TOT白皮书中文版

比原313社区基于Bytom侧链的安全可扩展的链上生态系统

Bytom 313社区

注意：本项目不作私募，只做技术探讨

摘要

尽管Bytom(比原链)广受关注，但它的底层技术仍处于早期阶段，远未获得大规模采用。 目前区块链技术公认的问题之一是可扩展性：比原链之前也提出了自己的主侧链模型，但是现有的侧链模型承载的应用少，且还不具备迅速处理大量交易的能力，诸如支付宝，微信支付和数字资产交易所这样的能够产生大量交易的的平台几乎不可能被部署到现有的比原链上。所以我们313社区提出了基于比原的社区侧链。

我们的出发点是构建一条高性能的比原侧链，弥补比原UTXO模型和智能合约的不足。因此我们提出了，这是一种创新且安全的区块链模型，旨在提供横向扩展性和高交易吞吐量。通过实施我们创新的分片架构和共识机制，TOT(Three One Three)系统可以更好的承载应用。

我们主要对以下几点内容进行了讨论：

1. **高效分布式分片（EDS）** – EDS是一种创新的分片方案，通过结合客户端–服务器随机找寻机制和加密分类进行的领头节点选举，保证其可生成使足够大的分片，并具有较强的偏差抗性。
2. **TransEpoch** – 在epoch转换期间验证者到分片的分配，同时保持交易的可操作性。
3. **Atomix** – 一种新颖的两步分片间原子提交协议，保证拜占庭设置中的交易原子性。
4. **可信度证明（PoB）** – 一种开创性的拜占庭式共识协议，采用Believable–First（置信度优先）的原则，保证系统的安全性和活跃性，同时大大提高了交易吞吐量。参与出块的门槛低，社区的每一个成员都可以成为TOT出块节点，组织节点和被选节点。
5. **微型块（MSB）** – 一种新颖的、可最大限度地减少验证人的存储和引导成本的机制。

目录

1 背景

TOT是为服务导向的生态系统提供底层基础建设。TOT平台不仅为用户提供了一种完全去中心化的方式来交换在线服务和数字商品，而且还使开发者能够部署面向大量用户群体的大型去中心化应用。通过一系列突破性创新，如高效分布式分片（“EDS”）和置信度优先（“Believable–first”）原则，我们能够在保证安全性的同时极大地提高系统吞吐量。
我们基于分布式系统中的Sharding（分片）的设计概念开发了EDS。分片这一概念在分布式系统和数据库中被广泛采用，以支持并行交易处理。受计算机科学中经典的“分而治之”原则的启发，EDS将整个TOT网络划分为若干称为Shard（分片）的子空间。我们可以将每个分片视为并行运行其自己的共识协议的微型网络。与传统区块链技术让整个网络共同验证同一组交易不同，EDS机制下每个子网络分片将自行组成共识组并同时进行交易验证。因此，即使网络规模和交易数量增长迅速，系统的吞吐量也可以显着提高。而且，为了确保网络的均匀分割，我们开发了一个具有高偏差抗性的分布式随机性生成协议，以便在分片过程中引入无偏且透明的随机性。
除了EDS外，TOT还包含其他许多创新的技术，以支持高性能和灵活性的大规模dApp的部署。它允许开发人员开发多种产品：从传统的、具有垄断性质在线服务提供商的替代品到之前认为不可能的全新商业模式。此外，TOT还提供了还有许多好处，包括：避免网络攻击，高级别数据安全以及不可变属性。
在开发TOT区块链的过程中，我们彻底研究了所有当前Bytom(比原)的区块链解决方案，以便从以前的尝试中学习。

2 相关研究

2.1 状态机复制

简而言之，区块链技术是一种状态机复制协议。每种状态机复制协议必须满足两个重要的属性：

1. *安全性*，即网络中的所有服务器具有相同的交易记录；
2. *活跃度*，即客户的交易被快速提交并记录在日志中。

实现状态机复制有两种本质上截然不同的方式：经典式共识和区块链式共识[18]。经典共识通常采用类似Paxos的共识算法，并在对共识节点有先验知识的权限环境下使用。像淘宝这样的软件公司的服务器就是一个很好的例子，他们的服务器集体使用经典的方式来复制和存储信息，而经典算法建立了基本原则以形成数据排序的共识。

