$

(*f*)验证：

$$\prod_{j=n_1}^{n_z}\mathrm{Token}_i^j(\mathrm{mod}\,N)\overset{?}{=}c_5 \tag{67}$$

(*g*)验证：

$$g^{(c_2)^{\mathrm{sk}_i}}h^{\mathrm{vin}_i^{[t_3:t_4]}}(\mathrm{mod}\,N)\overset{?}{=}c_4 \tag{68}$$

如果验证通过，则监管方可相信银行Bank_i给出的转入资金$\mathrm{vin}_i^{[t_3:t_4]}$是正确的.

(1-2)私人审计

① 资产. 对于资产审计，因为不需要使用监管方独有的陷门信息，私人审计过程与监管方审计过程相同. 因此银行不仅能够接受监管方的审计，也可对任意其他银行或私人审计做出可信答复. 另一方面，计算c_1,c_2所需$\mathrm{comm}_i^j,\mathrm{Token}_i^j\left(j=1,2,\cdots,t\right)$在账本中公开存储，审计者可以从账本中直接获取$\mathrm{comm}_i^j,\mathrm{Token}_i^j$进行计算和验证，因此资产审计是公开可验证的.

② 支出审计和收入审计. 对于支出审计和收入审计，由于私人审计者不具备陷门信息，无法像监管方一样自行区分交易类型，对于支出审计和收入审计需要银行给出更多的信息才能完成审计，并且支出审计和收入审计必须绑定同时执行.

私人审计者询问银行Bank_i“交易t_4~t_5之间你一共转出了多少资金，收入多少资金”，银行进行如下操作：

(*a*)Bank_i找出交易t_4~t_5内发起的全部交易，其承诺设为

$$\mathrm{commout}_i^{j'\left(j'=k_1',k_2',\cdots,k_u'\right)};$$

(*b*)计算：

$$\prod_{j'=k_1'}^{k_u'}\mathrm{commout}_i^{j'}=g^{\mathrm{sk}_i}h^{\mathrm{vout}_i^{[t_5:t_6]}}(\mathrm{mod}\,N) \tag{69}$$

(*c*)找出交易t_4~t_5内接收转账的全部交易，其承诺设为

$$\mathrm{commin}i^{\wedge}(j^{\wedge''})(j^{\wedge''}=k_{-1}\Lambda'',k\_2\Lambda'',\cdots\cdots,k\_w^{\Lambda''}) \tag{70}$$

(*d*)计算：

$$\prod_{j''=k_1''}^{k_w''}\mathrm{commin}_i^{j''}=g^{\mathrm{din}_i^{[t_5:t_6]}}h^{\mathrm{vin}_i^{[t_5:t_6]}}(\mathrm{mod}\,N) \tag{71}$$

(*e*)找出其余与Bank_i无关的交易，其承诺设为

$$\mathrm{commzero}_i^{j'''}(j'''=k_1''',k_2''',\cdots,k_x''') \tag{72}$$

(*f*)计算：

$$\prod_{j'''=k_1'''}^{k_w'''}\mathrm{commzero}_i^{j'''}=g^{\mathrm{dzero}_i^{[t_5:t_6]}}h^0(\mathrm{mod}\,N) \tag{73}$$

(*g*)向私人审计者发送：

$$\begin{aligned}&\left(j'=k_1',k_2',\cdots,k_u',j''=k_1'',k_2'',\cdots,k_w'',j'''\right.\\&\quad=k_1''',k_2''',\cdots,k_x''',(g^{\mathrm{sk}_i},\mathrm{vout}_i^{[t_5:t_6]}),\\&\quad\left.(\mathrm{din}_i^{[t_5:t_6]},\mathrm{vin}_i^{[t_5:t_6]}),\mathrm{dzero}_i^{[t_5:t_6]}\right)\end{aligned} \tag{74}$$

私人审计者进行如下验证.

