Blockstack——一个由区块链保障的命名和存储系统

Muneeb Ali 1）2） Jude Nelson 1）2） Ryan Shea 2）

Michael J. Freedman 1）

1)Princeton大学, 2)Blockstack 实验室

摘 要 在过去的几年中区块链如比特币和域名币和他们各自的P2P网络显示出了重要的应用并且在没有信任的各方的命名系统中建立了承诺。用户可以注册有意义的人名并且安全地与其传输数据。只有注册它们的特定私钥的拥有者可以书写或更新名值对。理论上，许多分布式的系统可以用区块链网络来构建，例如新的分布式版本的DNS和PKI。然而, 由于该技术相对较新, 而且发展迅速, 因此只有很少的生产数据或经验可用于指导设计这种公平交易。

在这篇论文中，我们描述了在域名币区块链之上构建的分布式的 PKI 服务的大型部署的经验。我们提出了各种挑战：网络可靠性、吞吐量和安全性,这都是我们需要克服的难题，尤其在面对域名币区块链上超过33000项的注册和更新还有200000个交易时。此外, 我们还讨论了我们如何设计新的基于区块链的命名和存储系统——Blockstack的经历。我们详细的描述了为什么我们将新系统从域名币网络转换到了比特币网络，并从这个转移中得到了操作上的经验教训。Blockstack是作为一个公开的软件被发行的，目前为55000名用户提供了生产PKI的系统。

**Blockstack: A Global Naming and Storage System Secured by Blockchains**

Muneeb Ali 1）2） Jude Nelson 1）2） Ryan Shea 2）

Michael J. Freedman 1）

1)Princeton University, 2)Blockstack Labs

**Abstract** Blockchains like Bitcoin and Namecoin and their respective P2P networks have seen significant adoption in the past few years and show promise as naming systems with no trusted parties. Users can register human meaningful names and securely associate data with them, and only the owner of the particular private keys that registered them can write or update the name-value pair. In theory, many decentralized systems can be built using these blockchain networks, such as new, decentralized versions of DNS and PKI. As the technology is relatively new and evolving rapidly, however, little production data or experience is available to guide design tradeoffs.

In this paper, we describe our experiences operating a large deployment of a decentralized PKI service built on top of the Namecoin blockchain. We present various challenges pertaining to network reliability, throughput, and security that we needed to overcome while registering and updating over 33,000 entries and 200,000 transactions on the Namecoin blockchain. Further, we discuss how our experience informed the design of a new blockchain-based naming and storage system called Blockstack. We detail why we switched from the Namecoin network to the Bitcoin network for the new system, and present operational lessons from this migration. Blockstack is released as open source software and currently powers a production PKI system for 55,000 users.

# 1引言

密码货币区块链和它们各自的P2P网络在金钱交易之外也是有用的。它们提供了加密的可审核的、仅追加的分类帐, 这些分类账已经用于构建新的、去中心化的 DNS [41] 和公钥基础设施 (PKI) [43] 的版本, 以及其他应用程序, 如文件存储 [23] 和文档时间戳 [15]。因为区块链没有中心节点的信任或失败，所以它们启用了一种新的去中心化式的应用程序和服务, 以最大限度地减少用户将信任放在单个方 (如 DNS 根域名服务器或根证书颁发机构) 的程度。

区块链网络引起了许多爱好者，工程师和投资者的极大关注。事实上，在过去几年中, 有11亿美元投资于区块链创业公司 [19]。随着资本的快速注入,区块链的基础设施正在快速部署 [18], 区块链正在成为公共的公共基础设施, 用于建立去中心化的系统和应用。然而，区块链网络仍处在初级阶段，几乎没有可用的生产数据来指导交易的设计。

许多基于区块链的非金融方面的应用需要一种命名系统可以把可读的名字与任意值关联起来。区块链对命名系统的全局状态提供了共识, 并为状态更改提供了仅追加的全局日志。写入的名值对作为对全局日志的追加只能在新的区块中宣布。全局日志在逻辑上是集中的 (网络上的所有节点都看到相同的状态), 但在组织上去中心化 (没有中央控制日志)。

基于区块链的命名的去中心化的特性带来了极其重要的安全方面的好处，但是目前在某些方面区块链也显示出了技术上的局限。单个的区块链记录通常按千字节 [49] 的顺序排列, 并且无法保存大量数据。创建和更新记录的滞后时间受区块链的写入传播和领导选举协议限制, 并且通常大约需要10-40 分钟 [14]。每轮中的新操作总数受网络节点平均带宽的限制 (对于比特币, 当前平均值为∼1500 新操作每新的圆 [2])。此外, 新节点 需要从一开始就独立地审核全局日志: 随着系统的进步, 引导新节点的时间线性增加。

