分类账:分布式分类账的隐私保护审计

麻省理工学院媒体实验室 Neha Narula 德克萨斯大学奥斯丁分校的威利·瓦斯奎兹；

麻省理工学院媒体实验室

https://www . usenix . org/conference/nsdi 18/presentation/naru laa

这篇论文被收录在
第十五届 USENIX 网络化研讨会
系统设计和实施(NSDI '18)。
2018 年 4 月 9 日至 11 日美国华盛顿州伦顿

书号 978-1-939133-01-4

分类账:分布式分类账的隐私保护审计

Neha Narula威利·瓦斯奎兹 Madars Virza
麻省理工学院媒体实验室 奥斯汀德克萨斯大学* 麻省理工学院媒体实验室

摘要

分布式分类账(例如。区块链)使金融机构能够有效地调节跨组织交易。例如，银行可能使用分布式分类账作为数字资产的结算日志。不幸的是，这些分类账要么对所有参与者完全公开，披露敏感的战略和交易信息，要么是私人的，但不支持第三方审计，不向审计师披露交易内容。审计和财务监督对于证明机构遵守监管至关重要。

本文介绍了 zkLedger，这是第一个保护分类帐参与者隐私并提供快速、可证明正确的审计的系统。银行创建的数字资产交易仅对参与交易的组织可见，但可以公开验证。审计员向银行发送查询，例如“您的资产负债表上某项数字资产的未偿金额是多少？并获取响应和该响应正确的加密保证。zkLedger 比以前的工作有两个重要的好处。首先，zkLedger 使用 Schnorr 类型的非交互式零知识证明为快速、丰富的审计提供了新的证明方案。与 zk-SNARKs 不同，我们的技术不需要可信的设置，只依赖于广泛使用的密码假设。第二，zkLedger 提供完整性；它使用列分类账结构，这样银行就不能对审计员隐藏交易，参与者可以使用滚动缓存快速生成和验证答案。我们实现了 zkLedger 的分布式版本，它可以在不到 10 毫秒的时间内对 10 万个交易的分类帐的审计员查询产生可证明的正确答案。

一 介绍

机构聘请可信的第三方审计师来证明他们遵守法律法规。传统上，这是由审计公司完成的，如德勤、普华永道、安永和毕马威(被称为“四大”)，他们一起审计 99%的标准普尔 500 的公司[19]。这种类型的审计既费力又耗时，因此监管机构和投资者无法实时获取有关机构财务状况的信息。此外，可信任的第三方也可能出错。这方面最著名的例子是 2002 年阿瑟·安德森公司的倒闭，因为它没能抓住安然公司 1000 亿美元的会计欺诈。

最近，金融机构正在探索分布式分类账(或区块链)，以在多方不信任的环境中降低验证和对账成本。分布式分类账支持所有参与者的实时验证(称为公共可验证性)，但以隐私为代价——每个参与者必须下载所有交易以验证其完整性。对于依靠保密来保护战略和知识产权(例如。交易策略)，以及遵守有关数据隐私的法律法规的组织(例如，欧洲的一般数据保护法规[24])。

支持隐私的分布式分类帐通常以两种方式之一运行:要么仅提交分类帐上的事务散列，使用可信第三方独立验证事务[22，23]，要么使用加密提交方案隐藏事务内容[17，42，47，51]。前一类分类账的缺点是，参与者无法再核实私人交易的完整性，从而消除了分类账的分布式效益。后一类仍然具有公开的可验证性，但要么显示交易图[17，42]，要么需要可信的设置，如果被泄露，将让对手无法察觉地创建新资产[47，51]。现有的保护隐私的分布式分类账都

*麻省理工学院媒体实验室完成的工作。

源代码和论文全文:zkledger.org。

没有为现实世界的系统提供一个重要的属性——高效审计。

本文介绍了 zkLedger，这是第一个支持强事务隐私性、公共可验证性和实用、有用的审计的分布式分类帐系统。zkLedger 提供了强大的交易隐私:对手无法知道谁在参与交易或交易量，最重要的是，zkLedger 没有显示交易图或交易之间的联系。交易时间和转移的资产类型是公开的。zkLedger 中的所有参与者仍然可以验证交易是否维护了重要的财务不变量，如资产保护，审计人员可以向参与者发出一组丰富的审计查询，并收到与分类帐可证明一致的答案。zkLedger 支持一组有用的审计原语，包括总和、移动平均线、方差、标准差和比率。审计人员可以使用这些原始数据来衡量整个系统或单个参与者的财务杠杆、资产流动性不足、交易对手风险敞口和市场集中度。

一组银行可能使用 zkLedger 为场外市场交易数字资产构建结算日志。在这些市场中，买家和卖家通过电子交易进行匹配，交易频繁，快速结算有助于降低交易对手风险。一旦交易被确认，银行可以在 zkLedger 中启动资产转移作为交易，当在分类帐中被接受时，结算交易。每个银行都将纯文本交易数据存储在自己的私有数据存储区中。在 zkLedger 中，参与者将价值承诺存储在分布式分类账中，而不是存储纯文本交易。重要的是，这些承诺可以同态组合。一家银行可以向审计人员证明，它的资产负债表上有多少资产，方法是打开它参照该资产所做的所有交易承诺的产品。审核员可以确认打开的产品与分类账上承诺的产品一致。

设计分类账需要克服三个关键挑战:

提供隐私和审计。第一个挑战是保护隐私，同时允许审计人员对分类账中的数据进行可证明正确的计算。zkLedger 是第一个通过组合几个加密原语同时实现这一点的系统。为了隐藏价值，zkLedger 使用彼得森承诺[41]。彼得森承诺可以同态组合，因此验证者可以例如确认输出之和小于或等于输入之和，从而节省资产。除此之外，审计员可以结合承诺来计算分类账中不同行值的线性组合。以前的区块链保密系统也使用彼得森承诺来隐藏价值，但最终揭示了交易之间的联系，并且不支持私人审计[17，34，42]。

zkLedger 在分类账上使用交互式地图/简化范例和非交互式零知识证明来计算超出总和的测量值。这些是广义施诺尔证明[48]，速度很快，只依赖于广泛接受的密码假设。银行可以证明对分类账中值的函数的重新承诺，如 f : v → v2，这使审计师可以计算方差、偏斜和异常值等度量，而无需披露单个交易细节。

审计完整性。由于审计师无法确定谁参与了哪些交易，zkLedger 必须确保在审计期间，参与者不能遗漏交易以对审计师隐藏资产。我们称这个性质为完备性。与此同时，我们不想向审计师透露谁参与了哪些交易。zkLedger 在分类账中使用了一种新颖的表格结构。交易是一行，其中包括每个参与者的条目，空条目与涉及资产转移的条目无法区分。参与者的所有转帐都在分类帐的相应列中。审计员在审计参与者时会审计每个交易，这意味着参与者不能隐藏交易。这带来了效率挑战，zkLedger 通过使用承诺缓存和审计令牌来解决这个问题，如下所述。

效率。第三个挑战是有效地支持所有这些。zkLedger 实现了一些优化:每个参与者和审计员都保持承诺缓存，这是分类账中每个参与者列的滚动产品；这使得生成资产证明和回答审计变得更快。为了降低通信成本，zkLedger 的设计使得参与者不必交互来构建交易的证据；支出者可以单独创建交易(这类似于其他区块链系统的工作方式)。但恶意支出者可能会试图在对其他银行的承诺中编码不正确的值——我们必须确保所有的承诺和证据都是正确的，并且每个参与者都有他们以后对审计做出反应所需的东西。为此，我们设计了一套每个人都可以公开验证的证据——使用不正确证据的交易将被忽略。这些证明确保每个参与者都有一个审核令牌，他们可以使用该令牌来打开该行的承诺，并且所有证明和承诺都是一致的。审计令牌和一致性证明是可公开验证的，但不会泄露任何交易信息。它们也是非交互式的，因此 zkLedger 即使在银行无法沟通的情况下也能取得进展，并且它们是为特定的银行编码的，因此一家银行的令牌不能被另一家银行用来对审计员撒谎。

交易创建和验证最慢的部分是范围证明，它确保资产的价值在预先指定的范围内，并防止恶意攻击者无法察觉地创建新资产。范围证明的大小是其他证明的 10 倍，并且需要 5 倍的时间来证明和验证。zkLedger 的简单实现可能需要多个范围证明，但是通过使用析取证明，我们可以将不同的值复用到每个条目的一个范围证明中。

总之，本文的贡献在于:

- zkLedger，第一个实现强隐私和完整审计的分布式分类账系统；

- 一种结合使用审计令牌和 map/reduce 的快速、易于理解的加密原语的设计，以计算可证明正确的查询答案；
- 对 zkLedger 的评估，显示有效的交易创建和审计；和
- 对 zkLedger 可以支持的查询类型的分析表明，zkLedger 可以有效地处理一组有用的审计度量。