(*a*)验证：

$$\left(\prod_{j'=k_1'}^{k_u'}\mathrm{commout}_i^{j'}\right)^{\mathrm{pk}_i}\overset{?}{\equiv}g\cdot h^{(\mathrm{pk}_i\cdot\mathrm{vout}_i^{[t_5:t_6]})}(\mathrm{mod}\,N) \tag{75}$$

该步骤是为了验证序号$j'=k_1',k_2',\cdots,k_u'$的交易是否都是Bank_i发起的转账交易，同时也能验证$\mathrm{vout}_i^{[t_5:t_6]}$是否正确；

(*b*)验证：

$$g^{(\mathrm{dim}_i^{[t_5:t_6]})}\overset{?}{**}=\prod_{j''=k_1''}^{k_w''}\mathrm{Token}_i^{j''}(\mathrm{mod}\,N) \tag{76}$$

该步骤是为了判断序号$j''=k_1'',k_2'',\cdots,k_w''$的交易中是否含有$\mathrm{Bank}_i$发起的转账交易，若相等则证明$j''=k_1'',k_2'',\cdots,k_w''$不含有$\mathrm{Bank}_i$发起的转账交易；

(*c*)验证：

$$\prod_{j''=k_1''}^{k_w''}\mathrm{commin}_i^{j''}\overset{?}{*}:g^{\mathrm{din}_i^{[t_5:t_6]}}h^{\mathrm{vin}_i^{[t_5:t_6]}}(\mathrm{mod}\,N) \tag{77}$$

(*d*)验证：

$$g^{(\dim_i^{[t_5:t_6]})}\overset{?}{=}\prod_{j''=k_1''}^{k_u''}\mathrm{Token}_i^{j''}(\mathrm{mod}\,N)\tag{78}$$

该步骤是为了判断序号$j'''=k_1''',k_2''',\cdots,k_x'''$的交易中是否含有$\mathrm{Bank}_i$发起的转账交易,若相等则证明$j'''=k_1''',k_2''',\cdots,k_x'''$不含有$\mathrm{Bank}_i$发起的转账交易;

(*e*)验证:

$$\prod_{j'''=k_1'''}^{k_u'''}\mathrm{commzero}_i^{j'''}\overset{?}{\equiv}g^{\mathrm{dzero}_i^{[t_5:t_6]}}(\mathrm{mod}\,N)\tag{79}$$

若验证(*d*)通过,则排除了$j'''=k_1''',k_2''',\cdots,k_x'''$是$\mathrm{Bank}_i$发起的转账交易,在此基础上若能通过验证(*e*),则还能说明j'''中不含有Bank_i接收转账的交易,则$j'''=k_1''',k_2''',\cdots,k_x'''$全部为与$\mathrm{Bank}_i$无关的交易.

当所有验证通过后,私人审计者可以相信$\mathrm{vin}_i^{[t_5:t_6]}$和$\mathrm{vout}_i^{[t_5:t_6]}$是正确的.另一方面,审计过程中所需$\mathrm{comm}_i^j,\mathrm{Token}_i^j$在账本中公开存储,审计者可以从账本中直接获取$\mathrm{comm}_i^j,\mathrm{Token}_i^j$进行计算和验证,因此支出审计和收入审计也是公开可验证的.

(2)平均值

监管方(私人审计者)除了进行求和查询之外,还可以发出更复杂的查询,例如求各类平均值.监管方可以询问银行Bank_i"交易t_7~t_8之间的平均转账金额、平均收款金额"等.在4.3节中已经给出了监管方区分银行不同交易类型的方法,因此能够实现平均值的计算.对于已知的其他承诺方案,由于无法区分交易的类型,无法实现平均值的计算,或者必须引入更复杂的机制,比如对交易值是否为0再进行承诺.相比之下,本文能够简单地实现平均值的计算,这得益于陷门的设计.

4.4　优化

可以对上述过程进一步优化以提高发起交易、验证和审计的效率.具体来说,各银行可以随着交易的创建不断缓存账本中承诺的乘积,在发起交易和开展审计时直接使用,有效缩短了生成交易和给出审计答复所需时间;由于大多数交易并不包括所有银行,每个银行都可以预生成承诺值为0的承诺,也能有效缩短交易生成时间.定期更换g,h有助于提升系统的安全性.

