北京理工大學

本科生毕业设计（论文）外文翻译

外文原文题目： Blockene: A High-throughput Blockchain Over Mobile Devices

中文翻译题目： Blockene：移动设备上的高吞吐量区块链

基于区块链的出租车调度系统的完善

The perfection of the taxi dispatching system based on blockchain

学　　院：计算机学院

专　　业：计算机科学与技术

班　　级：07111908

学生姓名：蒙思洁

学　　号：1120193602

指导教师：陆慧梅

Blockene：移动设备上的高吞吐量区块链

摘　要

我们引入了 Blockene，这是一种区块链，它将成员节点的资源使用量减少了几个数量级，只需要智能手机即可参与区块验证和共识。尽管 Blockene 是轻量级的，但它提供了高吞吐量和可扩展到数百万参与者的规模。Blockene 在智能手机中消耗微不足道的电池和数据，使数百万用户能够在没有激励措施的情况下参与区块链，以集体诚实的方式确保交易安全。Blockene 通过一种基于将存储和八卦传播委派给不受信任节点的新型分离-信任设计来满足这些要求。我们通过原型展示了 Blockene 提供 1045 个事务/秒的吞吐量，并在智能手机上以非常低的资源使用率运行它，这为构建安全、分散的应用程序指明了一个新的范例。

目 录

第 1 章 介绍

区块链提供了一种强大的系统抽象：它们允许相互不信任的实体（成员）以分散的方式共同管理交易账本。

如今，所有区块链都需要成员节点运行具有大量网络、存储和计算资源的强大服务器。基于工作证明的区块链将资源使用推到了极致，需要大量的计算来解决难题，但即使是基于股权证明的联盟链和区块链也会产生大量的网络和存储成本，以使区块链在高交易吞吐量下保持最新状态。因此，如今的区块链仅限于成员具有强烈参与动机的用例，因此可以承担高昂的资源成本。例如，在联盟链中，业务效率提高，而在加密货币中，成员赚取货币。

有趣的是，区块链的高资源需求也削弱了一些真实应用的可靠性。区块链要求大多数 (通常是三分之二) 成员是诚实的，当大量成员参与时，这种特性更容易得到保证。然而，考虑到高资源需求，区块链的广泛采用是困难的，特别是在成员没有直接参与激励的情况下。毫不奇怪，如今成员数量较多的公共区块链以加密货币为目标。在本文中，我们介绍了 Blockene，这是一种超轻量级、大规模的区块链，它为现实世界的交易提供了高吞吐量。凭借其轻量级和可扩展性，它能够被数百万用户广泛采用。通过实现大规模的参与，Blockene 使多数诚实的假设成为可能。凭借高吞吐量，Blockene 支持真实世界的事务速率。Blockene 的关键突破在于，它不再要求成员运行强大的服务器，而是成为第一个使成员能够以一流公民身份参与共识的区块链，即使在像智能手机这样的轻量级设备上运行，也可以降低数量级的成本。

网络：区块链依赖于成员之间的 p2p gossip 算法；在高交易速率下，八卦传播每天需要数十 GB 的数据传输；Blockene 在智能手机上每天只需要 60MB 的数据传输。

存储：区块链中的成员节点保留整个区块链的副本（高吞吐量时为 TB）；在 Blockene 中，成员只需要几百 MB 的存储空间。

计算：即使是典型区块链的八卦传播成本也会耗尽移动节点的电量；Blockene 确保电池每天的消耗量低于 3%。

因此，用户在运行 Blockene 时不会产生明显的成本。由于 Blockene 的资源使用率较低，即使在智能手机上也是可行的，因此 Blockene 也可以在台式机上运行，其资源使用比最先进的设备要少得多。

Blockene 实现了三个相互冲突的财产：大规模参与、高吞吐量和轻量资源使用，迎合没有直接激励的情况（例如利他主义参与），以及处理包括公共资金在内的各种用途的交易。Blockene 与其他区块链架构的比较如表 1 所示。

示例应用：经审计的慈善事业。全世界每年向非营利组织的慈善捐款超过 5000 亿美元。然而，从捐赠者的角度来看，资金最终使用缺乏透明度，使得捐款容易被非营利组织利用或管理不善，特别是在监管执法不力或因腐败而受损的地区。一个提供从捐赠者到最终受益人的公开、端到端资金跟踪的系统，除了激励捐赠者之外，还会对非营利组织施加市场压力。区块链可以提供这种跟踪，但考虑到所涉及的资金规模，一个小的成员联盟不能被信任来运营区块链。理想情况下，这样的区块链应该由数百万公民无私地共同控制。政府/公共支出也有类似要求。

Blockene 的关键技术:Blockene 采用了一种新颖的系统设计，该系统基于一种新的安全模型——分离信任架构。区块链中有两种类型的节点：公民节点和政治节点。公民在智能手机上运行，是区块链的真正成员，即他们在共识协议中拥有投票权；因此，我们假设三分之二的公民是诚实的（这是数百万公民的合理假设）。另一方面，政客在服务器上运行，不受信任，即不参与共识。政客人数较少（只有几百人），我们只要求其中 20% 的人诚实。尽管政客们做着存储区块链等繁重的工作，但我们的协议能确保即使 80% 的政客与三分之一的恶意公民勾结，公民也能够检测和处理恶意行为。公民通过使用一种被称为复制可验证读取的新原语来应对政客的高度不诚实：公民从多个政客那里读取相同的数据，即使其中一个（共 25 个政客）是诚实的，也可以得到正确的数据。

公民通过运行拜占庭共识来执行事务验证，并决定要提交的区块和由此产生的全局状态。为了让数百万公民达成共识，Blockene 借用了 Algorand 的一个想法（经过修改，使其电池友好），在那里，一个不同的（约 2000）公民随机委员会被加密选择为每个区块运行共识。与 Algorand 不同，Blockene 在委员会成员参加会议前几分钟将他们的信息公开: 这使 Blockene 能够降低在 “公民” 上的数据和电池成本。虽然这似乎增加了对委员会进行有针对性攻击的机会，但我们在 §4.2 中讨论了为什么这不是一个严重的问题。

为了降低公民的存储/通信成本，只有政治家才能存储区块链和全球状态（即关键值对），从而使公民不必八卦传播所有区块（约 50GB/天）。公民只从政客那里读取一小部分数据（例如，当前区块交易的关键值），然后写出新区块。此外，由于政

客不受信任，公民无法依赖他们返回的最新正确值，例如给定的密钥。Blockene 使用了一种基于采样的 Merkle 树读/写的新技术，该技术降低了通信成本，同时确保了对 80% 的恶意政客的容忍度。

在委员会中，公民通过政治家而不是直接八卦传播来减少他们的沟通成本; 公民写的数据在政治家中被八卦传播，而感兴趣的公民从政治家那里阅读。

由于 Blockene 的参与是轻量级的，因此系统需要防止 Sybil 攻击；防止对手旋转大量虚拟节点以获得不成比例的投票份额。

为了阻止此类攻击，Blockene 要求参与者身份由大多数智能手机中可用的可信硬件（TEE）认证，并强制每个 TEE 在区块链上最多只能有一个活动身份，从而将参与的经济成本提高到唯一智能手机的成本。

为了限制 80% 的恶意政客对性能造成的损害，Blockene 采用了几种技术来限制他们说谎的能力。首先，我们使用一种称为预声明承诺的技术来检测一些恶意行为。第二，为了在 80% 的人不诚实的情况下，在政客之间可靠而高效地传播八卦传播，我们引入了一种被称为优先八卦传播的新技术。这些技术降低了政客的成本，使 Blockene 能够在智能手机上运行的情况下实现高吞吐量。

我们已经建立了一个 Blockene 的原型，公民节点实现为一个 Android 应用程序，政客节点实现为一个云服务器。我们从各个维度评估了 Blockene，并表明它实现了 1045 个事务/秒（6.8 MB/分钟）的良好事务吞吐量，同时确保了第 99 百分位的提交延迟为 270 秒。我们还展示了政客具有非常少的数据使用（61MB/天）和电池使用（3%/天）。