2 相关著作

zkLedger 与区块链的私人数据审计或计算以及隐私保护工作有关。zkLedger 通过创建一个新的分布式分类账表模型和应用一个使用零知识证明的新方案来实现快速的、可证明正确的审计。

2.1 基于私有数据的计算

先前的工作提出了一个多方计算方案，其中参与者使用一个安全协议来计算函数的结果，这些函数回答了关于系统金融风险的问题，zkLedger 旨在解决的相同问题[3，10]和网络安全[14]。

这项工作通过允许参与者对其数据保密，提供了优于现有分析系统的隐私优势。但是，它只支持整体系统审计，不是审计单个参与者的解决方案。也没有什么能阻止参与者在多方计算的输入中撒谎；它们没有达到完整性。

条款[21]是比特币交易所证明自己有偿付能力的一种方式，而无需披露其总持有量。Provisions 使用资产证明和负债证明，这与我们在 zkLedger 中使用的零知识证明非常相似。然而，在条款中，交易所可以从另一个比特币持有者那里“借用”私钥，从而证明他们实际上并没有持有的资产；事实上，多个交易所可以共享相同的资产。此外，条款没有提供完整性。通过使用分布式分类帐的柱状结构，zkLedger 实现了完整性。

在 Prio [18]中，不受信任的服务器可以对移动客户端数据进行私有计算。Prio 不对分布式分类账进行操作，因此不能保证公开的可验证性。

Prio 要求所有服务器进行合作，以便客户端证据得到验证；zkLedger 可以容忍不合作的参与者。

一些系统使用可信硬件提供私有和正确的计算[4–6，49，52]。在我们的环境中，我们不能保证所有参与者都信任同一个硬件提供商。此外，使用这样的系统来审计提供可信硬件的公司会产生利益冲突。

有许多系统根据加密数据进行计算，以在服务器受损的情况下保护用户的机密性[25，31，32，40，43，44，50]。这些系统解决的问题与 zkLedger 试图解决的问题不同。相反，我们对多方生成的私有数据提供交互式的、可证明正确的审计。2.2 保护隐私的区块链

比特币是 2009 年发行的分散加密货币，是第一个区块链[37]。许多公司已经探索使用区块链记录资产转移。这些系统具有以下特征:(1)多个，可能是不信任的参与者，所有人都有写权限，没有单点故障或控制；(2)一个一致的协议，用一串散列来构造一个仅附加的、全局有序的日志，以防止篡改过去；以及(3)数字签名交易，以表明转让所有权的意图。

在比特币和大多数其他区块链货币中，所有交易都是公开的:每个参与者都会收到每笔交易，并且可以核实所有细节。用户通过为支付地址生成一次性使用的公钥来创建假名，但是交易金额和交易之间的链接仍然是全局可见的。机密交易[34]和机密资产[17，42]是比特币的扩展，它可以屏蔽交易中的资产和金额，同时仍然确保所有参与者都可以验证交易。虽然这些系统隐藏资产和金额，但它们会泄露交易图，并且不支持私人审计——审计人员需要访问所有纯文本交易，以确保完整性。仅交易图就泄露了大量信息[36、38、45、46]；例如，联邦调查局跟踪关联交易来追踪比特币，并在法庭上将其作为证据[28]。zkLedger 提供了更强的事务隐私和私有审计，但代价是可伸缩性。zkLedger 中的事务是根据整个系统中的参与者数量来确定大小的，随着参与者数量的增加，需要更多的时间来生成和验证。这使得 zkLedger 更适合参与者较少、需要更多隐私的分类账。Solidus [16]是一个分布式分类帐系统，它使用遗忘内存，隐藏银行客户之间的交易图和交易金额。虽然这种结构也提供了私有事务，但 Solidus 只能通过向审计人员显示系统中使用的所有密钥并打开事务来支持审计。zkLedger 在提供私有审计的同时，实现了类似 Solidus 的性能。

R3 的 Corda [22]和 Digital Asset Holding 的全球同步日志(GSL) [23]是面向依赖可信第三方传递信息的金融机构的分布式分类账。在 Corda，公证人验证交易并维护参与者的隐私，而 GSL 对其分类账进行分类，只在全球范围内存储一个散列值，并限制对细粒度交易数据的访问。两者都不支持私人审计。

另一种方法是零现金[47]及其相关实现 Zcash [51]，这是一种基于比特币的匿名加密货币。Zerocash 使用 zk-SNARKs [7]来隐藏交易金额、参与者和交易图。Zcash 中使用的 zk-SNARKs 可以扩展到处理政策以执行法规、KYC/反洗钱法律和税收[27]。这些策略不支

持任意查询，而是对可能发生的新型事务进行限制。这些想法尚未在实际系统中实施。

zk-SNARKs 对于某些语句来说是相当有效的，但不幸的是，这种效率的代价是在设置假设中付出的:到目前为止，所有具体有效的 zk-SNARKs 都需要可信的第三方来进行设置。不正确或被破坏的设置的后果是潜在的灾难性的:一个能够了解在设置过程中使用的秘密随机性的对手可以制造虚假陈述的欺诈证据，而这些虚假陈述的证据与真实陈述的证据是无法区分的。在我们的环境(国际银行业)中，这种证明将允许无限制地创建或销毁金融资产或负债。甚至可能没有可行的一方来执行一次性可信设置。例如，俄罗斯可能不信任美联储(Federal Reserve)或欧洲央行(European Central Bank)，或者被视为这样做在政治上可能不合适。虽然有可能减轻这种担心，例如，。通过在多方之间分配设置[8，11，12]，这个过程是繁重和昂贵的。理想情况下，系统的财务完整性根本不依赖于可信的设置。我们选择将 zkLedger 设计的共识关键部分建立在标准 NIZKs 的基础上。

3 分类帐概述

3.1 体系结构

系统参与者。有 n 个参与者，我们称之为银行，他们发布交易来转移数字资产，银行 1，...,。Bankn 和一名审计员，负责验证银行执行的交易的某些操作方面(例如。"某家银行有偿付能力吗？"). 这些角色并不明显；银行也可以审计。一个存款人或一组存款人可以从系统中发放和提取资产；例如，欧洲中央银行可能向系统中的银行发行 100 万欧元。所有资产的发行和提取由存款人控制，是全球性的公共事件。

交易。银行通过创建转移交易来交换资产，转移交易的细节是隐藏的。转让交易记录了 Banki 将资产 t 的 v 股转让给 Bankj 的事件。我们的方案支持一家银行向多家其他银行转账，但为简单起见，我们假设每笔交易中有一家支出银行和一家收款银行。银行可能通过交易所决定 zkLedger 之外的转账交易细节。我们假设他们使用加密频道。

仅限追加的分类帐。银行将交易提交给仅附加的分类帐，该分类帐在全球范围内对所有有效交易进行排序。如果数字资产只存在于分类账中，那么分类账中的转移是数字资产合法保管的变更，而不仅仅是所有权变更的记录，并且审计员可以保证银行没有隐藏资产。这种分类账可以由可信的第三方、银行自己或者通过像以太网或比特币这样的区块链来维护。维护容错的全局有序日志不在本文讨论范围之内，但可以使用标准技术来完成[15，30，39]。

3.2 密码构造块

承诺计划。为了保护他们的隐私，参与者银行不公开支付细节，如交易金额。取而代之的是，银行将隐藏的承诺过帐到只追加的分类帐；特别是，zkLedger 使用彼得森承诺[41]。设 G 是 s =|G|元素的循环群，设 G 和 h 是 G 的两个随机生成元。然后是对整数 v∑{ 0，1，...,。*S1 }的形式如下:选择承诺随机性 r，并返回承诺 cm := COMM(v，r) = gvhr*。

彼得森承诺是完美隐藏的:承诺 cm 没有揭示承诺价值 v。以类似的方式，承诺在计算上也是有约束的:如果一个对手可以用两种不同的方式打开一个承诺 cm(对于相同的 r，两个不同的值 v 和 v0)，那么同一个对手可以用来计算 logh(g)，从而打破 g 中的离散对数问题。在 zkLedger 中，我们选择 G 为椭圆曲线 secp256k1 上的点群。

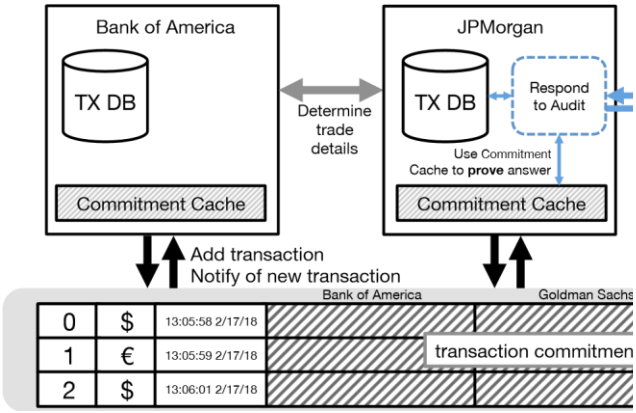
彼得森承诺的一个非常有用的性质是它们是可加同态的。如果 cm1 和 cm2 是 v1 和 v2 值的两个承诺，分别使用承诺随机性 r1 和 r2，则 cm := cm1 cm2 是使用随机性 r1+r2 对 v1+v2 的承诺，如 cm =(gv1hr 1)(gv2hr 2)= gv1+v2hr 1+R2.为了加快交易生成和审计，zkLedger 广泛使用了额外组合承诺的能力。