我们认为, 尽管有这些可伸缩性和性能方面的挑战,区块链为构建安全、去中心化的服务提供了重要的基础结构。篡改区块链的成本随着他们的增长而增长: 今天, 它将需要数亿美元来攻击像比特币这样的大型区块链 [1]。

这些好处促使我们使用区块链构建一个新的去中心化的 PKI 系统。我们的系统使用户能够注册独特的,可读的用户名和关联的公钥, 如 PGP [53], 连同其他的数据。我们的 PKI 系统中不需要任何中央或受信任的一方。本文介绍了我们在域名币网络上运行PKI 系统的经验, 这是迄今为止在区块链上建立的最大的服务之一。我们概述了在超过3.3万项的注册和更新以及在域名币网络上发送超过20万项交易时必须克服的挑战。

我们的生产部署导致了许多有趣的体验, 我们观察并分析了以前未发现或记录的网络异常和安全问题。**我们发现了一个关键的安全问题, 即单个矿工在域名币网络上始终拥有超过51% 的总计算能力**(参见 [35] 关于矿工的51% 攻击和计算能力的细节)。51%算力攻击是对区块链的最严重攻击之一, 并会影响其安全和去中心化的性质。

此外, 我们还在域名币网络上遇到了与广播交易有关的长期网络问题。网络的可靠性通常取决于 区块链网络的监控和维护的活跃度, 以及网络运营的财务激励。因此, 出于安全和可靠性的考虑, 基于区块链的服务应该使用最大、最安全的区块链,即比特币区块链（在编写文章时）。

我们关于域名币的经验启发我们设计和实施一个新的基于区块的命名和存储系统, 使用了比特币区块链，称为 Blockstack。与以前的基于区块链的系统不同, Blockstack 将其控制和数据层注意事项分开: 它在区块链中只保留最小的元数据 (即数据哈希值和状态转换), 并使用外部数据进行实际的批量存储.Blockstack 通过使用检查点和 "跳过列表" 来限制新节点必须审核以开始的块的集合, 从而实现了新建节点的快速引导。我们将Blockstack 公开发行[13]。

修改像比特币这样的数据分布式系统 (并引入之前没有涉及的新功能) 是相当困难的, 特别是系统仍然需要达成 "共识"。运用 Blockstack, 我们扩展了区块链的单状态机模型, 允许任意状态机, 而不需要在底层的区块链中进行打破共识的更改。这个设计在我们的工作之前是非直观的。事实上, 过去三年的标准做法是将主要的比特币区块链分叉, 以增加新的不同的功能。我们有关域名币区块链的经验表明, 启动新的、较小的区块链会导致安全问题 (如减少了攻击网络所需的计算能力), 因此应尽可能避免。

本文作了如下贡献：

· 我们提出了除比特币之外的区块链的安全和网络可靠性的第一个分析，并报告了区块链的一个重要的安全问题, 其中一个主要的区块链即域名币中, 有一个矿工拥有超过51% 的算力并保持了数月。

· 我们报告了合并挖掘（一种流行的保护较小的区块链的方法）, 目前在实践中的失败。专用于区块链的总计算能力目前不足以维护多个区块链的安全。

· 我们介绍了 Blockstack 的逻辑分层的设计：虚拟链, 它为区块链引入了新的功能, 而不需要在底层的区块链中进行打破共识的更改。

· 我们提出了在底层的区块链故障的情况下从一个区块链迁移到另一个的模板, 以及我们的系统从域名币到比特币成功迁移的经验。这是在区块链上运行的系统的第一个跨链迁移。

# 2动机和背景

在本节中, 我们描述了构建没有中心信任节点的命名系统的动机, 并提供了区块链的相关背景。本文我们使用术语*命名系统*意味着 (a) 名称是**可读的**, 并且可以由人挑选, (b) 名值对有**很强的所有权**-即它们可以由秘钥对拥有, 而 c) **没有中央信任方**或故障点。建立一个带有这三个特征的命名系统根据 Zooko 不可能三角 [32] 被认为是不能实现的，并且多数传统命名系统只提供这三个特征中的两个[31]。域名币 [41] 使用基于区块链的方法提供了第一个具有全部三个特征的命名系统:可读、强所有权和分散。