2.2 权益证明PoS

权益证明的概念初次在一个线上区块链论坛中提出[31]，并被PPcoin [22]，PeerCoins [23]和Nxt [5]等几个加密货币所采用。 PoS本质上是每一单位的权益持有者有一票投票权，因此对于每个验证者而言，拥有更多权益将拥有更高的投票权。因此，验证人没有经济动机来损害

整个区块链网络。对于攻击者来说，攻击的代价是巨大的，因为他们必须拥有大部分权益。在早期的开发中，研究这已经发现权益证明共识机制易受“无利害关系（Nothing-at-Stake）”攻击的影响。由于网络中的服务器/节点在投票验证区块时没有任何的激励/惩罚措施，因此为了获得利益最大化，节点将同时对多个区块进行投票，而没有动力去达成收敛的共识，导致区块链安全性遭到破坏。在后来的一些研究中，slasher算法[3]解决了这个问题，它对违规节点实施了惩罚。许多其他项目也被归类为权益证明的应用[1–3,26] [12] [10]。尽管PoS实现了复制状态机协议的活跃性，但它仍然面临中心化和安全问题等挑战。例如，拥有更多代币的验证者将更有可能打包新区块并被奖励更多代币，从而导致潜在的中心化问题。此外，先前的研究结果表明，只有在代币交易不频繁的情况下，权益证明共识协议才能是一个可证实的安全的且配置强大的一致性协议[19]，这可能意味着，为了安全性，权益证明共识协议会存在一个吞吐量上限。

3 区块链架构

TOT区块链的架构与现有的众所周知的区块链相似，比如Bytom(比原链)，节点通过Gossip协议传播数据。系统将数据和状态切分成不同分片。系统中每一个节点都将被包含在一个分片中。未使用的交易（UTXO）存储在相应分片中节点的内存中。这产生了几个新的挑战。

- 如何将系统分成分片。
- 如何在每个分片中达成一致。
- 如何执行分片间交易。

为了公平和安全的解决上述问题，我们必须执行许多随机操作。例如，将节点分配到分片中，选择每个分片中的领头节点。因此，我们必须首先设计一个不可伪造的，无偏的（均匀随机）分布式随机数生成协议。利用随机数生成协议，上述问题可以逐个解决。
在本文的其余部分，我们将介绍用于解决这些挑战的技术和方法。

- 在第4章中，我们将详细探讨 ***分布式随机性协议（Distributed Randomness Protocol, DRP）*** ，当恶意节点的比例低于某个预定的阈值时，该协议是不可伪造且无偏见的。DRP生成的随机数用于将系统分成分片，将节点分配给不同的分片，并选择每个分片中的领头节点。
- 在第5章中，我们介绍了 ***EDS*** ––一种用于形成分片（即所有参与状态记录以及处理的验证节点的一个子集）的新方案，并且通过加密方法结合使用DRP和基于VRF的领头节点选举使得这些分片包含足够多的节点并且耐偏性强
- 在第6章中，我们介绍了 ***TransEpoch*** – 在epoch转换期间将验证节点重新分配至不同分片中，同时保持交易的可操作性。
- 在第7章中，我们展示了 ***Atomix*** ––一种新颖的两步分片间原子提交协议，保证拜占庭设置中的交易原子性。
- 在第8章中，我们介绍 ***置信度证明（PoB）*** – 一种开创性的拜占庭式共识协议，采用置信度优先原则，保证系统的安全性和活跃性，同时大大提高了分片中的交易吞吐量。
- 在第9章中，我们介绍了 ***微型块（MSB）*** – 一种新颖的机制，可以最大限度地减少验证节点的存储和进入成本。

4 分布式随机性协议

传统的生成随机性的方法，如工作量证明机制[13]或可信任的信标[6]具有计算浪费和中心化问题。使用密码学工具来生成分布式随机数不仅可以节省资源，而且可以数据保证安全。
用于TOT区块链中的节点分片分配和领头节点选举时需要使用随机数来保障公平。目前有多种算法可以在分布式系统中生成随机数。在这里我们介绍一个最适合TOT区块链场景中需求的算法。在TOT区块链中，分布式随机数的产生器具有以下要求：

1. 必须以一定比例抵制不诚实的参与者（包括客户端和服务器）。详细来说，当不诚实的参与者比例低于一定数量时，系统能够继续运行，并且不会发生任何不良情况。
2. 除了可忽略的概率外，最终的随机数必须是不可伪造和无偏的（一致随机的）。