公钥加密。每个银行 I 还生成一个 Schnorr 签名 keypair keypair，它由一个密钥 ski 和一个公钥 pki := hski 组成，并将公钥 pki 分发给所有其他系统参与者。

非交互式零知识证明。zkLedger 依靠非交互式零知识证明来对支付细节进行隐私保护
(NIZKs) [9]。简而言之，零知识证明涉及两方:证明者和验证者，前者持有一些私有数据，后者希望确信这些私有数据的某些属性。例如，证明者可能知道承诺 cm 的开始，并且希望说服验证者承诺值 v 在某个范围内，。0 ⩽ v < 106.使用 NIZKs，证明者可以产生一个二进制字符串π，即证明，它同时说服了验证者，但没有揭示关于 v 的任何其他东西。验证 π 不需要证明者和验证者之间的任何交互，证明者可以将 π 附加到分类帐中，在那里它可以被系统的任何一方验证。

理论上，NIZK 证明系统存在于 NP 中的所有性质，而 NIZK 的实际可行性在很大程度上取决于手头性质的复杂性。特别是循环群中的代数性质，如离散对

数的知识，Pedersen 承诺中承诺的值的等式，或类似的有非常有效的 NIZK 证明系统。zkLedger 的设计是精心构造的，因此所有的 NIZK 证明都有特别有效的构造。



3.3 安全目标

zkLedger 的目标是隐藏交易的金额、参与者和链接，同时维护一个可验证的交易分类帐，并让审计员收到对其查询的可靠答案。具体来说，zkLedger 允许银行发布隐藏的转账交易，这些交易仍然可以由所有其他参与者公开验证；每个参与者都可以确认交易保存了资产，并且资产只有在支出银行的授权下才能转移。比如 Banki 转账 10,000 C 给 Bankj，银行和金额都是隐藏的。交易的资产(C)和时间不隐藏。zkLedger 还隐藏了交易图，这意味着先前的交易首先向 Banki 提供了 10，000 加元。

审计员可以向银行查询分类账中的内容，例如“银行持有多少欧元？银行应该能够做出承诺，使审计师相信银行对审计问题的回答是正确的，这意味着与分类账上的交易一致。zkLedger 确保如果银行给审计员的答案与分类账不一致，审计员将很有可能发现这种欺诈企图(当然，必须始终接受可信的答案)。

3.4 威胁模型

zkLedger 并不认为银行会诚实行事——它们可以试图窃取资产、隐藏资产、操纵账户余额或对审计人员撒谎。我们假设银行可以任意串通。zkLedger 将交易的金额和参与者保密，只要交易中的支出者和接收者都不与观察者(如审计者)串通。我们假设分类账没有遗漏交易，并且是可用的。zkLedger 无法抵御观察网络流量的对手；例如，如果只有两家银行在交换信息，那么假设分类账中的交易涉及这些银行是合理的。除了审计必然泄露的内容外，没有任何内容被泄露。但是，频繁的审计可能会泄露交易内容；例如。如果审计员在每次交易后都要银行的资产。

四 设计

创建分类账的挑战是实际支持完整、保密的审计——审计员不应能够看到单个银行交易，但银行不应在审计期间向审计员隐藏资产，审计员应能够发现错误答案。

图 1 显示了 zkLedger 的概述。有些银行决定带外交易，然后通过将交易附加到分类帐来结算。分类账确保所有银行和任何审计师都能看到新的交易。每个银行和审计员都维护一个承诺缓存，承诺缓存是对用于创建交易和更快响应审计的合计值的承诺。每个银行也有明文交易数据的私有存储。

本节的其余部分描述了 zkLedger 事务格式，银行如何创建事务，以及银行如何回答审计员的简单查询。

4.1 处理

zkLedger 中的分类帐是一个表，其中交易对应于行，银行对应于列。每笔交易都有一个银行账户。图 2 显示了一个有 n 个银行的分类账。交易中的每一项都包括对某一价值的承诺，即借记或贷记给银行的资产金额。例如，如果 Banki 想将一项资产的 100 股转让给 Bankj，I 在交易中的分录将包含对-100 的承诺，j 的分录将包含对 100 的承诺.交易中的所有其他条目将包含对 0 的承诺，因为没有任何其他银行余额被更改。这个方案有一个很好的特性，外部观察者可以看到一家银行的整个专栏，并且知道这代表了该银行的全部持股。

隐藏金额。如 3.2 所述，zkLedger 不包括交易中的纯文本值。相反，zkLedger 使用彼得森承诺来承诺转让交易中的价值。这使得价值承诺完全无法区分——外部观察者无法区分对正值、负值或 0 的承诺.回想一下，对值 v 的承诺是 cm := COMM(v，r) = gvhr。如果需要，证明者可以向知道 cm 的验证者揭示 v 和 r，并且验证者可以确认这是一致的。

因为 zkLedger 中的事务包含每个银行的条目，所以有一个大小为 n 的向量 cm~承诺~v 中的值。每个承诺 cmk 使用一个新的承诺随机性 rk。对于不参与交易的银行来说，大多数条目将包含 0 承诺，但对于外部观察者来说，这并不明显。

zkLedger 维护以下财务不变量:转移交易不能创建或破坏资产

支出银行必须同意转账，并且必须实际拥有足够的特定资产来执行该交易

在公共区块链，验证者可以通过查看交易历史和当前交易来确认这些事情是真实的，并确保支出银行有资金支出。然而，在 zkLedger 中，这些值是不公开的。相反，我们创建一组证明，消费者可以创建这些证明来证明不变量得到了维护。消费者可以在不与任何其他银行交互的情况下创建交易。

首先，zkLedger 引入了一个余额证明(πB)。这是交易保存资产的证明；没有资产被创造或破坏(当然，公开发行和撤回交易没有这样的证据)。更正式地说，承诺的价值观应该让$\sum_{k=1}^{n} v_k = 0$满意。为了证明这一点，证明者小心地选择 rk:这也应该是$\sum_{k=1}^{n} r_k = 0.$的情况.如果这是真的，并且值的和也是 0，则验证者可以检查以确保该行中的承诺的∏nk=1cmk = 1。

接下来，zkLedger 必须确保支出银行实际上拥有要转移的资产。为此，zkLedger 引入了资产证明(πA)。其他保护隐私的区块链系统使用未使用的交易输出，或 UTXOs，以显示资产证明和防止双重支出。例如，如果爱丽丝想给鲍勃寄一枚硬币，她选择她的一枚硬币，创建一个新的交易，将硬币发给鲍勃，并包括一个指向她收到硬币的前一次交易的指针。前一个事务是一个输出。系统中的所有验证器都保持输出只能使用一次的不变性。不幸的是，在没有 zkSNARKs 的系统中，这会泄露事务图。在 zkLedger 中，银行通过在其列中创建对资产价值总和的承诺来证明其拥有资产，包括这笔交易。如果总和大于或等于 0，则银行有资产要转移。请注意，这是正确的，因为银行的列代表它已经收到或花费的所有资产，彼得森承诺可以同态地添加到列和行中。为了提供正确金额的证明，银行必须已经看过之前的每笔交易。这意味着银行必须连续创建交易。在自己的条目中，如果该值为负，银行会提供密钥知识证明，以表明它授权了该交易。这需要创建一个分离证明——要么条目 I 的提交值 vi ≥ 0，要么事务的创建者知道 Banki 的密钥。

范围证明。因为承诺值在椭圆曲线组中，并且依赖于模数，所以我们需要确保承诺值在可接受的范围内。要看原因，注意如果 N 是群的阶，那么 COMM(v，r) = COMM(v+N，r)；没有办法区分这两者。如果没有检查以确保承诺值在[0，n1]范围内，恶意银行可能无法察觉地创建资产。为了解决这个问题，我们使用了机密资产[42]中描述的范围证明，它使用了博洛曼环签名[35]。zkLedger 支持高达一万亿的资产价值。范围证明是交易中最昂贵的部分；如上所述，我们的方案需要两个范围证明——一个用于承诺值，另一个用于列中资产的总和。通过引入辅助承诺 cm0i，我们可以将两个范围证明压缩为一个范围证明。cm0i 要么是对 cmi 中值的重新承诺，要么是 m 行之前的列中值的总和,∑mk=0vk，这可以通过计算列中承诺的乘积,∏mk=0cmk 来实现。然后，我们可以对 cm0i 中的值做一个范围证明。要么是支出银行，在这种情况下，cm0i 必须是对该金额的承诺，要么是另一家接受资金或不参与的银行，在这种情况下，cm0i 可以是任何一家(无论是哪家)。