## 2.1关于区块链的背景

区块链提供可公开写入的全局追加日志。对全局日志 (称为交易) 的写入被组织为区块, 每个块将多个交易打包成一个单元写入. 写入全局日志需要以交易费用的形式进行付款。参与区块链网络的节点遵循领导者选举协议, 以决定哪个节点可以写入下一个块并收集各自的交易费用。并非网络中的所有节点都参与领导选举。积极竞争成为下一轮领导的节点称为矿工。在每一轮的开始, 所有矿工开始运算一个新的从上一个区块产生的问题, 第一个解决问题矿工可以写下下一个区块。在比特币中, 这些计算问题的难度是由协议自动调整的, 使得大约每10分钟产生一个新的区块。参见 [14] 关于如何区块链如何工作和达成共识的更多细节。

## 2.2域名币的命名系统

域名币是比特币的第一个分岔并且是除比特币以外仍然运作的最早的区块链, 在2016年5月以密码货币资本总额达到500万美元[6] (密码货币的资本总额是交易所交易的货币价值乘以其存在的货币数量)。启动域名币的主要动机是创建一个类似的 DNS系统, 它用区块链将域名映射到DNS记录 [41] 中来代替DNS根服务器。鉴于区块链没有中心信任节点, 基于区块链的DNS更难审查，而且注册的名字在没有得到拥有者各自的私钥时不能被其他人获取[31]。更改存储在区块链中的注册名称需要非常高的计算资源, 因为重写区块链数据需要工作量证明 [8]。在我们的工作之前, 通常的做法是开始新的区块链(通过从比特币分叉) 引入新的功能, 并进行相应的服务/应用程序所需的修改, 这就是域名币所采取的方法。

与 DNS 一样, 注册新名称需要成本。注册费不利于人们注册许多他们实际上并不打算使用的名字。在域名币中, 注册费的接收者是一个 "黑洞" 密码地址这就使得那些钱不能被追回[31]。域名币定义了一个定价函数, 用于决定命名注册的费用随时间而变化的方式。域名币支持多个命名空间 (如 DNS 中的TLDs), 并且相同的定价和名称过期规则适用于所有命名空间。按照约定, d/ 名称空间用于域名。

在域名币中, 名称注册使用两阶段提交方法, 其中用户首先**预购**一个名称哈希值, 然后通过显示实际的名称和关联的值来**注册**名值对。这样做是为了避免对未予确认的名称注册的提前交易[31]。名称登记在固定的时间后过期, 以新的写入块 (目前为3.6万块, 转换为大约8个月) 来度量。域名币还支持更新与名称关联的值以及所有权转移。

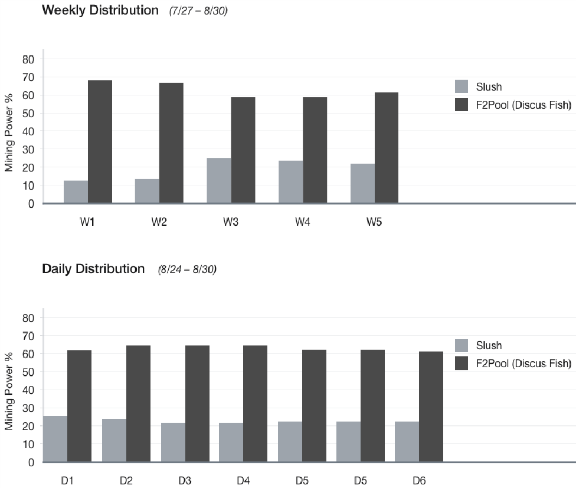
## 2.3基于区块链的PKI系统

我们使用域名币建立一个 PKI 和身份系统, 称为Blockstack ID, 通过在之上启动新的命名空间u / 。我们规定了用于发布公钥的格式, 如 PGP [53], 以及区块链[3]中的其他配置文件数据。这类似于规定DNS 记录的格式。域名币已经支持可读的名称和注册名值对。域名币为每个名称值对提供了有限的存储, 我们通过使用名值对的链接列表扩展了存储容量。我们还提高了我们的产品系统通过提高域名币的读取性能。