5　安全性

定理1　如果离散对数问题是困难的,那么敌手Adv能够窃取资产(伪造其他银行发起对自己的转账交易)的概率是忽略不计的.

证明　考虑敌手$\mathrm{Bank}_{\mathrm{Adv}}$想要伪造交易$t'$,使$\mathrm{Bank}_i$向自己发起金额为$v$的转账交易.为此$\mathrm{Bank}_{\mathrm{Adv}}$需向账本提交交易,在账本$\mathrm{Bank}_i$所在列伪造承诺$g^{\mathrm{sk}_i}h^{-v_i'}(\mathrm{mod}\,N)$,在自己所在列计算承诺$g^{d_{\mathrm{adv}}^{t'}}h^{v_{\mathrm{adv}}'}(\mathrm{mod}\,N)$,其中$v_i'=v_{\mathrm{adv}}''=v$.由于$\mathrm{Bank}_{\mathrm{Adv}}$不知道$\mathrm{Bank}_i$的私钥$\mathrm{sk}_i$,需要伪造$d$,使$g^dh^{-v}$通过$\pi_3$,即需满足:

$$(g^dh^{-v})^{\mathrm{pk}_i(q-1)}=g^{q-1}(\mathrm{mod}\,N)\tag{80}$$

进一步转化:

$$\begin{aligned}&(g^dh^{-v})^{\mathrm{pk}_i(q-1)}\\&=g^{d(q-1)\mathrm{pk}_i}(\mathrm{mod}\,N)g^{d(q-1)\mathrm{pk}_i=g^{q-1}(\mathrm{mod}\,N)}\\&(g^{d\cdot\mathrm{pk}_i})^{q-1}=g^{q-1}(\mathrm{mod}\,N)\end{aligned}\tag{81}$$

即需使d满足:

$$g^d=g^{-\mathrm{pk}_i}(\mathrm{mod}\,N)\tag{82}$$

$\mathrm{Bank}_{\mathrm{Adv}}$要找到$d$使$d=(\mathrm{pk}_i)^{-1}(\mathrm{mod}\,\varphi(N))$,而$\varphi(N)$是未知的,该求解过程等同于在RSA密码体制中已知公钥求解私钥的过程,因此$\mathrm{Bank}_{\mathrm{Adv}}$能够窃取资产的概率是忽略不计的.

定理2　本方案具有抗抵赖性,即任意Bank_i无法否认账本中任意一笔由自己发起的交易.

证明　由定理1,Bank_i无法伪造其他银行发起的交易,若需对某笔自己发起的交易进行抵赖,它只能选择不将自己的sk_i"嵌入"交易中,即发起交易时不使用sk_i作为自己的d_i^t.此时,交易中未"嵌入"任一银行的sk(交易中任意d_i^t都不与所对应银行的私钥sk相同),则账本无法通过π_3,交易终止,该交易不会被记入账本.因此,只要是记入账本的交易,其发起方都无法进行否认.

定理3　对于任意T_9时刻,敌手$\mathrm{Bank}_{\mathrm{Adv}}$的资产为$a$,$\mathrm{Bank}_{\mathrm{Adv}}$无法伪造$a'(a'\neq a)$,使监管方相信其资产为$a'$,即便是面对任意私人审计,$\mathrm{Bank}_{\mathrm{Adv}}$也无法进行欺骗.

证明　设T_9时刻对应的交易序号为t_9.

$$\mathrm{comm}_{\mathrm{adv}}^{[1:t_9]}=g^{\sum_{j=1}^{t_9}d_{\mathrm{adv}}j}h^{\sum_{j=1}^{t_9}v_{\mathrm{adv}}j}(\mathrm{mod}\,N)\tag{83}$$

其中,$\sum_{j=1}^{t_9}v_{\mathrm{adv}}j=a$,记$\sum_{j=1}^{t_9}d_{\mathrm{adv}}j$为$d$.