这满足了上述金融不变量。然而，我们在 zkLedger 中做出的一个特殊的设计选择是，一家支出银行可以在不与其他银行交互的情况下创建一项支出自己资产的交易。这意味着恶意银行可能会创建保持金融不变量但格式不正确的交易。我们将在描述审计如何工作后解决这个问题。

一旦创建，银行就会广播该交易，并将其附加到分类帐中。如果银行维护分类账，则每个银行负责在接受交易到分类账之前验证交易。如果由第三方维护分类账，则第三方应在接受交易之前核实交易证明。

交易示例。图 2 显示了美国银行向高盛转移 100 万欧元的交易。美国银行创建转账交易，公开发布交易 id、时间戳和资产类型(欧元)。美国银行承诺从其自身资产中扣除的金额，在自己的分录中为 1，000，000 英镑，在高盛分录中为 1，000，000 英镑。对于其他每家银行，美国银行承诺为 0.这是为了隐藏参与转账的银行；除了美国银行之外，没有人能够区分这些承诺，以确定哪些承诺是非零值。然后，美国银行将交易广播到分类账。分类帐维护者验证交易并将其附加到分类帐中。一旦被记入总账，这相当于从美国银行向高盛转账 100 万欧元
萨克斯。

4.2 审计协议

审计员有一份分类账的副本，并与银行互动，计算其私人数据的功能，以便了解分类账所代表的金融系统。

审计员通过向银行发出查询来审计银行，例如，“你在时间 t 持有多少欧元？Banki 用一个答案和一个证明回答与分类账上的交易一致的证据来回应审计员。审计员将银行分类账中的承付款项乘以欧元，并用总数验证证明和答案。这是对银行所持欧元总额的承诺。

这里的关键见解是，给定这个表的结构，审计员可以阅读银行 I 的专栏，并知道它正在查看涉及 I 的每一笔资产转移。在没有实际转移资产并把控制权交给另一家银行的情况下，我没有办法在分类账中“隐藏”资产。相比之下，在传统的审计过程中，银行不能简单地向审计员展示一些资产负债表。

银行可以相互勾结，暂时为对方持有资产；例如，Bankj 可能会将资产转移到 Banki，然后再收回。在这段时间内，这些资产将成为班克控股的一部分。但是，在审计员提出查询后，银行不能串通，因为审计员已经指定了查询适用的时间 t。任何转移都必须在 t 之后。所以在这一点上，一家恶意银行创建一个新的交易将资产转移到另一家银行已经太晚了。

如上所述，银行可以在没有任何交互的情况下创建转移其自身资产的交易。这在大多数区块链系统中是常见的，在这些系统中，创建有效的交易只需要发送者的签名。接收方没有协议内的方式来反对传输。考虑到我们的表结构，每个银行都会受到每个交易的影响，因为银行必须

	Metadata		$Bank_1$	$Bank_2$	$Bank_3$	...	$Bank_n$
ID	Asset	Time	(Bank of America)	(Goldman Sachs)	(JPMorgan)	...	(Citigroup)
1	e	13:06:01 2/17/18	COMM(-10^6,r_1) $Token_1$ $\pi_1^A, \pi_1^B, \pi_1^C$ $\pi_1^A, \pi_1^B, \pi_1^C$	COMM(10^6,r_2) $Token_2$ $\pi_2^A, \pi_2^B, \pi_2^C$ $\pi_2^A, \pi_2^B, \pi_2^C$	COMM(0,r_3) $Token_2$ $\pi_3^A, \pi_3^B, \pi_3^C$ $\pi_3^A, \pi_3^B, \pi_3^C$	⋯ ⋯ ⋯	COMM(0,r_n) $Token_n$ $\pi_n^A, \pi_n^B, \pi_n^C$

Figure 2: Contents of the ledger pertaining to a transaction that sends 10^6 euros from $Bank_1$ to $Bank_2$. Note that while asset type (euro) is visible as part of the metadata, the transaction amount (10^6 e) and participating institutions (Bank of America and Goldman Sachs) remain private. We use Pedersen commitments, so the commitment part of the row has values $g^{-106}h^{r1}$, $g^{106}h^{r2}$,and h^{r3}, …, h^{rn}. Similarly, audit tokens pictured have values $(pk_1)^{r1}$, $(pk_2)^{r2}$, …. For each bank $Bank_i$, the corresponding column also includes proof-of-assets π_i^A, proof-of-balance π_i^B, and 一致性证明 πiC。

总计其栏中回应审计员的所有承诺，甚至包括与其无关的交易承诺。恶意的 Banki 可以创建一个交易，并且不通知另一个 Bankj 在其条目中使用的 rj，即使它没有将资产转移给 Bankj。银行将无法对审计员作出回应，因为它无法在其专栏中公开承诺的产品。

为了证明转移交易的完整性，zkLedger 必须确保一个额外的不变量:

所有银行都有足够的交易信息，可以向审计员公开承诺

zkLedger 通过要求支出银行在每个条目中包含一个可公开验证的令牌来实现这一点。这被定义为 Tokenk := (pkk)rk。班克用这个令牌为审计员打开其承诺的产品，而不需要知道 rk。

使用审计令牌。考虑查询银行列中的值的总和。回答这个问题的一种方法是揭示∑vk 和∑rk。然后审计员将简单地检查这些简单值是否与同态计算值相一致。

然而，银行不一定知道所有的承诺随机数 rk(特别是，对于银行没有参与的任何交易，这些值是未知的)，因此天真的方法不起作用。

一种方法是询问交易的编制者(即。发送者)对 rk 进行加密，以便非参与银行 Bankk 可以对其进行解密。为了防止发送银行将“垃圾”密文放在分类账上(从而使班克无法通过审计员的查询)，人们需要加密值和承诺之间一致性的零知识证明。为这种说法构造一个具体有效的证明并不简单:简而言之，标准加密方案(例如。ElGamal)在组元素中嵌入纯文本，而 Pedersen 承诺在指数中有这个值。

我们的洞见是，班克不需要开∑rk 来证明∑vk 是正确的。相反，假设 Bankk 想要声明 s = g∑vkh∑rk 打开一个值∑vk。为此，银行计算 s0 = s/g∑vk = h∑rk，t = ∏k Tokenk = hsk∑rk。请注意，审计员也可以从分类账和声称的答案∑vk 计算 s0 和 t 的值。

银行只要证明 logs0 t = logh pk 就够了。请注意，两个对数的计算结果都是 sk，因此银行可以在不知道∑ri 的情况下提供这一证明。此外，如果这个方程成立，那么 t1/sk = s0 = s/g∑v，但是如果∑v 是不正确的，那么 sk 的知识将揭示 g 和 h 之间的线性关系，这被我们的安全假设排除了。

为了证明令牌中的 r 与 rk 中的 r 相同，我们需要一个额外的一致性证明(πC)。这是一个零知识证明，断言对于每个 k，用于形成 cmk 和 Tokenk 的值 rk 是相同的。(关于这种证明的细节，见附录二构建。

请注意，审计令牌仅对开立承诺的银行有用；尽管是公开的，但恶意银行不能使用另一家银行的令

牌成功地打开不正确的结果或了解其他银行的交易信息。

4.3 最终交易结构

对于 m 行中的转账交易，每个分录 I 包含以下项目:

- 承诺(CMI):(g vihri)Pedersen 对我们正在传递的价值的承诺。
- 审核令牌(Tokeni): (pki)ri。这是用来回答审计，而不知道承诺中使用的随机性。
- 平衡证明(πB):一个零知识证明，主张承诺的价值满足$\sum_{k=1}^{n} v_k = 0$.
- 资产证明(πA):新的承诺 cm0i、相应的令牌 0i 和零知识证明，证明 cm0i 是对 cmi 中的值的重新承诺，或者是对∏mj=0cmj 中的值之和的重新承诺，并且 cm0i 在范围[0，240]内。如果 cmi 中的承诺值为负，则证明声明银行 I 同意转账。
- 一致性证明(πC):断言 cmi 和 Tokeni 中使用的随机性的两个零知识证明是相同的，cm0i 和 Tokeni 中使用的随机性也是相同的。这是为了防止恶意银行将数据添加到分类账中，从而阻止另一家银行为审计员打开其承诺。

事务可能以明文形式包含附加元数据，也可能不包含。例如，银行可能希望代表客户包括加密的账号、地址或识别信息，以满足 1970 年《银行保密法》中规定的旅行规则[1]。zkLedger 也支持对事务中的元数据进行审计，但是它没有公开验证附加元数据的方法。

4.4 添加或删除银行

zkLedger 可以支持动态添加或删除银行，如果这样做是公开的。参与者(或另一个机构)将签署的交易附加到分类账中，指明应该添加或删除哪些银行以及哪些列。例如，要向图 2 所示的分类帐中添加一家新银行，所涉及的银行会向分类帐中追加一笔交易，表明要添加 Bankn+1.从那时起，所有事务都应该包含 n + 1 个条目。银行 n+1 在每笔交易中的资产证明将从银行 n+1 被添加的行开始。同样，如果一家银行被撤销，以后的交易不应包括该银行的分录。由于所有参与者都可以看到哪些银行被添加或删除，因此他们可以相应地调整他们的证明和验证。