本文的主要贡献如下：

- 我们提出了第一个区块链系统，在该系统中，节点可以作为一流成员参与共识，同时在智能手机等轻量级设备上运行，支持高规模的成员和高吞吐量。

- 我们提出了一种新的分离信任设计，该设计具有一个新的安全模型，该模型由资源受限的公民（诚实的大多数）和资源密集的政客（不诚实的大多数）组成，公民通过以可验证的方式将繁重的工作交给不受信任的政客来执行验证和共识。

- 我们进行了一些新颖的优化（例如，预先声明的承诺、基于采样的 Merkle 树读/写、优先级八卦传播），尽管 80% 的政客是恶意的，但这些优化仍能取得良

好的性能。

- 通过深入的理论分析，我们证明了 Blockene 满足安全性、活性和公平性。
- 我们对该架构进行了全面的实证评估，证明了其作为共享可扩展的区块链服务的可行性。
- 本文的其余部分结构如下：在 §2 中，我们提供了区块链的背景知识，并在 §3 中讨论了现有的区块链架构。§4 概述了 Blockene 及其威胁模型，§5 介绍了其设计。我们在 §6 中讨论了对资源密集步骤的优化，在 §7 中提供了安全性和活性证明，并在 §8 中描述了它的实现。我们在 §9 中评估了 Blockene，并得出结论（§10）。

第 2 章 背景

在本节中，我们将讨论区块链中的关键原理和抽象及其应用。

2.1 基本属性

区块链是交易的分布式账本。在没有可信机构（例如银行）管理账本的情况下，一组相互不可信的各方共同验证交易，并维护一致的账本，前提是至少有一定数量的参与者（例如三分之二）是诚实的。区块链必须提供安全性、活跃性和公平性。安全性确保诚实的参与者对账本有一致的看法。活跃性确保恶意参与者不能通过阻止新的块添加来无限期地拖延区块链。公平性确保提交到区块链的所有有效交易最终得到提交。

2.2 建立区块链

区块链是一个复制的、对等的分布式系统，建立在以下基本原理之上:

用于全局状态的 Merkle 树：区块链的一个关键部分是跟踪键及其当前值的全局状态数据库。这种全局状态以防篡改的方式进行管理，通常使用 Merkle 树，其中叶节点包含键-值对，而每个中间节点包含子节点连接内容的哈希散列。根是表示整个状态的单个哈希值。键的更新只需要沿着从该叶到根的路径重新计算哈希。给定根，任何键的值都可以通过到根的有效散列路径来证明。

签名交易：区块链中的基本工作单位是交易。一次交易读取并更新全局状态中的一些键（例如，从 Alice 向 Bob 转账 1000 美元）。为了有效，（a）交易必须是被签名的（b）给交易签名的用户必须有权访问密钥（c）“语义”完整性必须通过（例如，不能超支）。

加密链接：区块链是区块的列表。块是事务的列表。通过加密链接确保块的排序；每个块都嵌入前一块内容的加密哈希散列。

八卦传播：处于区块链交换状态的参与者以点对点的方式相互交换。例如，当一个新块被提交到账本里时，它必须被发送给其他成员。这种交流是通过多跳八卦传播进行的，并最终保持一致。

共识协议：区块链中的关键原语是分布式共识协议，它处理拜占庭故障（例如，PBFT、Nakamoto 或 BBA），因为少数参与者可能是恶意的。拜占庭式共识要求至少

2/3 的参与者诚实，并需要多轮沟通。

第 3 章 与现有区块链的比较

在本节中，我们简要介绍了现有区块链架构的相关工作。Blockene 提供了三个特性：轻量级资源使用、大规模参与和高事务吞吐量。我们使用相同的三个维度将 Blockene 与相关工作进行比较。

3.1 成员节点使用的资源

现有的区块链在参与成员节点的资源使用方面具有广泛的范围，这取决于用于一致性的机制。我们首先讨论成员的计算成本，然后讨论网络和存储成本。

计算成本。就计算成本而言，最昂贵的是基于 Nakamoto 共识的区块链，也称为工作证明；例如比特币和以太坊。在 Nakamoto 共识中，第一个解决计算密集型密码难题的成员节点被选为提交新块的获胜者。因此，这样的区块链在成员节点处需要大量的计算资源。

为了解决工作证明区块链的高计算（和能源）成本，出现了两种流行的替代架构。第一种是联盟链（例如 HyperLedger ），通过将区块链成员限制在少数节点，它可以运行传统的拜占庭共识算法，而不是基于计算密集型工作证明的共识。第二种体系结构是股权证明区块链，它将成员节点的投票权与成员节点在区块链上的资金数量联系起来。这些区块链的例子有 Algorand、Ouroboros、PeerCoin 等。从本质上讲，股份证明区块链针对的是加密货币应用程序，这种“股份”是有意义的。

网络和存储成本。尽管上述两种架构（即联盟链和股权证明区块链）解决了成员节点的原始计算成本，但它们对于智能手机来说仍然太昂贵。特别是，它们占用了大量的网络和存储资源，因为它们要求成员节点始终与区块链的“当前”状态保持最新。考虑到这种区块链实现的高交易速率（每秒 1000 笔交易），跨成员节点复制整个状态的成本很高：在 1000 笔交易/秒时，区块链每天将提交大约 9GB 的数据，这需要在成员节点之间传播八卦传播，导致每个成员节点必须承担大约 45GB/天的网络成本（假设一次八卦传播会辐射给五个邻居）。此外，这样的区块链将在成员节点上消耗 TB 的存储，因为每个成员节点都存储区块链的本地副本。

即使是针对智能手机的区块链也采用了相同的理念，即成员节点不断更新，从而导致网络和存储开销。一些区块链通过分片处理存储成本。OmniLedger 是最新的一个区块链，它允许参与者只存储区块链的一个碎片。它使用 Byzcoin（比兹币）的一

个变体来快速达成共识。RapidChain 还使用分片来降低存储成本。这两项工作的规模仅限于几千名参与者，而且还要求参与者存储整个区块链的大部分（1/3 或 1/16）。

轻量级但无法使用的节点。一类“轻量级”区块链采用了“不平等成员”的方法：只有第一层资源密集的成员参与共识并拥有投票权，而第二层成员只是作为只读查询前端，不参与共识。在这样的模型中，“多数诚实”属性必须完全由重节点来满足，因为轻节点对安全性没有贡献。毫不奇怪，由于责任有限，“轻”节点不会消耗太多资源。这种架构的一个例子是以太坊中轻节点和重节点之间的分离。

Blockene。相比之下，Blockene 为参与共识和区块验证的一流成员实现了轻量级的资源使用。此外，与以太坊不同，以太坊依赖于重节点中的诚实多数（只有重节点才能投票），Blockene 容忍高达 80% 的“重”节点（即政客）腐败。Blockene 的成员只需要一部智能手机和可忽略不计的数据传输（<60 MB/天，即低三个数量级）和可忽略不计的计算（每天电池使用率 <3%）。它通过使成员节点能够以提交特定块所需的最小状态进行操作，并且每天只执行几次工作，即不努力始终保持最新状态来实现这一点。

3.2 参与程度

由于区块链的安全性基本上依赖于大多数参与成员的诚实，区块链需要防止大量参与者的串通。联盟区块链为特定的业务流程精心构建区块链，使成员在区块链的成功中拥有共同的激励。有时，用上述担保组建联合体是不可行/困难的；在慈善事业的例子中，如果少数成员控制着区块链，他们可能会串通起来，比如，为穷人的捐赠提供便利。此外，联盟区块链与一组实体之间的特定业务流程紧密相连，除了限制互操作性之外，还导致了高设置和运营开销。