(1)对于监管者

由于监管方可以计算得到任意$d_{\mathrm{adv}}j$,因此$\mathrm{Bank}_{\mathrm{Adv}}$需找到$a'$,满足:

$$\begin{aligned}g^dh^{a'}&=g^dh^a(\mathrm{mod}\,N)\\h^{a'}&=h^a(\mathrm{mod}\,N)\end{aligned}\tag{84}$$

由:

$$g=h^x(\mathrm{mod}\,N)\tag{85}$$

又有:

$$<g>=<h>\tag{86}$$

由于g为循环群,所以要使:

$$h^{a'}=h^a(\mathrm{mod}\,N)\tag{87}$$

则需:

$a=a'$或$a'=a+n\cdot N(n=1,2,3\cdots)$.

由于限制了交易金额范围为$[0,N-1]$,所有$a=a'$时上式成立,因此$\mathrm{Bank}_{\mathrm{Adv}}$无法伪造$a'$.

(2)对于私人审计

对于私人审计而言，审计者不知道$d_{adv}j$，因此Bank_{Adv}需找到a'和d'，满足：

$$g^{d'}h^{a'}=g^{d}h^{a}(\text{mod}\,N) \tag{88}$$

即需使得：

$$g^{d'-d}=h^{a-a'}(\text{mod}\,N) \tag{89}$$

Bank_{Adv}若先给定a'，则问题转化为求d'，满足：

$$g^{d'-d}=h^{a-a'}(\text{mod}\,N) \tag{90}$$

Bank_{Adv}若先给定d'，则问题转化为求a'，满足：

$$g^{d'-d}=h^{a-a'}(\text{mod}\,N) \tag{90}$$

由于Bank_{Adv}并不知道陷门$p-1$的值，问题的求解难度等同于离散对数问题，因此Bank_{Adv}无法对私人审计实施欺骗.

证毕

定理 4 对于任意交易区间t_{10}~t_{11}，敌手Bank_{Adv}的支出和收入分别vout和vin，Bank_{Adv}无法伪造vout′, vin′（vout′≠vout, vin′≠vin），使监管方相信支出和收入分别为vout′, vin′，即便是对面对任意私人审计，Bank_{Adv}也无法进行欺骗.

证明 由定理3可知，对于任意给定comm=$g^{d}h^{a}(\text{mod}\,N)$，要想通过审计，受审者只能忠实地回复$(d,a)$，不存在$(d',a')$使得$\text{comm}=g^{d'h^{a'}(\text{mod}\,N)}$.

于是对于支出和收入审计：

（1）对于监管者

监管者拥有陷门信息，并且能够对交易区间t_{10}~t_{11}内的交易进行分类，能够独立将t_{10}~t_{11}内的交易进行分类，因此能够自行将各类交易的承诺相乘分别得到∏commout, ∏commin, ∏commzero, Bank_{Adv}要想通过验证只能给出正确的vout和vin进行回复.

（2）私人审计者

私人审计者不知道陷门信息，只能要求Bank_{Adv}对交易进行分类并将各类交易的序号告知私人审计者. 假设将各类交易的承诺相乘，其结果分别为∏commout, ∏commin, ∏commzero，对应被承诺值为vout, vin, vzero. 于是对于Bank_{Adv}作恶的方式为给出错误的交易分类及序号，计算错误的∏commout′, ∏commin′, ∏commzero′，最后给出错误的vout′和vin′作为回复. 现说明要想通过私人审计，必须满足：vout′=vout, vin′=vin.

由上文支出审计和收入审计中私人审计者的验证方法，Bank_{Adv}要想通过验证（a）需要使∏commout′中对应的交易全部为自己发起的转账交易，那么Bank_{Adv}可以只承认部分自己发起的交易作为∏commout′，将其余转账交易隐藏在∏commin′和∏commzero′之中. 现说明这种情况也是不可能的. Bank_{Adv}要想继续通过验证（b），必须满足∏commin′中不能含有Bank_{Adv}发起的转账交易，于是只能将其余转账交易隐藏在∏commzero′中. 但是Bank_{Adv}要想继续通过验证（d），必须满足∏commzero′中也不包含Bank_{Adv}发起的转账交易，于是通过验证必须保证∏commout=∏commout′，因此vout′=vout.