4.5 优化 zkLedger 采用了几种优化来加快这些证据的生成和验证，并支持更快的审核。首先，在银行的专栏中缓存承诺的产品可以提高审计和凭证创建的速度。每家银行都按行和按资产存储一个滚动的承诺产品，这样它就可以快速生成资产证明并回答审计人员的询问。使用这些缓存，银行可以快速回答审计员对分类账中行子集的查询。

zkLedger 中的大多数交易并不包括每一家银行。每个银行都可以为值 0 预先生成许多范围证明.我们通过并行化范围证明生成和验证来加快事务吞吐量。

5 审计

审计是金融体系的重要组成部分，监管者使用各种技术来衡量系统性金融风险。通过使用总和、平均值、比率、方差、共同方差和标准偏差，zkLedger 中的审计员可以在其他测量中确定以下内容:

- 杠杆比率。zkLedger 可以显示一家银行的账面资产与其他资产相比有多少。这有助于估计交易对手风险。
- 专注。监管者使用一种叫做赫芬达尔-赫希曼指数(HHI)的方法来衡量一个行业的竞争力。
- 实时价格指数。审计人员可以了解场外交易资产的价格，因此不会通过交易所进行跟踪。

zkLedger 本身支持求和，这意味着分类帐中存储的值的线性组合。这来自彼得森承诺的加法结构。但是 zkLedger 还支持更通用的查询模型，可以分两部分考虑:一个 map 步骤和一个 reduce 步骤。

基础审计。考虑一个基本的例子，一个审计师想要确定一家银行账面上有多少资产。如 4.2 所述，审计师将按资产过滤行，将银行列中的条目相乘，然后要求银行打开承诺产品。这仅需要审计员和银行之间的一轮通信，并且消息是恒定大小的，与行数无关。由于 zkLedger 的承诺缓存，这是非常快的。

映射/缩小。审计员可以发出更复杂的查询，这可能需要交换更多的数据，或者可能需要参与者查看分类帐中的大多数行。让我们考虑一个审计师，他想知道给定银行和资产的平均交易规模。审计员不能简单地通过合计银行的承诺栏和用开放值除以行数来验证银行的答案，因为这样的计算会有不正确的分母。也就是说，当银行没有参与交易时，其在该行的列将被承诺为 0，并且应该被贴现。为了确定正确的分母，审计师和银行运行以下协议:1 .过滤。银行将按资产过滤行。

2. 产生新的承诺。对于每一行，银行将承诺单个位 b，1 或 0，这取决于银行是否参与交易，并创建一个银行已正确完成此再承诺的证明。至关重要的是，审计师无法区分这些承诺，因此银行的交易不会被披露。我们称这种产生新承诺的行为为地图步骤。映射步骤还需要生成新值计算正确的

证明；在我们的例子中，对于每一笔交易，当且仅当交易值不等于 0 时，银行将产生一个 b = 1 的 NIZK 证明。

3. 计算非零交易的数量。银行计算对位~b 的新承诺的同态和，并打开它来显示有多少交易是非零的。这是减少步骤。这是计算平均交易规模的正确分母。除了总和所揭示的以外，审计员不能告诉任何关于~b 中的值的事情。
4. 回应审计员。然后，银行向审计员发送其列中值的总和、位承诺向量和相应的 NIZK 证明、其非零交易数 n 以及承诺中 r 值的总和。
5. 验证。**审核员通过验证承诺是否正确完成来验证映射步骤，并通过确认位承诺向量的乘积是** gnh $\sum_{k=1}^{N} r_k$**来验证缩减步骤和非零事务的数量。**
6. 计算答案。审计员根据银行列和非零交易数的总和计算平均值。

审计师可以用类似的方法向银行询问异常交易。对于每一行，如果该行的事务值超出指定范围，则该行将提交位 b，其中 b = 1。当计算平均值时，审计员可以验证这些承诺是否正确，并获得总和。然后，银行只能打开 b = 1 的交易，审计员确切知道应该打开多少交易。

更复杂的审核查询需要多个映射并减少计算。例如，以下是审计员如何了解交易值 v1 的变化，...,。*vN:*

1. 计算平均交易价值。执行上述协议，计算非零事务数 n 及其平均值 v。
2. 应用平方图。对于其行中的每个条目 vi，银行生成新的承诺 cm0i 至 v2i，并将这些承诺发送给审计师。银行还提供 NIZK 证据，证明每个 cm0i 中隐藏的价值正好是总账中承诺的 vi 值的平方。
3. 应用减少步骤。**审计员计算承诺** cm0i **的乘积，银行通过揭示以下内容，将这一承诺公开**为 $= v_1^2 + \cdots + v_N^2$ $= \sum_{i=1}^{N} \bullet$ *里*。审计师确认承诺的乘积等于 gVhR。

审计员现在计算方差 σ 如下:

$$\sigma^2 = \frac{1}{n}\sum_{v_i \neq 0}(v_i - \bar{v})^2 = \frac{1}{n}V - \bar{v}^2.$$

我们注意到，尽管上面使用的平方映射对应于第二矩(方差)，zkLedger 也可以计算更高的统计矩(例如。偏斜度和峰度),并分别使用立方和四次幂映射。关于 zkLedger 支持的测量值列表，请参见附录 A。

zkLedger 可以通过以下方式支持有限的信息发布使用更复杂的简化映射。例如，银行可以对取整后的总值做出承诺(例如。到前两位小数)，并使用范围证明(也在 zkLedger 中实现)来表明舍入是正确的。仅仅披露手头数量的数量级就能让各方平衡信息披露的粒度。

6 履行

为了评估 zkLedger 的设计，我们在 Go 中实现了 zkLedger 的原型。我们的原型使用 btcec 库[2]的修改版本，其中包含用椭圆曲线 secp256k1 计算的参数和方法.我们使用 Go 的内置 SHA-256 实现来实现我们的加密散列函数，并通过对“我的袖子里什么都没有”字符串 SHA256(0)和 SHA256(1)应用点解压缩来确定性地选择 g 和 h。我们的原型由大约 3200 行代码组成，其中 40%实现了 zkLedger 使用的加密工具(零知识证明、范围证明等)。

zkLedger 中曲线的实现使用 Go 的 big.Int 类型，我们并不努力以高效的方式压缩或序列化它。更优化的实现可以压缩曲线点。我们的范围证明实现了机密资产中使用的协议[42]。我们的 NIZKs 是基于广义 Schnorr 证明的，是三种移动交互协议；为了使它们不具有交互性，我们应用了 Fiat-Shamir 启发式算法[26]，其中我们使用 SHA256 散列函数来实例化随机 oracle。我们的原型实现没有实现 5 中描述的复杂查询，因此我们没有在 7 中评估它们。

七 估价

我们的评估回答了以下问题:

#	成分	创造	核实	大小
2k	承诺	0.5 毫秒	0.5 毫秒	64B
2k	一致性	0.7 毫秒	0.8 毫秒	224B
k	分离的	0.9 毫秒	0.9 毫秒	288B
k	范围	4.7 毫秒	3.5 毫秒	3936B

- zkLedger 中不同证明的存储、证明、验证费用有多高？(7.2)。
- 审计如何随着分类账的大小而变化？ (7.3)。
- zkLedger 如何随着银行数量的增加而缩放？ (7.4)。

表 1:k 家银行交易中每个凭证组件的数量。创建和验证 12 核组件的大小和时间。范围验证创建并验证了额外内核的优势。

7.1 实验装置

微基准。我们在 12 核 Intel 机器上运行微基准测试，该机器具有 i7-X980 3.33 GHz CPU 和 24GB RAM，在 Ubuntu 上运行 64 位 Linux 4.4.0 16.04.3。 每个微基准运行与银行创建和验证交易相同的代码。

分布式实验。我们在一组 12 个虚拟机上运行分布式实验，每个虚拟机具有 4 个英特尔至强 E5-2640 2.5 GHz 处理器内核、24GB 内存和与上述相同的软件设置。有一个审计员，一个提供分类账服务的服务器，以及不同数量的银行，每个服务器一个。服务器通过 TCP 使用网络/rpc Go 包进行通信。所有实验都使用 Go 版本 1.9。

7.2 分类帐中的证明费用

表 1 显示了在 zkLedger 中证明和验证交易证明的时间。在一个交易条目中，有两个承诺、两个一致性证明以及一个析取和范围证明。每个银行都有一个交易条目。表 1 还显示了各种证明的大小，以字节为单位。这些大小是根据内存结构中底层字段的大小来估计的；这些证据可以进一步压缩。范围证明决定了交易的规模。