3. 不诚实的参与者不能多次尝试产生有利于自己的随机数，即使在多个不诚实的参与者串通的情况下也无法做到。
4. 第三方能够验证输出结果是否为诚实的运行协议所产生（即验证它是否满足上述所有要求）。

为了达到这些要求，我们提出使用称为分布式随机协议（DRP）[24]的客户端 – 服务器协议，其中客户端通过非交互式零知识证明（NIZK）和可公开验证的秘密共享（PVSS）与一组服务器进行通信而产生一个不可伪造的，均匀分布中的随机值。在协议的运行时，在协议完成并显示最终的随机数之前，协议中的任何参与者都无法获知有关最终输出的任何信息，从而确保不诚实的客户端无法尝试多次生成随机数字而选择更有利的随机数。

协议由两个阶段组成 – 随机数的生成和随机数的验证。简单来说，它的工作原理如下：最初，客户端通过向所有服务器广播包括随机生成的均衡分组的消息来启动协议运行。在第一阶段，每台服务器都会生成一个随机输入值，并仅使用PVSS为同一组的其他成员创建“共享投票”。在收到来自所有服务器的NIZK [25]证明的加密共享投票后或服务器超时后，客户端会从每个组中选择一部分服务器的输入内容。这允许客户端固定每个组的密钥和协议的输出结果。在第二阶段，一旦客户端收到全局对于输入内容的集体签名（CoSi），服务器就会解密并将其共享投票发送给客户端[25]。然后，客户端将恢复的组的密匙合并起来以显示最终的随机输出。

5 高效分布式分片

利用上文介绍的分布式随机协议（DRP），实现高效分布式分片并不困难。但是，由于此协议是由全部验证者共同执行并完成的，该协议只有在没有恶意节点或故障节点的情况下才能正常工作。因此，我们必须为具有恶意节点或故障节点的情况设计后备方案。为了克服这个问题，我们提出了一个解决方案，使用Algorand [9]和Omniledger [8]算法来选举一个领头节点。

5.1 算法 – 使用后备协议进行领头节点选举

算法1： 带有备份节点的领头节点选举协议

输入：

1. 是查看数量计数器
2. 是一个验证者
3. 是的私钥
4. 是当前的epoch
5. 是同步说需要的时间上限

输出：运行DRP的最小有效Lottery的验证者

1. 对于每一个，每一个计算一个方程，使用可验证的随机函数以及参数：其视图和节点的私钥。
2. 然后，在时间限制内，验证者互相传递（gossip）这些lottery。每个验证者在过程中不断收集记录3个他们所接收到最小的lottery。
3. 在时间界限之后，验证者确定他们迄今为止看到的最小值有效的lottery。
4. 最小值有效的lottery对应的验证者被选为领头节点者，而第二和第三最小值有效的lottery对应的另外两个验证者被用作备份领头节点者。
5. 如果选定的验证者成功运行DRP协议，则该节点将输出协议结果以及相应的正确性证明（correctness proof）广播给所有其他验证者。
6. 每个验证者可以使用来计算一种排列方式，并将结果划分为个相同大小的部分，因此决定了从每个节点到分片的映射。
7. 在时间限制 之后，如果选定的验证者未能启动DRP协议，则验证程序将当前运行标记为失败，并在该时间段的其余部分中排除该领头节点。在这种情况下，备份领头节点将被用于

运行DRP协议。如果两位备份领头节点者连续失败，则算法将回退至第1步，整个协议将重新运行。

5.2 分析

领头节点者选举机制提供了与第4章中描述中相同的必要属性。每个验证者，对于每一次视图，在一个epoch内只能生成一个有效的lottery。 DRP协议的设计提供了可扩展性。 由于私钥被保密，因此VRF的输出是不可预知的。 鉴于我们的同步时间界限，所有其他验证者都会在内看到lottery。 如果恶意节点赢得lottery，其能执行的操作也是有限的 – 要么选择合作并按照规则运行DRP协议，要么决定使此epoch失效。如果发生任何恶意/异常情况，恶意节点将被排除参与该epoch的其余部分。