现证明vin′=vin. 对于∏commin′和∏commzero′，因为私人审计者无法对上述两类交易进行区分，Bank_{Adv}可以：①将部分接收转账的交易混入∏commzero′中，或者②将与Bank_{Adv}无关的交易混入∏commin′中. 若出现情况①，则使得∏commzero′中实际被承诺金额不为0，因此无法通过验证（e）；对于情况②，若将k个与Bank_{Adv}无关的交易混入∏commin′中，则vin′=vin+0·k=vin.

综上所述，Bank_{Adv}要想通过验证只能给出正确的vout和vin进行回复.

证毕

6 实验

实验环境配置为Win 11操作系统，16 GB RAM，CPU Intel Core i5-10400 CPU 2.90GHz-4.30 GHz. 本文中涉及到大量大整数的幂运算、乘法运算和取模运算，因此采用python语言编写实验程序. 哈希函数采用安全加密标准SHA256算法.

由于运算涉及大整数，特别是对于大整数的高次幂运算（幂运算的底数和指数均可能为大整数，实验中可达1 024位），如果不加以优化而采用通常小整数的幂运算方法，不仅运算效率很低，还容易造成内存溢出. 为此，实验时采用蒙哥马利算法实现大整数的幂运算.

图2反映了银行数量与comm和token生成时间之间的关系，安全参数k=1 024，安全参数k是指随机数的大小，以位为单位.

可以看出，随着银行数量的增长，生成comm和token的时间随之线性增加，这对于交易是不利的. 但注意到，由于交易中除交易双方外，其余银行承诺中的金额v均为0. 因此银行可以预生成大量随机数，预存v为0的comm和对应token. 在此情况下无论银行数量的多少，交易中生成comm和token的时间不超过图2中银行数量为2时生成comm和token所花费的时间.

图3表示安全参数k与方案中各操作执行时间之间的关系. k分别取64、128、256、512、1 024. cost1代表π_1（π_1用于证明生成的交易承诺是合规的）的验证时间，cost2代表π_2（π_2用于证明生成的交易不是交易金额为0的无效交易）的验证时间，cost3代表π_3（π_3用于验证交易发起方的身份）的验证时间，cost4代表π_4（π_4用于证明交易发起方有足够的资金发起交易）的生成时间，

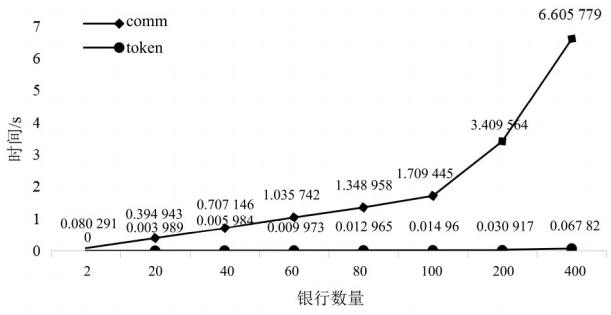

图2　银行数量与comm和token生成时间关系示意图

ver4代表π_4验证时间.

结果表明,π_1,π_2,π_3的验证时间不随k的增加而显著变化,且π_1,π_2的验证用时很短可以忽略不计;π_4的生成时间和π_4的验证时间随k的增加线性增长,在各操作中耗费时间最多.