防止多数人勾结的另一种方法是实现大规模参与；通过招募大量参与者（比如数百万人），多数人的勾结可能变得困难且不太可能。比特币、以太坊和 Algorand 等大多数“公共”区块链都能实现大规模参与。Blockene 也支持大量参与者，但与当今大多数以加密货币为目标的公共区块链不同，Blockene 不与加密货币绑定（例如，没有股权证明），而是支持通用业务交易。与联盟区块链不同，Blockene 还可以在少数成员之间存在潜在勾结的现实世界场景中启用。

3.3 交易吞吐量

基于工作证明的公共区块链吞吐量低（约 4-10 个事务/秒）。基于股权证明的 Algorand[21] 是第一个拥有约 1000 笔交易/立方秒的公共区块链 3 联盟区块链，由于参与者规模较小，且传统共识（如 PBFT），每秒可提供 1000 笔交易。

与 Algorand 类似，Blockene 也提供高交易吞吐量。通过不与加密货币应用程序绑定，Blockene 可以为类似于联盟区块链的传统商业应用程序提供服务。

3.4 对参与者的激励

由于资源成本高（计算、网络或存储），现有区块链需要对参与者进行激励（例如，在加密货币中挖掘硬币，或在联盟中提高业务效率）。依赖此类激励的区块链无法用于慈善等应用（§1）。为了在没有激励的情况下扩大规模并实现利他参与，参与的成本必须可以忽略不计。

表 3-1 区块链架构比较

区块链	成员规模	交易速率	开销	是否需要激励?
公共 (例，比特币)	**数百万**	4-10 /sec.	巨大 (工作量证明)	是
联盟链 (例，[13])	数十	**1000s /sec.**	高	是
Algorand[21]	**数百万**	**1000-2000/sec.**	高	是
Blockene	**数百万**	**1045/sec.**	**小**	**否**

表 1 从这些维度比较了区块链架构。Blockene 是第一个实现以上所有功能的区块链：规模、吞吐量和低成本。凭借低成本，Blockene 支持真实世界的用例，即使参与者没有直接的激励，但他们愿意以微不足道的电池和数据使用量无私的运行后台应用程序。

3.5 其他相关工作

Blockene 基于委员会的共识很大程度上受到了 Algorand 的启发，与 Blockene 一样，Algorand 也不允许出现分叉，并且对区块链始终保持一致的观点。Algorand 和 Blockene 在抵御两种目标攻击的能力上存在权衡（见 §4.2 第 1 段）。HoneyBadger[28] 是最新的专为 O（100）个参与者的联盟区块链设计的系统。IOTA[19，20] 是另一个分布式账本系统，但目前依赖于一个集中的协调员来达成共识。

在基于工作证明的区块链中，与 Blockene 最密切相关的工作是混合共识 [31]。与 Algorand（和 Blockene ）类似，混合共识定期选择一组参与者，并且不允许对手在“参与者选择间隔”期间破坏节点。然而，它有一个很长的选择间隔（大约 1 天），也有分叉的可能性。

第 4 章 体系结构概述

在本节中，我们首先介绍我们的两层体系结构，它实现了轻量级资源使用、大规模参与和高事务吞吐量这三个相互冲突的属性。然后我们讨论了 Blockene 的威胁模型。

4.1 双层架构

Blockene 采用了一种具有不对称信任的新型双层架构。该架构如图 1 所示。

Blockene 中有两种节点：公民和政治家。公民资源有限（即使用智能手机），人数众多（数百万），是系统中唯一拥有投票权（即参与共识）的实体。政客们很强大，运行着服务器（类似于 Algorand 这样的现有区块链），人数也少很多（只有几百人），但他们没有投票权。政客只执行公民做出的决定，而不能自己做出任何决定。

低资源使用率使大量公民能够在没有激励的情况下参与，而政治家人数很少，将由对特定用例感兴趣的大型实体（例如，在经过审计的慈善案例中，大型捐助者和基金会）管理。

由于公民参与了共识，至少三分之二的公民被要求诚实，而其他人可能恶意串通。这是合理的，因为 Blockene 允许数百万公民参与，使大规模腐败变得困难。然而，政客们的信任度要低得多。Blockene 只要求 20% 的政客诚实；剩下的 80% 的政客可能心怀恶意，相互勾结，还有三分之一的恶意公民。

4.1.1 把工作交给政客

从直觉上讲，考虑到双层架构，公民可以将存储和通信等昂贵的责任推卸给政治家。然而，由于 80% 的政客都是腐败的，一个公民所写的内容可能会被政客丢弃，或者，一个读操作可能会返回不正确的值。为了让政客们完成有用的工作，Blockene 使用了一种复制读写的新机制。公民对政客的读和写都是在随机安全的政治家样本中进行的。该样本的大小是固定的，以便在高概率下，样本中至少有一个政客是诚实的 (例如，样本大小为 25，这个概率是 $1-(0.8)^{25}=99.6\%$)。Blockene 对于少数公民 (占比 0.4%) 挑选所有不诚实的政客很有弹性。

4.1.2 职责分工

我们现在描述公民和政治家如何合作执行各种标准区块链任务：

存储：在传统区块链中，每个参与者都保存着整个区块链的副本，但 Blockene 的公民无法负担存储 TB 级别的数据。在 Blockene 中，只有政客存储账本和全局状态（即关键值数据库 §2）。公民根据需要从政客那里读取这些数据的子集。唯一的州公民存储（并定期更新）是一个有效的公民身份列表（§5.3）。

交易验证：由于公民是共识的实际参与者，他们验证交易，确保交易已被签署，并具有语义完整性（例如，没有重复支出）。为了执行验证，公民从政客那里读取交易，并从政客的全局状态中查找其中引用的密钥的最新值。然后，公民提出一个拥有有效交易的区块。

八卦传播：为了确保所有诚实的参与者都同意区块链的状态，参与者需要相互八卦传播。然而，正如 §3 所讨论的，公民之间的直接八卦传播是昂贵的。Blockene 通过让公民通过政客八卦来解决这个问题。当一个公民需要向其他公民广播信息时，它会向安全的政客样本发送信息。然后，政客们互相八卦传播数据；他们有能力这样做，因为他们有良好的网络连接。当其他公民需要时，例如当他们在委员会中时，他们会重复从政客那里阅读。对于通过政客传播的八卦传播，我们需要保证，一个诚实的政客传达的信息总是通过八卦传播传播给所有其他诚实的政客，当 80% 的政客是恶意的时，这是一个具有挑战性的属性；我们的自定义八卦传播协议在 §6.1 中进行了描述。因此，我们实现了与公民之间的直接八卦传播相同的语义，但公民的网络负载最小。

共识：公民通过政客的八卦传播参与共识。由于公民规模庞大，所有公民都无法参与共识。相反，我们用密码为每个街区随机选择一个公民委员会（约 2000 名成员）（§5.2）。

4.2 威胁模型

尽管我们的威胁模型与 Algorand 相似，但在 Algorand 和 Blockene 之间存在一个折衷方案，即对定向攻击的适应能力。一方面，Algorand 是基于股权证明的，这允许对手有无限时间瞄准股份更高的节点（这些节点将更频繁地出现在委员会中）；Blockene 避免了这种攻击，因为所有公民都有平等的投票权。另一方面，Algorand 保护委员会成员的秘密，直到他们履行职责，但 Blockene 在他们参与前几分钟（1-2 个块）暴露了他们的身份。为了节省电池，公民通常大约每隔 10 个区块就会轮询政治家的区块链当前状态（5.2），但当他们要加入委员会时，会在预期轮到他们之前不久