6 Epoch转换期间的可操作性

分片配置方案有很多种，例如静态配置和各种滚动方案。 TOT区块链使用动态滚动方案 – 它会在每个epoch分批更换验证者。这种配置将给TOT区块链一个闲置的时间段，只有在足够的验证者正确启动后，网络才能开始处理交易。许多类型的区块链都没有考虑如何确保系统在此期间可以运作的问题。
由于采用的是动态滚动的机制，每个交换批次的大小就成为过渡期间一个很重要的因素，并与系统的安全性高度相关。当交换批次大小增加时，剩余的诚实验证者数量不足以达成共识的风险相应增加。增加交换批次大小的另一个缺点是下载和引导信息会导致网络压力增加。鉴于我们的威分析模型中最多有的恶意节点，交换批量的最大尺寸应小于全部节点的。
为了在转换/空闲阶段保持完整的可操作性，我们需要一种方法来换出的验证者子集，并用新成员替换[8,24]。我们在Omniledger的方法[8]基础上提出了这一方法。它使剩余的验证者能够在新加入的节点正在下载历史数据和进行启动时继续提供服务。我们介绍如下的节点到分片转换分配协议 – TransEpoch。

6.1 节点到分片转换分配算法 – TransEpoch

算法 4: TransEpoch

输入:
1. 是节点的总数
2. 是每个分片的大小
3. 是换出批量大小，即在已知epoch中已知时间内将被换出的验证者的数量。

输出:
1. 设$k = \log \frac{n}{m}$
2. 对于每个分片，

- 使用从DRP生成的随机输出生成两个种子 以及
- 使用以及得到置换和并将节点划分成个分组。通过这种方式， 确定交换批量映射的节点。
- 用于当前节点，用于新加入的节点。

6.2 分析

在上面介绍的算法中，我们确保了每个分片转换中拜占庭容错（BFT）一致性的安全性。 原因有两个： 首先，我们确定团队的规模。 至少有的验证者在运行共识协议。 其次，通过将与每个epoch相关的随机性加入到产生批量验证者的置换中，我们同时保证了系统不受恶意验证者侵犯。

7 分片间交易

支持分片间交易的机制在我们的系统中至关重要，因为交易很可能发生在多个分片上。我们引入拜占庭分片原子提交（Atomix）协议来确保分片之间的原子性。该协议可防止双重支出并保持交易的一致性。我们的设计是Omniledger算法的变体。[8]
我们首先在UTXO状态模型中提出Atomix。以前的文献和研究已经证明，如果支持UTXO模型，Atomix可以确保我们的分片间交易机制也支持智合约同[27]。
简而言之，当一个交易同时经过a节点的A分片以及b节点的B分片，该算法执行以下操作：

---

1. 在分片A中创建交易TX，并让所有节点验证此交易。
2. 如果交易TX由分片A中的所有节点批准，则该交易会被记录在A的区块链中。与此同时，客户将八卦一份接受证明以认证交易，将a的资金锁定在一个UTXO上，并将其发送给B.
- 如果交易TX被A中的节点拒绝，则资金返回到a。
1. A的区块链将交易TX提交给B的区块链，并在要求接收者分片中的节点验证交易TX。
- 如果交易TX被B中的节点拒绝，则资金返回到a。
1. 如果交易TX被B的区块链中的所有节点批准，则该资金被释放给b。
- 如果TX被所有节点拒绝，该资金会被退回到a。

---

8 共识机制

8.1代币和动机

在TOT系统中，TOT代币与其他区块链系统中的代币一样，用于所有交易的价值载体以及支付使用资源的佣金费用。更重要的是，TOT代币也可被用作计算用户的可信度分数的一个变量。所有TOT代币都将在创世块中生成。TOT的初使值是1000W，后期每跨链一个BTM到TOT网络上，就会在TOT网络上产生一个对应的TOT。如果用户从TOT网络上跨走一个BTM，TOT网络上就会进行销毁。创世区块产生的Tot有35%是平均分给比原313群的用户，主要作为节点合伙人抵押使用。5%会根据比原持仓地址进行空投　剩余的60%作为节点出块的奖励。在TOT生态系统中，TOT代币可用于：

- **支付**：支付商家或其他社区成员提供的服务和商品。
- **佣金**：支付给节点作为运行智能合约，处理消息和交易，使用一般生态系统共享的资源（包括但不限于存储空间，计算能力等）的补偿。佣金可以激励验证者并防止恶意用户通过过度部署智能合约来损害社区的利益。
- **可信度**：TOT代币将被用于计算用户的可信度（在下一节中进行解释）。