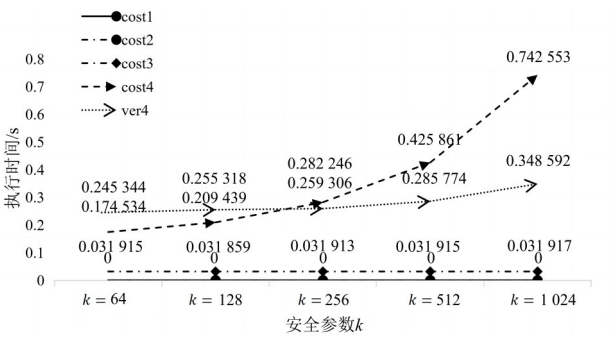

图3　安全参数k与各操作执行时间关系示意图

图4表示银行数量与方案中各操作执行时间之间的关系. 安全参数k取1 024.

可以看出,当k固定不变时,π_2,π_3的验证时间、π_4的生成和验证时间不随银行数量的增加而显著变化,只有π_1的验证时间随着银行数量的增加而线性增长.

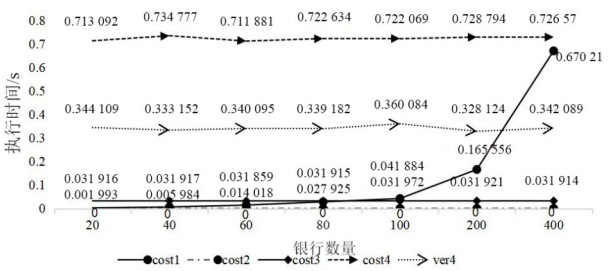

图4　银行数量与各操作执行时间关系示意图

图5表示交易数量与审计时间之间的关系. 实验时银行数为3,k=1 024. 图中横坐标为交易数量,取值别为500、1 000、2 000、5 000、10 000、50 000,对应纵坐标的4个数据从上至下依次表示资产审计(无缓存)、支出/收入审计(无缓存)、资产审计(有缓存)、支出/收入审计(有缓存)所耗费的时间. 缓存指的是4.4节中提到的优化方法,生成交易时将交易的承诺相乘,并将乘积进行缓存.

可以看出,无缓存时审计耗时随着交易数量的增加而线性增长;有缓存时,审计耗时不随交易数量的增长而显著变化,且用时很短,约为16 ms.

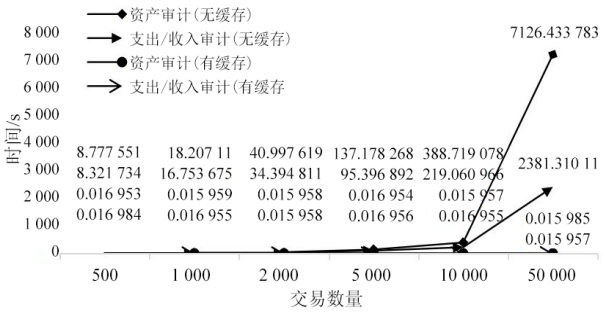

图5　交易数量与审计时间关系示意图

图6表示范围证明大小与证明生成时间之间的关系. 本文与prcash的方案相比,证明较小范围时耗时更多;随着证明范围的增大,在256位时耗时基本相同;当证明范围大于256位时,本文耗时更少,且具有明显优势.

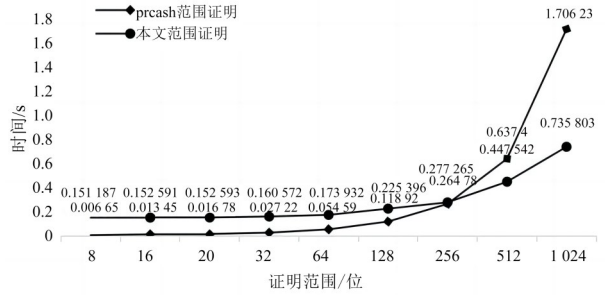

图6　范围证明大小与证明生成时间关系示意图

7　结论

本文提出了一种基于区块链的可审计隐私保护机密交易方案. 该方案采用Pederson承诺、零知识证明等密码学技术保护交易隐私,引入陷门实现多种审计功能,并且资产审计支持任意私人审计. 该方案在交易生成及验证方面具有较高的效率,在缓存条件下开展审计也较为高效. 但本方案在给出值较小的范围证明时耗时相对较高,我们将在后期研究中进一步完善.