（例如，1 个区块）再次轮询，从而将他们的身份暴露给恶意的政治家。这有可能为有针对性的攻击提供一个窗口（例如，通过贿赂委员会：§4.2.1）。

4.2.1 公民的攻击向量

对公民的贿赂攻击：由于 Blockene 隐含地提前几分钟（如 2 分钟）暴露了委员会的公钥，理论上，对手可以通过贿赂足够数量的委员会成员来进行有针对性的攻击。然而，我们认为这并不是一个问题，原因如下。首先，由于运营商级别的 NAT[4] 和智能手机的推送通知架构，仅凭 IP 地址，对手向公民" 发送消息" 提供贿赂是不难的；从恶意政客到公民的现有通道不能被滥用于此，因为公民身上未被篡改的 Blockene 应用程序将忽略该通道上的任何虚假流量。其次，由于委员会在每个区块都是随机选择的，基于拉动的贿赂，即公民（提前 10 个区块知道他们的选择–第 5.2 节）主动联系对手的情况不可能发生，因为这意味着违反了公民的诚实假设，即大于 70% 的人是诚实的。

公民的 Sybil 攻击。鉴于参与成本是轻量级的，Blockene 需要确保对手不能通过旋转几个虚拟节点来获得不成比例的投票份额（即 Sybil 攻击 [17]）。解决 Sybil 攻击的常见方法是工作证明（Proof-of-work），它是资源密集型的，不符合 Blockene 的目标；另一种选择是股权证明（Proof-of-stake）[21]，参与者的投票权与区块链上的" 股权"（金钱）数额成正比，但它是针对加密货币的。

在 Blockene 中，我们通过利用智能手机中的可信硬件（TEE）来防止 Sybil 攻击 [6,11]，并确保一个智能手机在区块链上最多有一个身份。因此，Blockene 对参与施加了经济成本，即智能手机的成本；这是在拥有智能手机时已经产生的沉没成本，但由于每个身份都是唯一的智能手机，所以可以防止 Sybil 。

特别是，每个 TEE 都有一个独特的公钥，由平台（Android/iOS）供应商认证。TEE 可以认证一个由应用程序生成的 EdDSA 公钥-私钥对；这个生成的公钥作为 Blockene 上的身份。Blockene 的全局状态跟踪有效公钥的集合，以及授权它的 TEE 的公钥/证书。当有人提出增加新成员的交易时，Blockene 会查找 TEE 的公钥，看该 TEE（即同一个智能手机）是否已经在 Blockene 中拥有一个身份；如果是，它就拒绝该交易。因此，Blockene 上的每个公民都与一个独特的智能手机联系在一起，这使得单个实体在 Blockene 上获得大量参与在经济上是不可行的/没有吸引力的。

请注意，Blockene 只假设谷歌/苹果为 TEE 公钥签署的每份证书都对应于一个独

特的智能手机。它不依赖于单个 TEE 的安全性（不像在 TEE 内运行区块链共识，例如 SGX[33]，它开启了损害完整性和安全性的侧信道攻击）。因此，TEE 的身份可以被替换/与其他独特的身份结合起来。在印度，可以使用 Aadhaar-ID[1，12] 的单向哈希值（可数字验证，生物特征消除，12 亿范围）。也可以使用其他经过去重处理的 ID（如 SSN），并辅以数字验证。

4.2.2 政客的攻击向量

应对政治家中 80% 的不诚实行为是 Blockene 设计中的主要技术挑战之一。政治家的恶意行为分为两种：可察觉的和隐蔽的。可察觉的恶意行为，即有一个简洁的撒谎证据，可以通过列入黑名单来提高性能。例如，如果一个政治家在一轮中只发送一组交易，但有两个由同一个政治家签署的版本，这是有证据可查的。隐蔽的恶意行为更难处理，也是我们技术的重点。我们列出了政治家可以采用的广泛的（非详尽的）隐蔽攻击类别。

陈旧性攻击：当一个公民节点向政客询问一些状态（例如，最新承诺的区块）时，政客可以返回一个陈旧的区块。这样的响应似乎是有效的，因为旧的区块也会被法定的公民签署（§5.3）。

分离视图攻击：政客可以有选择地回应一些公民，而不回应其他公民，导致诚实的公民看到的世界观分裂。更糟糕的是，一个政客可以对不同的公民子集做出两种不同的回应。在一个协调的分裂视图攻击中，恶意的政治家只能在他们自己之间说闲话，所以不诚实的政治家拥有某种特定数据。然后，恶意政客可以有选择地将这些数据转发给一些公民（例如，§5.5.2）。

丢弃攻击：恶意政客可以丢弃公民写的数据，而不将其提交给其他政客，也不将其传给其他政客。同样，在阅读时，政客可以选择不回应，即使政客有数据（§4.1.1）。

拒绝服务攻击：由于政客是通常托管在云中的强大服务器，我们假设诚实的政客采用公共云提供的标准 DoS 保护 [2，3]。对于公民来说，大多数 ISP 采用运营商级 NAT 来处理移动电话上的 IP 地址爆炸 [4]，这也提供了 DoS 保护。恶意的政客可以通过要求比他们需要的更多的数据来使我们的八卦传播协议更加昂贵（6.1 节）。

Sybil 攻击：对手可以通过启动几个节点，试图将政客的不诚实比例推到 80% 以上。然而，在很少数量（比如 200 个）的情况下，我们设想政治家节点将有一个强大的带外注册机制（例如，将他们映射到真正的实体，比如每个财富 500 强公司对应

一个实体），因为他们中只有 20% 需要诚实（不像公民那样）。

Blockene 可以防止包括上述攻击在内的政客的可察觉和隐蔽的恶意行为。

第 5 章 设计

在这个章节，我们展示了更多关于政客和公民如何在 Blockene 的关键步骤上进行协调的细节。

5.1 系统配置

我们首先概述了 Blockene 的系统配置。Blockene 中的公民使用智能手机运行，所以我们假设他们的网络带宽很低，即 1MB/s。我们选择一个 9MB 的区块大小（以摊销每个区块的固定成本），包含大约 90k 个交易（每个约 100 个字节，包括一个 64 字节的签名）。我们假设政客之间的网络带宽为 40MB/s（代表云中的带宽，例如，美国东部和西部的 Azure 和谷歌云虚拟机之间的带宽）。我们选择政客的数量为 200。每个区块所做的工作只取决于委员会的规模，因此该系统可扩展到数百万个公民。

交易发起人提交已签署的交易给安全样本或所有政客，且在后台持续进行。交易可以修改发起人可以访问的密钥。来自同一发起人的交易可以相互依赖；我们通过跟踪全局状态中每个发起人的随机数来保持它们的顺序。在本文中，（在不丧失一般性的情况下）每笔交易访问三个密钥（借入一个密钥，贷出另一个，第三个密钥是随机数）。政客们在彼此之间进行八卦传播交易。

5.2 公民选举的委员会

验证和签署每个区块的委员会是以 VRF（一种可验证随机函数）为基础被选择的，它的灵感来源于 Algorand[21]，但有一个关键修改。Algorand 需要每个参与者检查每个轮次中自己是否被选择进入委员会。一个使用移动手机的公民不能承担起如此频繁的检查，因为每轮唤醒手机都会进行交流并导致电池严重耗尽。所以，Blockene 使用区块 N-10 的哈希，而不是根据前一个区块 N-1 的哈希计算 VRF，因此允许公民在每 10 个区块醒来一次。请注意，此修改仍然保留了我们的威胁模型中 VRFs 所需的安全保证。具体来说，对于一个公民来说，区块 N 的 VRF 被计算为 $Hash(Sign_{sk}(Hash(Block_{N-10})||N))$，其中 sk 是公民已知的私钥。如果 VRF 的最后 k 为有 0，则公民就在委员会中（因此公民是概率为 2^{-k} 的委员会的一部分，k 可以适当设置）。只有相关的公民可以生成 VRF，因为它需要私钥，但任何人都可以根据给定签名的公钥验证其有效性。

委员会规模：委员会的规模需要平衡性能和安全性。一个小委员会对性能来说是好的，但为了确保共识协议的安全，我们要求在任何委员会中，至少有 2/3 的公民是诚实的。由于我们的委员会选择是概率性的，根据 Chernoff 约束，即使我们有 2/3 的诚实公民，对于非常小的委员会规模也无法满足这一安全要求。委员会的规模随着不诚实公民的比例增加。我们对这种权衡进行校准，得到预期的委员会规模为 2000，公民不诚实的阈值为 25%。这些计算的细节出现在完整版本中 [34]。我们将在下面给出概述。

证明概述：我们证明关于一个区块的委员会的几个属性。如果 iyge 参与委员会的公民是诚实的，并且通过 m 个扇出读/写与至少一个诚实的政客交谈，我们称该公民为好的。否则，我们说该公民为坏的。对于一个有 25% 腐败的公民，80% 腐败的政客，m=25 时，我们表明我们的委员会满足以下属性 [7]：所有委员会的规模都在 [1700...2300] 的范围内（引理 1），每个委员会至少有 1137 个好公民（引理 2），每个委员会至少有 2/3 的好公民（引理 3），没有一个委员会有超过 772 名坏公民（引理 4）.