此外，作为TOT生态系统的一员，每个用户都可以通过验证交易和贡献资源（例如运行智能合约，提供存储空间等）来获取TOT代币。

正如前面部分提到的，传统的权益证明共识机制面临的一个主要挑战是集中化趋势。为了减轻这种风险，我们引入了Servi作为用户对社区贡献的衡量标准，并鼓励成员为TOT区块链的持续发展做出贡献。它具有以下属性：

- **不可交易**：由于Servi不作为交换媒介，Servi不能以任何方式进行交易或交换。
- **自毁**：区块验证后，系统会自动清除验证者所拥有的Servi余额，使可信度高的节点轮流验证区块，从而保证公平的区块生成过程。
- **自行发放**：在提供社区服务，评估另一方提供的服务或做出其他特殊贡献等一定的贡献后，Servi将自动生成并存入用户账户。

8.2 可信度证明

根据不一样的分片大小，区块链系统会在安全性和吞吐量之间具有天生的折衷。一个拥有大量小分片（包含节点数量小）的系统可以提供更好的性能，但对恶意节点的防御较弱，反之亦然。为了同时保证安全性和提高吞吐量，我们为TOT区块链提出了一个创新的可信度证明（PoB）共识协议。 PoB保证节点运行恶意行为的可能性微乎其微，同时通过大小为一的分片显著提高交易吞吐量。

可信度证明共识协议使用分片间可信度优先（Believable-First）方法。该协议将所有验证者分为两组，一个高可信分组和一个普通分组。高可信验证人在第一阶段快速处理交易。之后，在第二阶段，普通验证人对这些交易进行抽样和验证，以提供最终结果并确保可验证性。节点被选入高可信分组的机会由可信度得分确定，可信度得分由多个因素计算得到，例如：代币余额、对社区的贡献度、网络行为等。一个具有较高可信度分数的个体更有可能被选入高可信分组。可信验证人遵循程序来决定认证的交易及其顺序，并按顺序处理。可信的验证者也可以组成更小的组 – 每组一个验证者。交易将随机被分配到这些可信的验证者中。因此，它们以极低的延迟产生更小的区块。

但是，在此种情况下，由于只有一个节点正在执行验证，因此可能会有安全隐患。因此，一些恶意交易可能是由恶意的验证者犯下。为了解决这个安全问题，我们指定一个抽样概率p，即正常验证者将对交易进行抽样并检测其不一致性。如果验证人被检测为不当行为，将会丢失系统中的所有代币和声誉，而受到欺诈的用户将获得赔偿。可信度优先方法使得处理交易非常快速，因为只有一个（可信的）验证者正在进行验证，并且在正确的参数设置下，不端行为发生的可能性非常低。

在TOT系统中，分片策略文件分别指定高可信和普通分组的大小以及采样概率p。每个epoch开始时，所有的验证者都将通过分布式随机生成协议被分配到不同的分片。它们的状态将从相应分片的最后一个微状态块（MSB）中引导。根据可信度分数，验证者将被分配到高可信组（包含节点数较少）或分片内的正常组（包含节点数较多）。

在第一阶段，由高可信分组处理的交易产生的被验证过的区块。这些区块作为同时运行的普通分组的抽样重新验证输入。普通联盟也结合了来自多个处理组的输入。这可以最大化系统的吞吐量。如果交易成功进行验证，它们将被包含在已完成的块中，添加到分片的区块链中，并最终包含在MSB中。然而，当普通分组检测到任何不一致时，相应的验证交易将被排除在区块链之外，并且签名无效区块的验证者将被检测到并被追究责任。我们将惩罚方案设计得非常苛刻，以便验证人在任何情况下都不会有任何行为不端的动机。如果验证者被检测为行为不当，该验证者将丢失系统中的所有代币和信誉，并且将重新检查之前验证过的所有交易。考虑到极小的恶意激励以及可以量化的验证系统安全性的信心，客户可以在随时实现实时处理。

普通分组的运行基于ByzCoin [7]的拜占庭共识策略，因为它可以有效扩展到成千上万的共识团队成员。 ByzCoin使用集合签名（CoSi）[25]，一种使用多重签名的可扩展密码原语[20]来制作传统的共识算法，如PBFT [4]。 ByzCoin使用组播树来分发区块以提高性能，并回退到星型拓扑以实现容错。它可以确保在即使分片中有一些恶意节点的情况下，分片的所有诚实成员都能够达成特定的通用操作序列，同时保证了系统安全性和活跃性。