图 3 中的左图显示了创建和验证一个事务所需的时间，它改变了整个银行的数量，从而增加了每个事务的条目数。这表明，随着银行数量的增加，每个银行的事务创建和验证时间都线性增加，但并行化有所帮助。证明和验证范围证明在事务创建和验证中占主导地位，但这种成本也是高度并行化的。12 个内核在创建与 20 家银行的交易时带来 2.8 倍的加速；一家银行可以在不到 200 毫秒的时间内为多达 20 家银行创建或验证交易。

如 4.1 中所述，zkLedger 使用 Borromean 环签名来证明一个值在某个范围内，并且支持高达 240 的值。减少支持的值范围将降低范围验证成本，因为该成本与范围大小中的位数成线性关系。也有更新的证明系统，如 Bullet 校样，它可以创建更小范围的证明[13]。我们计划
在以后的工作中，用布勒特证据评估 zkLedger。

7.3 审计分类账的成本

图 4 中的左图显示，对于某些功能，审计时间与分类账中的交易数量无关。这是因为审计员和银行维护承诺缓存，该缓存已经具有向审计员证明其列中值的总和所必需的承诺产品。审计职能是衡量赫芬达尔-赫希曼指数，因此审计员与每家银行沟通。

当审计员不能使用承诺缓存时，可能是因为它处于离线状态，它必须处理整个分类帐来计算承诺产品。这也适用于更复杂的审计，如 5 中描述的类型，当审计员必须验证分类帐中每行的重新承诺时。这些成本显示在图 4 的中间图表中.此图显示了在不使用承诺缓存的情况下，审计员在不同大小的分类帐上计算赫芬达尔-赫希曼指数需要多长时间，因此审计员必须处理分类帐的每一行。在这些测量中，审核员不验证每一行。不出所料，此时间随着行数线性增加。这表明维护提交缓存对于实时审计很重要。但是，即使没有提交缓存，审核时间也是合理的:100K 事务 3.5 秒。这表明，复杂的审计查询(审计员在其中每行计算一组相似的操作)也需要几秒钟的时间。zkLedger 目前只维护每家银行每项资产的承诺产品缓存，但可以维护更多。

对于固定大小的分类帐，此审计功能按银行数量排序。图 4 中的右图展示了在 2000 笔交易的分类账上计算赫芬达尔希尔施曼指数的审计成本，因为我们改变了银行的数量，包括有无承诺缓存。审计员同时审计银行。此功能的审计成本随着银行数量的增加而略有增加，因为更多的银行增加了并行审计的可变性，审计员必须等待最后一家银行做出响应，然后才能计算最终答案。

在这些图中，每个点是运行审计查询 20 次的平均值，误差线代表平均值的一个标准偏差。

7.4 与更多银行合作

随着 zkLedger 中银行数量的增加，有两个显著的成本增加:创建线性增加的交易的连续步骤，以及验证随银行数量二次增加的交易。如同

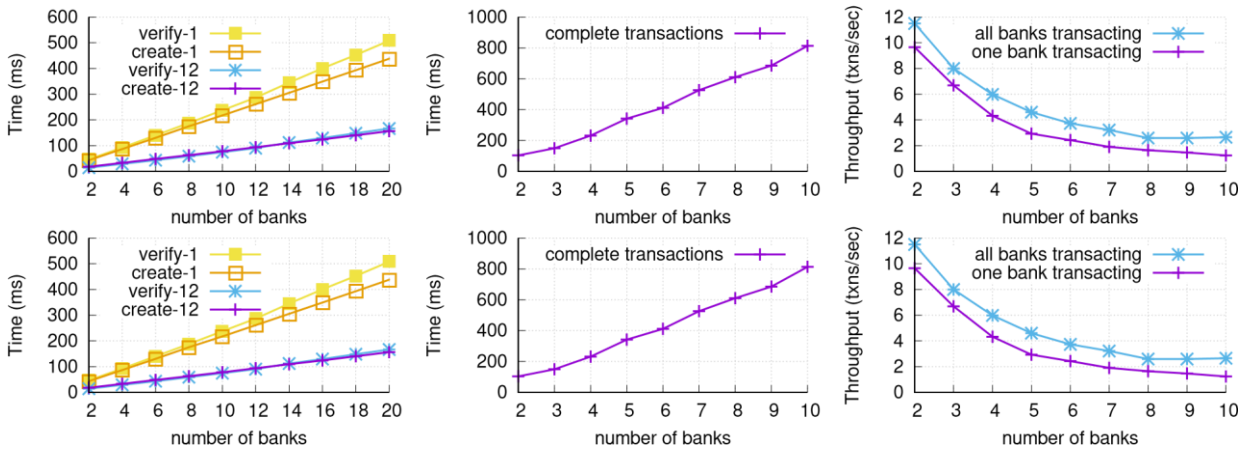


Figure 3: Transaction creation and verification time for one bank (left), varying the number of entries in the transaction. Single-threaded and multi-threaded performance, with 12 threads. Time to fully process a transaction including creation, broadcast to ledger, banks and auditor, and verification by all parties (middle). Throughput (right) varying the number of banks.

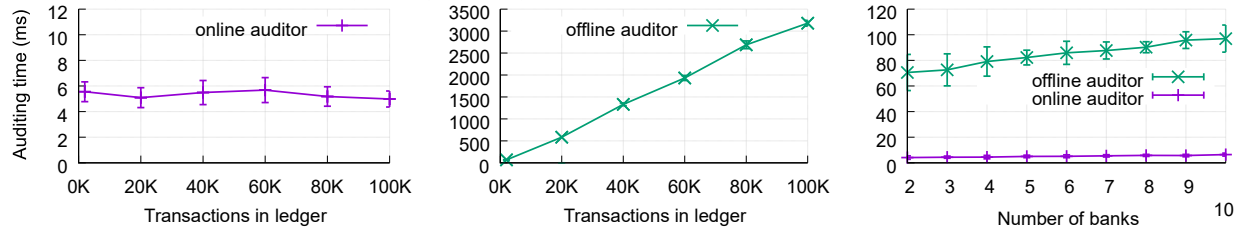


Figure 4: Time to audit ledgers of different sizes (4 banks), and with a varying number of banks (2000 row ledger). Audit time is independent of the size of the ledger (left) thanks to commitment caches maintained by the online auditor. When commitment cache optimization is turned off (middle) the audit time is linear in the size of the ledger. Audit time grows with the number of banks (right) and is much higher without commitment caches.

如第 4.1 节所述，银行需要使用交易 n1 的分录来创建交易 n。因此，尽管一家银行可以使用多个内核并行生成单个事务的证明，但多家银行不能并行生成不同的事务。在 zkLedger 中，银行在看到 n-1 之前开始创建交易 n，但在 n-1 被分类帐接受并验证之前，银行无法完成交易，这导致了一个固有的瓶颈。

第二大成本是围绕验证。每家银行都必须核实每笔交易，所以银行越多，每笔交易就越大，因此每家银行需要做的工作就越多。图 3 中的中间图表测量了一家银行创建和 zkLedger 中的所有参与者完全处理一个事务所需的时间。一家银行创建一笔交易，并将其发送到分类账，然后分类账将交易广播给所有银行和一名在线审计师。审计员和每个银行核实交易。随着银行数量的增加，工作量呈二次增长；但是，银行可以并行验证交易，因此处理交易的时间只会线性增加。图 3 中的右图显示，正如我们猜测的那样，随着银行数量的增加，zkLedger 的吞吐量会下降。该图中的一条银行交易线与中间图中的数据相同。

由于范围证明控制了事务创建和验证的成本，我们乐观地认为更快的范围证明实现将直接提高性能。zkLedger 目前的性能与 Solidus 相当，Solidus 是一种保护隐私的分布式分类帐，通过在线验证每秒可实现 3-4 笔交易，但与 zkLedger 不同，它不支持审计。

8 今后的工作

zkLedger 专注于对私有事务数据提供可证明正确的审计，但是如果分布式分类帐损坏，zkLedger 没有办法恢复。在这种情况下，维护分类账的各方必须走到一起重新创建历史交易。zkLedger 也不提供追索权，如果银行承诺对分类账进行非预期的交易。zkLedger 的未来版本可能会提供更正交易或参与者同意的回滚。

9 结论

zkLedger 是第一个分布式分类帐系统，提供强大的交易隐私性、公共可验证性和完整的、可证明正确的审计。zkLedger 支持一组丰富的审计查询，这些查询有助于衡量市场的财务健康状况。我们开发了一个使用非交互式零知识证明来证明交易保持财务不变量和支持审计的设计。我们的评估表明，zkLedger 在交易结算和审计方面具有合理的性能。

10 承认

我们感谢亚力山大·切尔尼亚霍夫斯基，萨德斯·德里亚，大卫·耶戈，罗纳德·l。Rivest，C.J. Williams，以及我们的牧羊人和评论者的有益评论。导致这些结果的研究获得了以下机构的资助:美国国家科学基金会科学技术中心信息科学中心(CSoI)，其授权协议为CCF-0939370；和人工智能基金的伦理与治理。

参考

[1]1970 年银行保密法，1970 年 10 月.12《美国法典》第 103 条.[2] Package btcec 实现了对比特币所需椭圆曲线的支持。2017 年 7 月.https://godoc.org/·github.com/btcsuite/btcd/btcec.