5.3 防叉式结构验证

Blockene 的设计是为了防止分叉发生。为了实现这一点，每个公民定期验证区块链的结构完整性，以确保哈希链和 VRFs 的一致性，并防止分叉。

跟踪本地状态：每个公民在本地记住区块编号 N，直到公民验证区块链的结构完整性，以及区块 N 到 N-9 的哈希值。此外，一个公民存储一个最新的其他有效公民的公钥列表。对于 100 万个公民来说，总存储大小是小于 100MB 的。

链接 ID 子块：为了使公民能有效地更新本地状态，作为每个区块 B 的一部分添加的新用户的公钥，被在 B 内的 ID 子块（SB）中被跟踪。SB 通过在 SB_i 中嵌入 $Hash(SB_{i-1})$ 而被连锁。为了帮助廉价验证，委员会成员签署了 $Hash(Hash(B_i)), Hash(SB_i), GlobalStateRoot(B_i)$。

增量验证：大约每 10 个块 (12-15 分钟)，每个公民执行一次 getLedger 调用，以验证增量结构的完整性（即从最后一个验证点到最新状态），并检查他是否会很快进入委员会（区块 N 的委员会是一个关于区块 N-10 的哈希函数）。为了找到最新的区块，公民查询政治家的安全样本以获得最新的区块代码。它挑选任何政客报告的最高数字，并且要求证明，即该区块的委员会和相应的 VRFs 的签名。因此，如果安全

样本中至少有一个政客是诚实的，公民将知道最新的区块哈希值。如果最新的区块大于 N+10，则首先验证区块 N+10。此外，它通过下载链式子块 $SB_{N+1}...SB_{N+10}$ 来刷新它的有效公钥集，这些子块包含每个区块中添加的新公民，根据链式哈希值验证 SB_i 的完整性。

新节点的冷静期：为了防止（低概率）攻击，即对手可以制造公钥对 [8]，以增加在特定区块 N 中获得更高的恶意分数的机会。我们允许公民在添加公民的区块后仅 k（=40）个区块进入委员会。为了验证这一点，作为 VRF 检查的一部分，公民的本地状态会跟踪“最近”添加的公民的块数。这类似于 Algorand[21] 中的“回看参数”。

证明概述：我们的 getLedger 协议 [34] 被用于验证账本高度 i+10。假设公民 v 最后验证了高度 i，而不需要明确地对所有 10 个区块的签名进行暴力验证。该算法可推广到验证任何高度 i+j。我们表明（引理 5），如果一个拥有高度为 i 的已验证状态良好的公民在第 i+11 轮调用 getLedger 协议并接受，那么该公民的更新结构状态与高度为 i+10 的区块链一致。使用这一点，我们可以证明诚实的公民可以获得取款连的异质结构状态，以及所有已注册的公钥，用于每一轮协议（推论 2）.

5.4 交易验证

公民执行验证交易签名的任务，检查交易随机数以检测重放攻击，并验证交易的语义正确性（比如双重支出）。但是，只有政客存储这个全局状态的 Merkle 树（§2.2）；在公民中保持一个巨大的且随时更新的全局状态是无法负担的。要验证交易，一个公民必须寻找其中引用的键的正确值。在提交时，公民必须用交易的新值更新 Merkle 树，并签署新的 Merkle 根。挑战是如何在不被信任的政客面前如何正确行事。

Merkle 根（伴随区块号码）是被先前的区块的委员会签署的，因此政客不能欺骗关于 Merkle 根的内容。一旦公民知道了最新的区块号码（§5.3），它也将了解到正确的 Merkle 根。为了验证密钥的返回值，公民要求政客发送该密钥的质询路径，即从叶到根的路径上的所有兄弟节点（哈希）。这是公民能重建 Merkle 路径，并且用签名的 Merkle 根和根的哈希值匹配。通过哈希值的安全性，政客不能提出虚假的挑战路径来验证。在一个有 10 亿个键值对的树中，挑战路径将包含 30 个哈希值。

Merkle 树中的密钥更新遵循类似的协议。公民可以用叶子上的心值构建一个局部 Merkle 树，并计算新的 Merkle 根。上面提到的读取和更新路径的过程都是很昂贵的，我们在 §6 中对它做了优化。

5.5 区块提案

像任何区块链一样，委员会成员可以设计一个新的区块来向区块链进行提交。

5.5.1 选择胜出提案者

为了效率，我们只允许区块委员会中的一个自己根据公民的 VRF 真正提出一个区块，它们被称为提案者。对于这个选择，我们使用一个附加的 VRF，它基于前一个块 N-1（而非 N-10）的散列；只有附加 VRF 的最后 k' 位为 0 的委员会成员才可以提出区块，而获胜者是拥有最小 VRF 的那个。在这个 VRF 中使用上一个区块的哈希可以确保对手直到最后一刻才知道提案者（与 Algorand 类似），从而防止提案者受到针对性攻击。任何委员会成员都可以在提案者中确定胜出的 VRF。所有提案者将它们的区块上传给政客，然后其他委员会下载胜出的提案者的区块。

5.5.2 预先声明提交

提议者需要将提案的区块上传到 25 位政治家的安全样本中。在 Blockene 中，区块大小约为 9MB，假设移动节点的带宽为 1Mb/s，这将需要 225 秒。为了优化此步骤，我们使交易选择过程具有确定性，这样任何公民都能重现原提案者所作的事情，而无需提案者显式上传整个区块。但实现确定性是有挑战性的，因为 80% 的恶意政客可以向不同的公民发送不同的交易。我们对交易进行预先声明提交的技术解决此问题。

1. **冻结交易**：在区块 N 的开始阶段，每个政客对它将要发送给待读取公民的确切交易集进行冻结。它通过创建一个 tx_pool 来实现这一点，其中包括一组（约 2000）的交易，然后生成一个提交记录，它是 tx_pool 签名的哈希散列和区块号[9]。恶意政客被强迫给给定的 N 号区块只发布一个提交记录，因为来自同一个政客的两个签名的提交记录是恶意行为的证明，并且能用于有效的黑名单；公民接下来在同一轮投票中放弃来自该政客的所有提交记录。直观来讲，在冻结记录的情况下，一个公民提交区块是，不需要上传整个区块，而只需要上传一个包含该区块提交记录的摘要，然后其他公民可以通过下载来自政客的提交记录的 tx_pool 来重现这些区块。

2. **确保足够多的诚实公民拥有提交**：一个恶意的政客可以只用它的 tx_pool 回应一部分公民，并拒绝回应其他人；因此，在提议的区块中提交的 tx_pool 有可能