为了确保稳定性，我们在可信度有限协议中使用备选方案。当一个分片没有足够的可信验证者来组建高可信分组时，不管是由于暂时停工或者处于生态系统的冷启动阶段，两部验证委员会会变成单步分组，即所有交易均由普通分组直接根据PBFT共识协议处理。

9 区块链存储剪枝算法

当前区块链正面临的另一个问题是区块链存储规模的迅速扩大[8]，这为新加入的验证人带来了繁重的工作负载。区块链一直在遵循相同的模式来存储历史数据。但是，对于高吞吐量区块链系统来说，这是一个至关重要的问题，因为存储需求将会飞速增长。为了尽量减少验证人的存储和冷启动成本，我们使用区块链存储剪枝算法来压缩分片区块链的完整状态。 我们使用基于State Block[8]的微型状态块（MSB）。 我们在下面介绍MSB生成协议。

9.1 算法 – MSB生成协议

算法 6: MSB生成协议

输入:

- 是当前的epoch
- 是当前分片

输出：对于某个中分片中的微状态块

1. 当epoch 结束时，分片领头节点将的所有交易保存在Merkle树中[14]。
2. 分片领头节点哈希Merkle树的根，用表示，并将放入中。
3. 验证人在没有任何正在等待的区块的情况下，在上运行共识机制。
4. 如果的正确性得到验证，则分片领头节点将批准的表头存储在分片的区块链中。
5. 在epoch 结束时，所有节点丢掉的内容，并保留的区块。

9.2 分析

通过对过去的区块进行查阅，我们可以检查新的交易。由于TOT区块链中的每个分片仅存储过去的MSB 头文件并且区块链状态分布在多个分片中，客户端无法通过在该区块中提供检查来证明过去交易的存在。我们通过将存储内容从过去的区块链转移到客户端来缓解这个问题。由于保留了最新的epoch的区块，因此客户可以要求分片的验证者为下一个epoch期间验证的交易创建存在证明。

验证者实质上是为MSB创建更高层次的链，使得交易验证可以从一个epoch的MSB跳到另一个MSB。这个MSB链保留最新的MSB的主体和所有以前的MSB headers。这一点很重要，因为客户如果想验证过去交易的话，需要有一个参考点。我们注意到，MSB可能包含多个指向常规区块header的多跳指针，以减小其证明的大小。

有了MSB，引导新的验证人和同步掉线的验证人变得更加有效率，因为验证人是从最后一个有效的MSB开始并仅回放区块链的最后一部分，而不是从第一个区块或从他们掉线的区块开始回放。如果比特币部署在TOT区块链上，目前带宽引导成本将降低两个数量级。当TOT区块链中出现新的分片时，这是至关重要的。由于采用随机分片机制，验证人会定期更改分片并需要频繁更新，这对区块链存储修剪技术有很大的好处。

参考文献

[1] Iddo Bentov, Ariel Gabizon, and Alex Mizrahi. 2016. Cryptocurrencies Without Proof of Work. In Lecture Notes in Computer Science. 142–157.

[2] Iddo Bentov, Charles Lee, Alex Mizrahi, and Meni Rosenfeld. 2014. Proof of Activity. ACM SIGMETRICS Performance Evaluation Review 42, 3 (2014), 34–37.

[3] Vitalik Buterin. 2014. Slasher: a punitive proof of stake algorithm. Retrieved January 9, 2018 from https://blog.ethereum.org/2014/01/15/slasher-a-punitive-proof-of- stake-algorithm/