[3] 阿贝，e。A.,。KHANDANI，a . E.,。还有 LO，a。W. 分享金融风险暴露的隐私保护方法。*《美国经济评论》102，3 (2012)，65–70*。

[4] ARASU，a。BLANAS，。EGURO，k，。考斯克，。科斯曼，d。RAMAMURTHY，r . 和文卡特森。基于密码的正交安全性。在 CIDR (2013 年)。

[5] BAJAJ，s 。 和西昂。可信数据库:一个可信的基于硬件的数据库，具有隐私和数据保密性。*IEEE 知识与数据工程学报 26，3 (2014)，752–765*。

[6] 鲍曼，a。PEINADO，。还有亨特，g。使用 haven 保护应用免受不可信云的影响。*美国计算机学会计算机系统学报(TOCS) 33，3 (2015)，* 8.

[7] BEN-SASSON，e 。 CHIESA，a，。GENKIN，d . TROMER，e，。和维尔扎，m。C 语言的缺点:在零知识的情况下简洁地验证程序的执行。《密码学进展——加密 2013》.Springer，2013，pp . 90–108.

[8] BEN-SASSON，e 。 CHIESA，a，。格林，m，。TROMER，e，。和维尔扎，m。简洁零知识证明的公共参数安全抽样。《2015 年电气和电子工程师协会安全和隐私研讨会论文集》(2015 年)，第 15 页。287–304.

[9] BLUM，m 。 费尔德曼，。和米卡里。非交互零知识及其应用。在第 20 届美国计算机学会计算理论年会论文集(1988)中，STOC '88，页。103–112.

[10] BOGDANOV，d 。 TALVISTE，。威廉姆森。为金融数据分析部署安全的多方计算。在金融密码学和数据安全国际会议(2012 年)，斯普林格，页。57– 64.

[11] BOWE，s 。 GABIZON，a . 绿色，m。一个多方协议，用于构建皮诺曹 zk-SNARK 的公共参数。密码学电子档案，报告 2017/602，2017。

[12] BOWE，s 。 GABIZON，a . 还有迈尔斯，我。随机信标模型中 zk-SNARK 参数的可扩展多方计算。密码学电子档案，报告 2017/1050，2017。

[13] BUNZ，b。BOOTLE，。BONEH，d . POELSTRA，a，。还有麦克斯韦，g。bullet 校样:用于机密交易等的简短校样。在安全与隐私(SP)，2018 年 IEEE 研讨会(2018)，IEEE。

[14] BURKHART。STRASSER，。很多，d，。还有迪米特罗普洛斯，x。Sepia:多域网络事件和统计的隐私保护聚合。*网络 1，101101 (2010)*。

[15] CASTRO。还有利科夫，b。实用拜占庭容错。《OSDI》(1999 年)，第一卷。99 页。173–186.

[16] CECCHETTI，e 。 张， 女，。JI，y，。KOSBA，a，。JUELS，a，。和 SHI。Solidus:通过 pvorm 进行的机密分布式分类帐交易。

[17] CHAIN， 爱达荷（Idaho 的缩写）。机密的 资产。https: //blog . chain . com/hidden-in-plain sight-transaction-private-on-ablockchain-835 ab 75 c 01 CB。

[18] CORRIGAN-GIBBS，h。还有 BONEH，d。Prio:私有的、健壮的、可伸缩的聚合统计计算。*arXiv 预印本 arXiv:1703.06255 (2017)*。

[19] 理事会，f . R. 2016/2017 年度审计进展报告全文，2017 年.http://www.frc.org.uk/ getattachment/915 c 15 a 4-DBC 7-4223-b8 aecad53 dbcca 17/Developments-in-audit 2016-17-Full-report . pdf。

[20] CRAMER。damgard \\,我，。还有 SCHOENMAKERS，b。证人隐藏协议的部分知识证明和简化设计。《密码学进展》，第 14 届国际密码学年会，美国加州圣巴巴拉，1994 年 8 月 21-25 日，会议录(1994)，页。174–187.

[21] DAGHER。G.,。BUNZ，b，。BONNEAU，。CLARK。还有 BONEH，d。条款:比特币交易所偿付能力的隐私保护证明。在第 22 届美国计算机学会计算机和通信安全会议记录(丹佛，科罗拉多州，2015 年)，美国计算机学会，页。720– 731.

[22] Corda，2017。https://github.com/corda/·科达。

[23] 数字资产控股，2017。http://digitalasset。com。

[24] 欧洲议会和理事会 2016 年 4 月 27 日关于在处理个人数据和此类数据自由流动方面保护自然人的第 2016/679 号条例(欧盟)，并废除第 95/46/EC 号指令(一般数据保护条例)。*《欧盟官方公报》L119(2016 年 5 月)，1–88*。

[25] FELDMAN。J.,。泽勒，w。页（page 的缩写）,。FREEDMAN，m . J.,。和费尔顿。W. Sporc:使用不受信任的云资源的群组协作。《OSDI》(2010 年)，第一卷。10 页。337–350.

[26] 菲亚特，a。还有沙米尔，a。如何证明自己:身份识别和签名问题的实用解决方案？《第六届国际密码学年会论文集》(1987)，《密码》87，页。186–194.

[27] GARMAN。格林，m，。还有迈尔斯，我。分散匿名支付的可问责隐私。密码学电子档案，报告 2016/061，2016.http://eprint.iacr.org/2016/061。

[28] GREENBERG。美国联邦调查局称，已从丝绸之路的所谓所有者罗斯·乌尔布里克特手中没收了 2850 万美元的比特币。*福布斯 25 (2013)*。

[29] HERFINDAHL，o . C. *钢铁行业的集中度*。纽约哥伦比亚大学博士论文，1950 年。

[30] LAMPORT，l 。 以及其他人。帕克斯做得很简单。*ACM Sigact 新闻 32，4 (2001)，18–25*。

[31] LI，j 。 KROHN。名词（noun 的缩写），。MAZIERES，。还有沙莎，D. E. 保护不受信任的数据存储库(sundr)。在 OSDI (2004)，卷。4 页。9–9.

[32] MAHAJAN，p 。 SETTY，s，。李，s，。克莱门特，。ALVISI，l，。DAHLIN，。还有沃尔费什，m。仓库:最少信任的云存储。*美国计算机学会计算机系统学报(TOCS) 29，4 (2011)，12*。

[33] MAURER，u . 统一知识的零知识证明。*第二届非洲密码学国际会议论文集(2009)，272–286*。

[34] MAXWELL，g . 机密交易。*https://people . xiph . org/Greg/secretary _ values . txt(2017 年 8 月访问)(2015)*。

[35] MAXWELL，g 。 还有 POELSTRA，a。博罗门戒指签名。*https://raw.githubusercontent。com/Blockstream/borro men _ paper/master/borro men _ draft _ 0.01 _ 34241 bb。pdf(2017 年 6 月查阅)(2015 年)*。

[36] MEIKLEJOHN，s 。 POMAROLE，m . 约旦，g . LEVCHENKO，k . MCCOY，。VOELKER，g . 米（meter 的缩写）），。和SAVAGE，s . 一把比特币:描述没有名字的人的支付方式。《2013 年互联网测量会议论文集》(2013 年)，ACM，页。127–140.

[37] NAKAMOTO，s . 比特币:点对点电子现金系统，2008 年。

[38] OBER。KATZENBEISSER，s . 还有哈马谢尔，k。比特币交易图的结构和匿名性。*未来互联网 5，2 (2013)，237–250*。

[39] ONGARO，d 。 和驱逐出去。K. 寻找一个可以理解的共识算法。在 USENIX 年度技术会议上(2014)，页。305–319.

[40] PAPADIMITRIOU，a 。 BHAGWAN，。北卡罗莱纳州 CHANDRAN，。RAMJEE，。HAEBERLEN，a，。SINGH，。莫迪，a，。和 BADRINARAYANAN，s。海底加密数据集的大数据分析。《在 OSDI》(2016 年)，页。587– 602.

[41] PEDERSEN，t . 页（page 的缩写）。非交互式信息论安全可验证秘密共享。《第 11 届国际密码学年会论文集》(1992)，《密码 91》，页。129–140.

[42] POELSTRA，a 。 BACK，a，。FRIEDENBACH，。MAXWELL，g，。和 WUILLE，p . 机密资产，2017 年.第四次比特币和区块链研究研讨会。

[43] POPA。A.，。华盛顿雷德菲尔德，。新泽西州泽尔多维奇，。还有巴拉克里希南，h。CryptDB:用加密的查询处理来保护机密性。在《20 世纪议程》中*第三届美国计算机学会操作系统原理研讨会(2011 年)，美国计算机学会，页*。85–100.