不会被所有诚实公民读取，从而阻碍达成共识。为了解决这个问题，我们执行以下三个步骤：一，我们限制一个政客的确切集为，根据区块号和前一个区块哈希值随机选出的 45 名政客集合，从这些政客处为给定区块拉去交易。不是从随机的安全样本中读取 tx_pools，而是公民从这 45 个被指定的政客处为一个区块读取 tx_pools。二，公民上传一个证人列表给政客中的安全样本，这个证人列表应包含公民能够成功下载的 tx_pool 列表。所有公民的证人列表会在政客间被广泛传播。三，提案者阅读其他所有公民的证人列表，并且选出那些 tx_pools 被至少阈值数量的公民成功下载的提交记录。这个阈值被固定为 $\widetilde{n}_b + \Delta$，其中 $\widetilde{n}_b$ 是任何委员会中恶意节点的最大个数（由引理 4 计算得到，为 772），并且 Δ 被选定为 350。直观地说，所有通过这个条件的提交（和 tx_pools）都可以得到至少 Δ 诚实的公民。由于 20% 的政客是诚实的，按照估计，45 个提交中至少有 9 个能通过这个测试。

3. **确保所有诚实公民可获取提交**：至少 Δ 的诚实公民的提交现在需要被传播给所有诚实公民。第 4 步中的每个公民重新上传 5 个随机的 tx_pools（包括来自于恶意政客的）给 1 个随机的政客。这（很大概率）确保了每个至少属于 Δ 诚实公民到达至少一个诚实政客（然后八卦传播给其他诚实的政客）。因此，其他诚实的公民可以成功地下载 tx_pool（通过质询政客的安全样本），从而防止恶意政客带来的分离-视图攻击。

4. **处理恶意提案者**：当区块提案的获胜者是恶意公民，它不需要证人名单准测，并且能选择一个很少有公民知道的 tx_pool 提交。只有当共识输出这个恶意提案者提议的区块时，这种攻击才可能发生，因此我们可以认为至少有 1/3 的诚实公民在共识开始时拥有所有的 tx_pools。为了确保所有诚实的公民都有能力下载所有需要的 tx_pools，我们将再次上传已随机选定的 tx_pools（第 9 步），现在包括从前面步骤下载的 tx_pools。在引理 10 和 11 中给出了捕获证明系统安全性所提供的保证的形式证明。在引理 10 和 11 中给出了捕获证明系统安全性所提供的保证的形式证明。

5.6 区块提交协议

区块链中主要的操作是给区块链中添加一个新的块。我们列举了以下的在提交区块 N 的过程中的关键步骤。一旦前一个区块（区块 N-1）从块 N−1 的委员会成员那里收集了阈值的签名（在我们的例子中设置为 850，§E.1[34]），区块 N 的协议就开始了。

1. 区块 N 选择一个新的公民委员会（使用区块 N−10 的哈希），用 C^N 表示。C^N 中的公民继续对最新提交的块号进行轮询，并在该数字为 N−1 时启动协议。

2. C^N 中的每个公民 C_i^N 从 ρ=45 名指定政治家处为当前区块下载 tx_pools 和提交记录。

3. 每个 C_i^N 上传一个带签名的、带有它下载记录的证人列表给政客的安全样本。

4. 每个公民 C_i^N 选择 5 个随机的 tx_pool，并把它们重新上传给 1 个随机政客。

5. C^N 中的每个提案者从政客的安全样本里下载 C^N 的所有证人列表，并且选择至少有 1122 票的提交（§5.5.2）。然后，它用这些提交记录生成一个区块提案，同时包括其 VRF，来证明提案者的资格。

6. 政客对区块提案/VRFs 和公民重新上传的 tx_pool 进行八卦传播。

7. 依赖其他公民在第 4 步中重传的内容，每个公民 C_i^N 尝试从政客的安全样本里去下载在第二步中缺失的 tx_pools。

8. 每个公民 C_i^N 从一个政客的安全样本中阅读 C^N 中所有提案者的 VRFs，并选取最小的正确 VRF 作为本地胜出者。如果 C_i^N 拥有胜出提案中的所有 tx_pools，它将携带该提交集加入共识，否则携带空集。

9. 每个公民 C^N 执行第二次重新上传的 10 个随机 tx_pools 给 1 个随机的政客。

10. C^N 中的公民通过政客之间的八卦传播运行共识协议（§5.6.1），每个 C_i^N 的投票在第 8 步中被决定。在最后，所有诚实的公民要么同意同一套提交集，要么同意一个空区块。根据政客的安全样本的共识的输出，C_i^N 下载了缺失 w.r.t 的 tx_pools。

11. 每个公民 C^N 通过从政客（§5.4）下载所有密钥的挑战路径来执行事务验证，并删除验证失败的事务。

12. 基于有效交易记录（第 11 步），每个 C_i^N 创建一个区块，使用更新的键值计算全局状态的新 Merkle 根，并对块散列、新 Merkle 根以及块号 N 进行签名。它将块散列、新 Merkle 根和这个签名上传给安全的政治家样本。

13. 当更多的签名的阈值数字被积累进区块 N 后，区块 N+1 开始。

我们完整的协议描述可以在算法 4 中找到。我们概述了 Blockene 的各种特性，即安全性、活性和公平性（在 §7 中）。

5.6.1 共识协议

对于共识（第 10 步），我们使用拜占庭协议（BA）算法实现字符串共识（这基于 [36]），它以黑盒的方式调用比特共识算法 BBA。这和 Algorand 使用的共识算法相同。公民带着本地胜出区块中的提交列表进入共识，作为输入。这里有两种相关场景。如果获胜提案者（即 VRF 更小的那个）是诚实的，至少在 2/3 的情况下如此，所有委员会中的诚实公民都会带着这一提案进入共识，只有小概率例外（引理 10）并且协议将在 5 轮后终结。但是，如果获胜提案者是恶意的，它可以与恶意政客合谋，分离诚实公民的观点。总的来说，这个共识协议将需要 11 轮。

第 6 章 优化

在本节，我们提出了两个关键的优化，这对于在 Blockene 中实现高事务吞吐量至关重要。

6.1 优先排序的八卦传播

问题。我们在 Blockene 内所要求的保证是，如果一个诚实的政客有一个信息，所有诚实的政客都会收到这个信息。因为政客中不诚实的比例很高，与少数邻居（例如，10 个）的标准多跳八卦传播不能提供这种保证，因为有一个非常重要的可能性，他们都是不诚实的，并放弃了信息。因此，安全的做法是向所有的其他政客进行全面的广播，这是昂贵的；当政客需要八卦传播由公民在委员会中重新上传的 tx_pools 时，每个政客可能有 45 个 tx_pools 来八卦传播；通过完全的广播，它将发送 0.2MB*45*200=1.8GB，在关键路径上需要 45 秒（@40MB/s）。

关键想法。我们利用了这样一个事实，即被不同政客八卦的信息有很高的重叠；每个政客都有相同的 45 个 tx_pools 中的一个子集，因为公民会随机选择一个政客来重新上传 tx_pools 的一个子集。此外，考虑到重新上传的性质，预料中，任何政客都只缺少几个 tx_pools，诚实的政客不会在状态上撒谎。

1. **握手**。每个政客向接收者 B_i 询问哪个 tx_pools 是它们已经拥有的，并且只发送丢失的那个。虽然这对诚实的政客很有效，但 80% 的恶意政客总是会说那些没有恶意的谎言，从而对系统造成更高的负载/延迟。

2. **自私的八卦传播**。由于恶意政客可能会撒谎说他们没有 tx_pools，我们给那些遗漏大量 tx_pools 的政客给予软惩罚。每个发送方政治家 A 支持具有 A 需要的最大 tx_pools 数量的同伴 B。在每一轮中，A 发送一个 tx_pool 给 B，然后也收到一个 tx_pool 作为回报。考虑到公民的随机重传，每个诚实的政治家一定只会缺少一小部分 tx_pools，并且因此会得到更高优先级。当 B 从其他节点获取 tx_pools 时，B 所能提供的内容列表会不断更新；要注意的是这个列表只会增长，不会缩小。

3. **激励节俭的节点**。一旦发送者收到了所有的 tx_pools，自私的八卦传播就失去了区分诚实和恶意的接收者的能力。为了解决这个问题，在获得所有的 tx_pools