参考文献

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system[EB/OL]. (2008-10-31) [2022-09-05]. https://nakamotoinstitute.org/library/bitcoin/.

[2] BECH M L, GARRATT R J. Central bank cryptocurrencies[J]. BIS Quarterly Review September, 2017, 1: 1-13.

[3] MILLS D, WANG K, MALONE B, et al. Distributed ledger technology in payments, clearing, and settlement[J]. Finance and Economics Discussion Series, 2016, 95: 1-36.

[4] CAROLYN A, WILKINS. Fintech and the financial ecosystem: Evolution or revolution[EB/OL]. (2016) [2022]. http://www. bankofcanada. ca/wp-content/uploads/2016/06/remarks-170616.pdf

[5] MAS. Working with industry to apply distributed ledger technology in securities settlement and cross border payments[EB/OL]. (2017) [2022]. http://www. mas. gov. sg/News-and-Publications/Media-Releases/2017/MAS-working-with-industry-to-apply-Distributed-Ledger-Technology.aspx.

[6] Koning JP. Fedcoin: A central bank-issued cryptocurrency[EB/OL]. R3 Report, 2016. https://www.scribd.com/document/422431659/FED.

[7] DANEZIS G, MEIKLEJOHN S. Centrally banked cryptocurrencies[C]//Proceedings 2016 Network and Distributed System Security Symposium. Reston: Internet Society, 2016: 21-24.

[8] STEFAN I. The e-krona and the payments of the future[EB/OL]. (2018) (2022). https://www. riksbank. se/globalassets/media/tal/engelska/ingves/2018/the-e-krona-and-the-payments-of-the-future.pdf

[9] AMANDA B. Now there are plans for 'e-krona' in cashshy sweden[EB/OL]. (2018) [2022]. https://www. bloomberg.com/news/articles/2018-10-26/riksbank-to-develop-pilot-electronic-currency-amid-cash-decline.

[10] 李智虎, 钟林, 许海清, 等. 可监管的电力区块链交易隐私保护技术研究[J]. 密码学报, 2022, 9(6): 1014-1027.
LI Z H, ZHONG L, XU H Q, et al. A supervised power blockchain transaction privacy protection system[J]. Journal of Cryptologic Research, 2022, 9(6): 1014-1027. (in Chinese)

[11] GAI K K, WU Y L, ZHU L H, et al. Privacy-preserving energy trading using consortium blockchain in smart grid[J]. IEEE Transactions on Industrial Informatics, 2019, 15(6): 3548-3558.

[12] 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[J]. 密码学报, 2022, 9(6): 1053-1066.
XIAO Y, FENG Y, LI Y N, et al. A privacy-preserved scheme for blockchain transaction based on homomorphic encryption[J]. Journal of Cryptologic Research, 2022, 9(6): 1053-1066. (in Chinese)

[13] 陈露, 相峰, 孙知信. 基于属性密码体制的区块链安全技术研究进展[J]. 电子学报, 2021, 49(1): 192-200.
CHEN L, XIANG F, SUN Z X. A survey of blockchain security technologies based on attribute-based cryptography[J]. Acta Electronica Sinica, 2021, 49(1): 192-200. (in Chinese)

[14] 佘维, 霍丽娟, 刘炜, 等. 一种可隐藏敏感文档和发送者身份的区块链隐蔽通信模型[J]. 电子学报, 2022, 50(4): 1002-1013.
SHE W, HUO L J, LIU W, et al. A blockchain-based covert communication model for hiding sensitive documents and sender identity[J]. Acta Electronica Sinica, 2022, 50(4): 1002-1013. (in Chinese)

[15] GAI K K, WU Y L, ZHU L H, et al. Differential privacy-based blockchain for industrial Internet-of-Things[J]. IEEE Transactions on Industrial Informatics, 2020, 16(6): 4156-4165.