[44] POPA。A.，。斯塔克，。HELFER，j . 瓦尔迪兹，。新泽西州泽尔多维奇，。KAASHOEK，m . F.，。巴拉克利什南，H. 使用聚酯薄膜在加密数据上构建网络应用。在 NSDI (2014 年)，页。157–172.

[45] REID，f 。 哈里根先生。比特币系统中的匿名性分析。社交网络中的安全和隐私。Springer，2013，pp . 197–223.

[46] 罗恩，d。还有沙米尔，a。完整比特币交易图的定量分析。在金融密码学和数据安全国际会议(2013 年)，斯普林格，页。6–24.

[47] SASSON，e . B.，。CHIESA，a，。加曼，。格林，m，。MIERS，I，。TROMER，e，。和维尔扎，m。零现金:比特币分散匿名支付。在安全和隐私(SP)，2014 年电气和电子工程师协会研讨会(2014 年)，电气和电子工程师协会，页。459–474.

[48] SCHNORR。-P。智能卡高效签名生成。*密码学杂志 4，3 (1991)，161–174*。

[49] SCHUSTER。COSTA，。FOURNET，。GKANTSIDIS，C。 PEINADO，。MAINAR-RUIZ，g，。俄罗斯-NOVICH，m . Vc3:使用 sgx 在云中进行可信的数据分析。在安全和隐私(SP)，2015 年电气和电子工程师协会研讨会(2015 年)，电气和电子工程师协会，页。38–54.

[50] TU，s。KAASHOEK，m . F.，。MADDEN，s . ZEL--DOVICH，n . 处理加密数据的分析查询。《VLDB 基金会学报》(2013 年)，第一卷。6，VLDB 捐赠，页。289–300.

[51] Zcash，2017。http://z.cash。

[52] 郑，w。戴夫，阿，。BEEKMAN，j . G.，。POPA。A.，。GONZALEZ。E.，。斯托伊卡，我。不透明:一个被遗忘和加密的分布式分析平台。《在 NSDI》(2017 年)，页。283–298.

A 审核查询

图 5 是 zkLedger 支持的测量类型列表，包括估计运行时间和超出泄漏测量的数据。例如，如 5 中所述，计算交易规模差异需要泄露平均交易规模和每家银行的交易数量。

B 零知识 证明 和 隐私保证

为了建立我们的零知识协议，我们依赖于莫勒的以下一般结果(定理 3，[33]):

定理 B.1。*让(H1，？和(H2,⊗)是两个(不一定可交换的)群，f : H1 → H2 是群同态:f(x？y) = f(x)⊗f(y)。设`∈ Z，u ∈ H1，C ⊂Z 是这样的:*

1. 所有 c1、c2 ∈ C 的 gcd(C1 C2，`)c1 6 = c2)，以及

2. f(u) = z `。

语言存在一个 2-可提取 σ 协议

Figure 5: Types of supported auditing queries, their running time to audit based on the number of banks k and the number of rows in the ledger n, 尺寸

	Time	Additional information leaked
Sum total of asset(s) per bank	$O(1)$	none
Outlier transactions per bank	$O(n)$	none
Concentration	$O(k)$	Sum totals per bank
Ratio holdings	$O(k)$	Sum totals per bank, number of transactions per bank
Mean transaction size per bank	$O(kn)$	Number of transactions per bank
Variance, skew, kurtosis	$O(kn)$	Mean per bank, number of transactions per bank
Real-time price averages	$O(kn)$	Number of transactions and average per bank over time period t

L := {z : ∃w s.t . z = f(w)}。此外，一份协议和一份关于哪些信息被泄露给审计人员的说明。

如果1/|C|s 可以忽略，则s 轮的存在是知识的证明，如果|C|是多项式有界的，则是零知识的证明。

使用定理 B1，我们现在可以统一我们系统中使用的大多数零知识证明的处理。例如，一致性证明 πC 依赖于以下结果:

定理 B.2。*设 G 为 r 阶循环群，G，h，pk 为 G 的任意三个元素。语言 Laux 有一个 2 可提取的 σ 协议:= {(cm，Token) : ∃v,r s.t .* cm = gvhr ∧Token = pkr}。

*证明。*考虑 H1 = Zr ×Zr，定义群运算为分量加法，设 H2 = G×G，同样定义群运算为分量加法。那么 f(x，y) := (gxhy，pky)是 H1 和 H2 之间的群同态.事实上，f(x1+x2，y1+y2) = (gx1+x2hy1+y2，pky1+y2) = f(x1,y1)⊗ f(x2，y2)。此外，设置`= r 和 u = (0，0)我们知道，对于所有的 z ∈ H2，以下成立:z` = (1，1) = f(u)。因此，我们可以应用定理 B.1，得出 Laux 有一个 2-可提取 σ-方案。□

总而言之，zkLedger 中与承诺 cmi := gvihri 和审计令牌 Tokeni := (pki)ri 相关的三个证明具有以下形式:

- 资产证明(πA)。该证明由新的承诺 cm0i、审核令牌 Token0i 和零知识证明组成，零知识证明断言 cm0i 是对 cmi 中的值的重新承诺，或者是对∏mj=0cmj 中的值的总和的重新承诺。为了创建这个证明，zkLedger 依赖于定理 B.1 的成分证明；由于这些都是西格玛协议，我们应用标准的或组合[20]来获得最终的析取零知识证明。为了证明承诺值在范围内，我们使用了机密资产[42]中的范围证明。我们正在研究更近的证明系统(例如。bullet 校样[13])以进一步减小校样大小。
- 天平证明(πB)。在我们的实现中，这个证明是一个空字符串:证明者简单地选择承诺随机性服从条件∑ri = 0.通过这样的选择，审计员同态地添加承诺，并检查这种添加是否导致组的中性元素∏cmi = g∑vih∑ri = g0h0 = 1。
- 一致性证明(πC)。我们使用从定理 B.2 导出的两个证明来断言 cmi 和 Tokeni 中使用的随机性是相同的，cm0i 和 Tokeni 中使用的随机性也是相同的。

C 组合系统中的隐私

彼得森承诺提供信息论隐私。在 zkLedger 中，彼得森承诺与认证令牌和零知识证明一起发布。我们注意到零知识证明确实不会破坏承诺值的信息论隐私:零知识证明模拟器的输出与系统中各方产生的输出相同。然而，当结合彼得森承诺和认证令牌时，隐私保证就变得像我们现在解释的那样具有计算性。

承诺、审计令牌和公钥三元组(cm，token)的形式是(gvhr，pkr，pk) = (gvhr，hsk r，hsk)，这三个值唯一地确定了 v。也就是说，如果对手可以破解离散对

数问题，它可以为 sk 求解，用那个和 Token 的值来推断 r，最终恢复 v。也就是说，在决策迪菲-赫尔曼(DDH)假设下，没有信息被泄露。此外，DDH 假设被广泛认为适用于 zkLedger 的椭圆曲线组。

回想一下，对于随机选择的生成器 h 和指数 a、b、c，DDH 认为如果没有多项式有界的对手能够区分(h，ha，hb，hab)和(h，ha，hb，hc)形式的元组。假设有状态的对手 AzkL，当给定输入(g，h，pk)时，能够产生两个值 v1 和 v2，使得它能够将承诺(和相关联的审计令牌)v1 与承诺(和审计令牌)v2 区分开，即。对手能够区分分布(gv1hr、hsk r、hsk)和(gv2hr、hsk r、hsk)。我们现在展示如何使用 AzkL 来构建一个打破 DDH 假设的对手 ADDH。

在接收到它的挑战(h，x，y，z)后，其中(x，y，z)或者作为(ha，hb，hab)或者作为(ha，hb，hc)分发，对手 ADDH 如下进行。它对随机生成器 g 进行采样，并在输入(g，h，x)上调用 AzkL，x 现在充当银行公钥的角色。当 AzkL 返回两个值 v1 和 v2 时，DDH 对手 ADDH 挑选一个随机的 k∑{1，2}，准备 cmk = gvky，Token = z，并将(cmk，Token)发送给 AzkL。最后，如果 AzkL 对 k 的猜测是正确的，ADDH 回答 DDH 挑战的形式是(h，ha，hb，hab)(即。DDH 翻两番)，否则它会回应 DDH 挑战赛的形式为(h、ha、hb、hc)(即。随机四倍)。

请注意，当 ADDH 的挑战是 DDH 四重挑战时，zkLedger 的对手 AzkL 运行在它期望的发行版上。特别是，它的所有输入都是相对于 sk = a 和 r = b 正确形成的。然而，当 ADDH 的挑战是一个随机的四元组时，AzkL 的输入有信息——理论上没有关于提交值的信息:事实上，Token = hc 与 cm = gvhb 无关。因此，如果 zkLedger 的对手 AzkL 以不可忽略的优势赢得承诺隐藏游戏，那么 ADDH 在 DDH 游戏中也是如此。请注意，该证明通过标准混合参数扩展到多条目情况。