后，发送方将其对目的地 B_i 的优先级功能更改为 B_i 声称拥有的 tx_pools 的数量；因此，具有大部分 tx_pools 的诚实节点就会受到喜爱。同样，B 宣传的 tx_pools 列表只能增长，而非收缩，因为萎缩意味着 B 撒谎。此外，每个诚实的 B_i 同时从最多 k=5 个对等点请求其缺失的块；k=1 将节省数据，但如果对等点不诚实地延迟响应，则会产生很高的延迟。

6.2 基于采样的 Merkle 树的读/取

问题。步骤 11 中的 Merkle 树验证非常昂贵。在一个 10 亿节点的 merkle 树（30 层深）中，挑战路径是 300 字节（10 字节散列）；忽略压缩的下载 270K 的挑战路径是 81MB（约 81 秒延迟）。公民网站的计算量也很高（在读取过程中进行挑战路径的验证，在更新后计算新的根，共需要 1620 万次哈希计算）。

关键想法。我们以一种可证实的方式将大部分工作转移给政客们。由于 Merkle 树验证是在通过政客的八卦运行共识结束后完成的，政客们知道考虑构建区块的 tx_pools。因此，委员会的所有公民和政客都知道哪些键的值需要进行读取和更新。我们首先讨论了从 Merkle 树中正确读取值的优化问题。

1. **获取值**。每个公民只从一个政客那里得到所有的 270K 的键（没有挑战路径，是 1MB 而不是 81MB），然后询问一个安全的政客样本，这些值是否正确。由于这些政客中至少有一个是诚实的，它通过异常列表来提醒公民主义不正确的值。政客可以通过提供一个由已签名的 Merkle 根开始、并指示了这个键对应了不同值的挑战路径来“证明”一个值是不正确的。

2. **抽查**。如果很多值都是错误的，异常列表将遍非常巨大并且消耗存储。为了避免这样的情况，公民选取包含 k’=4500 的一个小的随机子集来用于初始时用挑战路径进行抽查。如果抽查对一个足够大的 k’ 是通过的，那么一个政治家只能针对一小部分（200 个）键撒谎（例外只会在很小的概率下发生）。因此，额外的抽查限制了异常列表的大小（引理 6）。

3. **意外列表协议**。为了用一个安全的样本来交叉验证这些值，公民会确定性地将这些值放进桶里（2000），上传这些桶的哈希值。当政客注意到一个桶是不能匹配的时，政客将发送桶索引和该桶中所有键的正确值。公民只有在与第一个

政治家的键值不一致的情况下才获取挑战路径。我们的抽查确保只有少数桶会不匹配。

边界情况。尽管做了以上这些事情，现在仍有小概率（$< 2^{-10}$）使公民获得不正确的值；我们认为这样的公民节点是恶意的，并对其进行适当处理（引理 7）。在算法 2 中给出了完整的协议和所有的证明。

写。更新 Merkle 树是一个更棘手的问题。由于所有被更新的键都缺乏旧的挑战路径，所以公民无法构造更新后的 Merkle 树 T' 的根。我们通过让政客计算 T' 的方法来解决这个问题，但现在公民必须验证政客给出的计算是正确的，即 T' 与更新的键的新值一致，未修改的键与旧树 T 的新值一致。我们实现这一点的方式是，在某个级别上破坏 T'，这一级别称为边界级别（该级别上的节点是边界节点）。公民从政治家的安全样本中获得 T' 的边界节点值。然后公民们运行一个抽查算法 - 他们随机选取一个边界节点子集，并要求一个政客证明该边界节点的正确性。接下来，公民在其他被选中政客的帮助下创建异常列表。这个列表表示哪些边界节点与公民不一致。接下来，公民依次纠正错误的边界节点，然后最终从边界节点计算出 T' 的正确根。

证明概述。在全文 [34] 中，我们证明了（引理 6）对于一个好公民来说，在成功抽查了关键值的 μ 分数后，只有（小部分）τ 值是不正确的，概率为 $1-\varepsilon_1$（这里，μ、τ 和 ε_1 是适当选择的参数）。而且，这些值将通过处理最多为 τ 的异常列表来得到修正。因此，一个好的公民得到正确值的概率为 $1-\varepsilon_1$（推论 3）。我们选取的参数（引理 7），使最多 18 个好公民会在读取过程中获得错误值，并在委员会中将它们视为坏公民，算作对它们的处理。在写协议中，我们可以证明异常列表的大小是有界的（引理 8），并且不超过 18 个公民会接受一个错误更新的 Merkle 树 T'（引理 9），此时我们再次将其纳入坏公民集合里。我们还表明了我们的算法在全局读/写方面比朴素算法的通信效率提高 3 ∼ 18 倍，计算速度提高 10 ∼ 66 倍。

第 7 章 安全性、活跃性和公平性的证明

在本节中，我们将简要概述全文 [34] 中关于 Blockene 的安全性、活跃性和公平性保证的证明。

在一个新区块得到签署并且被委员会成员中阈值数量（T’）的人提交后，一个委员会轮次 N 终结。T’ 将被设置为 850（考虑最大数量的坏公民在任何委员会，同时 36 个好公民可能读/写一个不正确的全球状态）。

首先，我们证明（引理 10 中），对于一个区块，如果一个好公民是获胜的提议者，那么（除了有界常数概率）所有好公民都将输出这个公民的提案作为共识协议的输出。在引理 11 中，我们证明了，相反地，如果一个恶意的公民是获胜的提议者，共识的结果是非空值，那么所有好公民都能下载提案中提交的交易。使用引理 7 和引理 9（见 §6 中的证明概述），我们接下来证明（引理 12）了在所有区块提交协议的最后，除 36 人之外，所有好的公民都将签署相同的块哈希和新的全局状态根，并且新的块与整个区块链和全局状态一致。现在，使用引理 12，安全性 (即所有诚实的公民同意所有提交的块且所有区块都与一个正确的交易序列一致) 通过一个归纳论证得到验证。接下来，为了论证活跃性 (敌对实体不能无限期地拖延系统，并且空块概率以一个小常数为界)，我们使用引理 12 和 10。

另外，我们还证明了引理 13 中的吞吐量边界（在期望中，已提交的块中有一个阈值的事务数）和引理 14 中的公平性（所有有效的事务最终都将被提交）。

第 8 章 实现

我们已经建立了 Blockene 的原型，它横跨两个组成部分: 公民节点和政客节点。

8.1 公民节点

公民节点是 SDK v23 上的一个 Android 应用程序，有 10200 行代码。它的设计是为了优化电池的使用，并作为一个后台应用程序运行，在初始设置后没有用户的参与。这个应用程序满足了公民参与的协议的两个主要阶段：被动阶段和主动阶段。在被动阶段，它使用作业调度程序 [9] 的服务定期轮询政客的 getLedger 调用。在主动阶段，当公民是委员会的一部分时，应用程序运行协议的步骤，处理失败、超时和重试来处理腐败的政客。主动阶段的实现使用了多线程事件驱动模型，建立在 EventBus 之上，以并行化和管道网络，计算以签名验证为例的密集的加密任务。

8.2 政客节点

政客节点使用 C++ 实现（11000 行代码）。它可以实现扩展到成千上万的公民加载，并处理八卦传播期间的突发加载。考虑到协议的状态机性质，我们将它建立在方便的 C-Actor-Framework[16] 框架上，该框架是基于通过协议步骤转换政客状态的“演员”。例如，BBA 演员，除了存储和服务公民提交的投票，还阅读投票以决定共识的结果。基于此，它会发出一个事件来构建更新后的 Merkle 树。