[4] Miguel Oom Temudo de Castro. 2000. Practical Byzantine Fault Tolerance.
[5] Nxt Community. Nxt Whitepaper. Retrieved January 9, 2018 from https://bravenewcoin.com/assets/Whitepapers/NxtWhitepaper-v122-rev4.pdf
[6] George Danezis and Sarah Meiklejohn. 2016. Centrally Banked Cryptocurrencies. In Proceedings 2016 Network and Distributed System Security Symposium. DOI:https://doi.org/10.14722/ndss.2016.23187
[7] E. Kokoris-Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford. 2016. Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing. In 25th USENIX Conference on Security Symposium.
[8] Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser†, Nicolas Gailly, Ewa Syta, Bryan Ford. 2017. OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding.
[9] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. 2017. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. In Proceedings of the 26th Symposium on Operating Systems Principles – SOSP '17. DOI:https://doi.org/10.1145/3132747.3132757
[10] G Maxwell And. 2015. On Stake and Consensus. Retrieved January 9, 2018 from https://download.wpsoftware.net/bitcoin/pos.pdf
[11] Ian Grigg. EOS – An Introduction. eos.io. Retrieved from https://eos.io/documents/EOS_An_Introduction.pdf
[12] J. Kwon. 2014. Tendermint: Consensus without mining. Retrieved January 9, 2018 from http://tendermint.com/docs/tendermint.pdf
[13] Loi Luu, Viswesh Narayanan, Chaodong Zheng, Kunal Baweja, Seth Gilbert, and Prateek Saxena. 2016. A Secure Sharding Protocol For Open Blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security – CCS'16. DOI:https://doi.org/10.1145/2976749.2978389
[14] Ralph C. Merkle. A Certified Digital Signature. In Lecture Notes in Computer Science. 218—238.
[15] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. bitcoin.org. Retrieved from https://bitcoin.org/bitcoin.pdf
[16] Rafael Pass, Lior Seeman, and Abhi Shelat. 2017. Analysis of the Blockchain Protocol in Asynchronous Networks. In Lecture Notes in Computer Science. 643—673.
[17] Rafael Pass and Elaine Shi. 2017. The Sleepy Model of Consensus. In Lecture Notes in Computer Science. 380—409.
[18] Phil Daian and Rafael Pass and Elaine Shi. 2016. Snow White: Provably Secure Proofs of Stake. (2016).
[19] Phil Daian Rafael Pass. Snow White: Robustly Reconfigurable Consensus and Applications to Provably Secure Proofs of Stake.
[20] C. P. Schnorr. 1991. Efficient signature generation by smart cards. J. Cryptology 4, 3 (1991). DOI:https://doi.org/10.1007/bf00196725
[21] Jagdeep Sidhu. 2017. Syscoin: A Peer-to-Peer Electronic Cash System with Blockchain-Based Services for E-Business. In 2017 26th International Conference on Computer Communication and Networks (ICCCN). DOI:https://doi.org/10.1109/icccn.2017.8038518
[22] Sunny King And. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. Retrieved 2012 from https://peercoin.net/assets/paper/peercoin-paper.pdf
[23] Scott Nadal Sunny King. 2012. Peercoin. Retrieved January 9, 2018 from https://peercoin.net/assets/paper/peercoin-paper.pdf

[24] Ewa Syta, Philipp Jovanovic, Eleftherios Kokoris Kogias, Nicolas Gailly, Linus Gasser, Ismail Khoffi, Michael J. Fischer, and Bryan Ford. 2017. Scalable Bias-Resistant Distributed Randomness. In 2017 IEEE Symposium on Security and Privacy (SP). DOI:https://doi.org/10.1109/sp.2017.45

[25] Ewa Syta, Iulia Tamas, Dylan Visher, David Isaac Wolinsky, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ismail Khoffi, and Bryan Ford. 2016. Keeping Authorities "Honest or Bust" with Decentralized Witness Cosigning. In 2016 IEEE Symposium on Security and Privacy (SP). DOI:https://doi.org/10.1109/sp.2016.38

[26] V Buterin And. 2015. Casper. Retrieved January 9, 2018 from https://blog.ethereum.org/2015/08/01/ introducing-casper-friendly-ghost/

[27] G. Wood. 2014. Ethereum: A Secure Decentralised Generalised Transaction Ledger. Ethereum Project Yellow Paper. (2014).

[28] Gavin Wood. 2018. ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER. ethereum.github.io/yellowpaper. Retrieved from https://ethereum.github.io/yellowpaper/paper.pdf

[29] 2017. Steem: An incentivized, blockchain-based, public content platform. steem.io. Retrieved from https://steem.io/SteemWhitePaper.pdf

[30] Whitepapers. bitshares.org. Retrieved from http://docs.bitshares.org/bitshares/papers/

[31] Proof of stake instead of proof of work. Retrieved January 9, 2018 from https://bitcointalk.org/index.php?topic=27787.0