[16] 祝烈煌, 高峰, 沈蒙, 等. 区块链隐私保护研究综述[J]. 计算机研究与发展, 2017, 54(10): 2170-2186.
ZHU L H, GAO F, SHEN M, et al. Survey on privacy preserving techniques for blockchain technology[J]. Journal of Computer Research and Development, 2017, 54(10): 2170-2186. (in Chinese)

[17] 姜轶涵, 李勇, 朱岩. ACT: 可审计的机密交易方案[J]. 计算机研究与发展, 2020, 57(10): 2232-2240.
JIANG Y H, LI Y, ZHU Y. ACT: Auditable confidential transaction scheme[J]. Journal of Computer Research and Development, 2020, 57(10): 2232-2240. (in Chinese)

[18] WÜST K, KOSTIAINEN K, ČAPKUN V, et al. PRCash: Fast, private and regulated transactions for digital currencies[M]//Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019: 158-178.

[19] TOM E J. Mimblewimble[EB/OL]. (2012-07-24) [2022-09-05]. http://mimblewimble.org/mimblewimble.txt.

[20] BUNZ B, BOOTLE J, BONEH D, et al. Bulletproofs: Short proofs for confidential transactions and more[C]// 2018 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE, 2018: 315-334.

[21] GREGORY M. CoinJoin: Bitcoin privacy for the real world [EB/OL]. (2017)[2022]. http://bitcointalk.org/index.php?topic=279249.0

[22] BENEDIKT B, BOOTLE Jonathan, DAN Boneh, et al. Bulletproofs: Effificient range proofs for confifidential transactions[EB/OL]. Technical report, Cryptology ePrint Archive,Report 2017/1066,2017.https://eprint.iacr.org/2017/1066,2017.

[23] CLINTONIO. Corda[EB/OL]. (2017) [2022]. https://github.com/corda/corda.

[24] DIGITAL ASSET. Digital asset holdings[EB/OL]. (20

17)[2022]. http://digitalasset.com.

[25] BEN-SASSON E, CHIESA A, GENKIN D, et al. SNARKs for C: Verifying program executions succinctly and in zero knowledge[C]//Annual Cryptology Conference. Berlin: Springer, 2013: 90-108.

[26] SASSON E B, CHIESA A, GARMAN C, et al. Zerocash: Decentralized anonymous payments from Bitcoin[C]// 2014 IEEE Symposium on Security and Privacy. Piscataway: IEEE, 2014: 459-474.

[27] WALSH S B, CAHN N, KUNZ C L. Digital assets and fiduciaries[M]//Research Handbook on Electronic Commerce Law. Cham: Springer International Publishing, 2016: 91-112.

[28] CECCHETTI E, ZHANG F, JI Y, et al. Solidus: Confidential distributed ledger transactions via PVORM[C]// Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2017: 701-707.

[29] NARULA N , VASQUEZ W , VIRZA M . Zkledger: Privacy-preserving auditing for distributed ledgers[C]//15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18). New York: ACM, 2018: 65-80.

[30] GOLDWASSER S, MICALI S, RACKOFF C. The knowledge complexity of interactive proof systems[J]. SIAM Journal on Computing, 1989, 18(1): 186-208.

[31] BOUDOT F. Efficient proofs that a committed number lies in an interval[C]//International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2000: 431-444.

[32] TSAI Y C, TSO R, LIU Z Y, et al. An improved non-interactive zero-knowledge range proof for decentralized applications[C]//2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON). Piscataway: IEEE, 2019: 129-134.

作者简介

盖珂珂　男,1982年生,博士,教授,博士生导师,CCF会员. 主要研究方向:区块链,数据安全,隐私计算. 中国电子学会会员编号:E190023886M.
E-mail: gaikeke@bit.edu.cn

陈思源　男,1994年生,硕士. 主要研究方向:区块链隐私保护.
E-mail: 3220200862@bit.edu.cn

祝烈煌　男,1976年生,博士,教授,博士生导师,CCF会员. 主要研究方向:密码学,网络和信息安全. 中国电子学会会员编号:E190010255M.
E-mail: liehuangz@bit.edu.cn