对于全局的状态来说，我们已经建立了一个 SparseMerkleTree(SMT)，它的叶索引是使用键的 SHA256 进行确定性计算的。因为这个树的深度是有界的，所以我们允许它的叶子节点有一些（小部分的）碰撞。任何键的挑战路径都包括所有的与该键共同定位的所有碰撞，因此可以计算叶哈希。为了防止单个叶节点的定向溢出，我们拒绝使一个叶节点超过阈值的键添加，从而迫使事务发起者使用不同的键。我们还实现了 DelteMerkleTree，它允许我们使用仅与有关的键成比例的内存高效创造和更新 SMT 的版本

我们的八卦传播实现了对常规消息进行简单的广播，并为 tx_pool 八卦运行一个有状态协议。我们将这些消息隔离到不同的端口/队列中，这样突发的八卦传播信息就会与广播的小消息（BBA 投票）隔离。为防止恶意公民让诚实政客担负起在他们的文章中散布流言蜚语的责任，我们限制了政客的集合，让公民基于 VRF 而具有确

定性。政客们不会从不符合标准的公民那里散播八卦的消息。

第 9 章 评估

我们从以下几个维度评估我们的 Blockene 原型。我们评估过程中回答的主要问题是:

- Blockene 提供了怎样的吞吐量和延迟?
- Blockene 处理恶意行为的效果如何?
- 对 Merkle 树和八卦传播的优化是有效的吗?
- 公民节点的负载（电池、数据使用）怎么样?

9.1 实验装置

在我们的实验里，我们使用了一个包含 2000 个公民节点和 200 个政客节点的设置。公民节点是设置在 Azure 上的单核虚拟机，具有 Xeon E5-2673, 2GB RAM，分布在 WAN 上的三个地理区域:SouthCentralUS 700 vm, WestUS 600 vm, Eas- tUS 700 vm。每个公民都运行安卓 7.1 映像，其速率限制为 1MB/s 网络上传和下载。政客节点运行在 8 核的 Azure vm 上，具有 Xeon E5-2673, 32 GB RAM，在 EastUS 和 WestUS 各分布为 100 个虚拟机。它们的速率被限制为 40MB/s 的网络带宽。在安全随机抽样的情况下，公民-政治家的交流跨越 WAN 地区。同样，政客之间的八卦传播也发生在各地。由于我们委员会的人数是 2000 人，每个市民都在每个街区的委员会任职。如果公民人数增加，比如 100 万，那么每 500 个街区就会有一个特定的公民进入委员会。除了每个公民的负载外，系统性能与公民总数无关，只是关于委员会规模的函数，所以这些数字代表了一个大的设置。

9.2 交易吞吐量和延迟

图 2 展示了在完全诚实和存在恶意的配置下 Blockene 提交 50 个连续区块的时间线。在完全诚实（0/0）的情况下，460 万个事务在 4403 秒内被提交，对应的吞吐量为每秒 1045 个事务，即 114KB/s。

我们同样在公民和政客的恶意行为下评估 Blockene。我们用 P/C 格式表示我们的恶意配置，其中 P 是恶意政客的比例，C 是恶意公民的比例。通过我们对参数的

选择（比如委员会规模），Blockene 保证在高达 80% 的恶意政客和 25% 的恶意公民存在时确保安全。但是，敌对行为会影响它的性能表现。在这些实验中，一个恶意公民通过两种方式攻击：（a）通过和恶意政客密谋来强制一个空块，并且提议一个只有恶意政客才拥有的 tx_pools 块。诚实的公民因此无法下载这个提交，并且将为空块投票;（b）通过操纵投票来强迫 BBA 共识协议进行额外的轮次。一个恶意的政客通过两种方式攻击：（a）不给出交易承诺，使 45 个 tx_pools 的一个子集为空，可能导致提交一个更小的块；（b）操纵八卦传播，作为槽孔来向多个对等体请求相同的块。如图 2 所示，Blockene 对各种恶意行为都是非常健壮的，而且可以优雅地降低性能。在 80% 的政客不诚实的时候，有效的 tx_pools 从 45 个减少到 9 个，导致区块只有 18K 的交易，而非 90K。当他们被选为提案者时，恶意公民会造成性能打击（空块 +BBA 轮次）；表 2 展示了更多恶意行为配置下的吞吐量。

图 3 显示了不同配置下系统事务延迟的 CDF，演示了跨事务的公平性。在完全诚实的情况下（0/0），Blockene 确保了 135s 的中卫延迟和 263s 的 99%-ile 延迟。在 50：10 和 80：20 这两个恶意配置下，延迟比预期的高。

9.3 公民和政客的时间线

图 4 展示了 10 个区块（每个重复模式都是一个块）间一个典型的政治节点的网络负载。在上传数据时的两个大峰对应于这个政客是 45 个被选来提供 tx_pools 之一的轮次。对每个块来说，有两个传输数据的小峰；第一个峰对应通过优先八卦传播的 tx_pools 的八卦传播，第二个峰是因为 BBA 共识中公民的八卦传播。

我们同样展示了 89 秒块延迟的分解，通过绘制在一个典型块中公民节点花费的时间。图 5 展示了其中一个区块中 2000 个公民节点的进度，分离出了协议的关键阶段；大部分的时间都用于事务验证阶段，及从政客手中获取 tx_pools。

9.4 优化的影响

我们现在评估了优先八卦传播和基于抽样的 Merkle 树优化。对于八卦传播，我们考虑在所有其他诚实政客拿到所有 tx_pools 之前每个政客需要多少上传/下载。例如在 0/0 的情况下，我们拥有 10K 的数据点（跨越 50 个区块和 200 个政客）。在这些样本中，我们绘制了第 50、90 和 99 个百分位数。我们在 80/25 的情况下建模的恶意策略是只有最低数量的城市公民拥有恶意政客的 tx_pools（Δ，§5.5.2），所有恶

意政客从所有诚实节点请求 tx_pools 的完整集合。如表 3 所示，优先八卦传播的网路负载对不诚实行为具有鲁棒性。尽管在恶意设置中，在所有诚实的政客获得所有 tx_pools 之前，数据传送也是非常少的。

表 4 对比了我们基于 Merkle 树优化的读取和更新的抽样的性能，以及为块中引用的所有键下载挑战路径的简单解决方案。简单的解决方案会导致更高的网络成本（这些数字在 gRPC 压缩后产生），以及公民的巨大网络成本。采用我们的优化措施后，网络成本降低了 10.8x，CPU 成本降低了将近 31x，从而显著提高了事务吞吐量。

9.5 公民的负载

最终，我们评估了公民节点由于运行 Blockene 的负载。我们感兴趣的两个指标是电池使用量和数据使用量。为了获取这些指标，我们运行了一款带有公民应用实际的手机（一加 5），作为委员会的一部分，与虚拟机中的 2000 个委员会成员一起测量电池的使用。在委员会工作五个区块后，电池的损耗约为 3%。公民对单个块产生的总网络流量为 19.5MB。

现在我们可以根据每块成本和每个公民预计进入委员会的次数来推算每天的成本。在 100 万公民中，大约每 500 个区块就会有一个公民参与，在我们的区块延迟约 90 秒的情况下，相当于每天两次。因此，预计的电池使用量每天小于 2%，并且数据使用量为每天约 40MB。此外，我们同样在一加 5 上测量了每十分钟唤醒一次手机的情况下，执行 getLedger 需要消耗 0.9% 的电池和 21MB 的下载数据。每五分钟唤醒一次手机，需要消耗 1.7% 的电池和 42MB 的数据下载。在总共 3% 的点亮使用和每天 61MB 的数据消耗的情况下，运行 Blockene 应用的用户几乎不会注意到它在运行。

第 10 章 结论

Blockene 首次实现了高吞吐量区块链，成员在可忽略不计的资源使用下在智能手机上执行区块验证和共识，从而开扩了一个更大的现实应用程序类。它们受益于区块链的安全性和去中心性的特性。结合新颖的架构、几种新技术、仔细的安全推理，Blockene 能同时提供三个相互冲突的属性：大规模参与、高事务吞吐量和成员节点的低资源使用。