我们在2014年3月启动了一个网络服务 [43], 使人们能够方便地在域名币的u/命名空间上注册名称, 并将配置文件数据与它们关联。在我们的网站服务中, 我们首先以用户的名义注册名称 (同时也支付注册费), 然后将名称转移到用户拥有的密码货币地址。我们的成就是第一个生产 PKI 系统之一, 它使用区块链将用户身份绑定到公钥 (请参见其他系统的6节)。默认情况下, 所有注册的名称都具有 ECDSA 公钥 [28] 绑定, 并且用户的子集也添加了其 PGP 密钥。根据一项研究由Harry et al. [31] 显示，我们的系统在域名币上有容量第二大的命名空间和最大的活跃用户的数量。

# 3.部署域名币的经验

在本节中, 我们介绍了在域名币上运行长达一年的生产系统的经验以及我们面临的挑战。我们介绍了在保护区块链(§3.1, §3.3, 和§3.5)，提高网络可靠性(§3.2)，和部署一致性中学到的教训。这些课程直接影响我们新系统Blockstack (§4)的设计。



1每周和每天的采矿分配

## 3.1区块链的安全

名称所有权的安全性与底层区块链和运行它的软件的安全性密切相关。保障区块链安全的最重要的因素是攻击区块链和篡改最近写入的数据的总成本。矿工经常汇集他们的资源形成一个挖掘池, 这实质上是网络上的超级节点 (在一个矿工节点后面有很多计算能力)。如果单个矿工 (或池) 控制下的计算能力超过了网络的其余部分 (称为51% 攻击), 那么该矿工就有能力攻击网络并重写最近的区块链历史记录、审查交易 (例如, 对于名称注册), 并使用双花攻击窃取密码货币[49]。这是因为它将在大多数时间赢得领袖选举, 并比不同意的矿工用更多的工作量证明产生更长的区块链。在区块链上控制大多数计算能力的成本越高, 区块链就越安全。

我们在2014后期注意到, 一个采矿池在域名币上一直有超过51% 的计算能力。最近, 情况变得更糟, 一个采矿池控制了超过60% 的域名币的计算能力。图1显示了在我们从域名币迁移我们的系统之前2015年8月的每周和每天的采矿权分配。事实上, 我们已经观察到 F2Pool (也称为Discus Fish)在一周内 控制了75% 的计算能力。在这种算力高度集中的情况下，域名币处于由单一的一方有效控制的状态;F2Pool 可以编写大多数新的块, 并可以随意破坏区块链的安全。

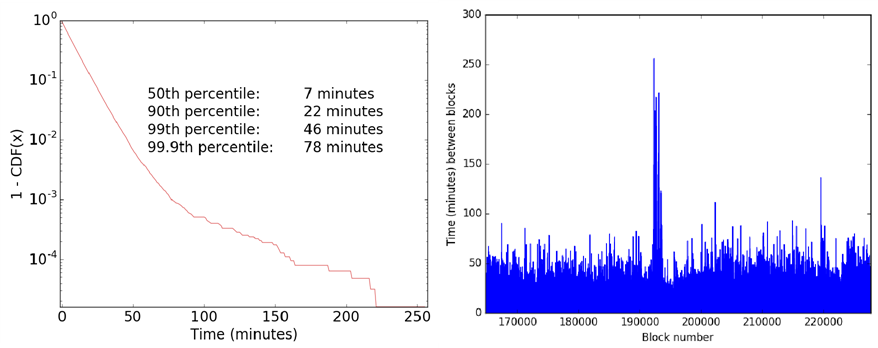
除了原始哈希算力之外, 软件漏洞也可以带来安全问题, 例如,一个域名币的漏洞允许人们从任何人那里窃取姓名 [26]。拒绝服务攻击是另一个攻击媒介;密码货币网络的对等点越多, 遭受拒绝服务攻击后网络的恢复能力就越强。

比特币目前拥有最大数量的计算能力来保证区块链数据的安全。由于有更多的发现漏洞赏金，比特币的代码库比其他区块链更活跃。域名币与比特币相比有更少的同等节点(170 比4600在 2016年1月 [4]), 也使它更容易受到 DDoS 攻击。在目前比特币区块链是迄今为止最安全的区块链。然而, 要为比特币引入新的功能是非常困难的, 因为这需要打破共识机制(3.4 节)。

**经验 #1: 在区块链安全性和向区块链引入新功能之间存在基本的权衡。**启动新的区块链网络是开发人员通常引入比特币不提供的新功能的途径, 例如, 许多新兴应用程序都感兴趣的命名系统。然而, 新的区块链比比特币的安全程度要低得多。在第4节中, 我们引入了Blockstack, 通过创建虚拟链将新功能作为比特币顶层引入, 从而克服了这种权衡。

## 3.2 网络可靠性和吞吐量

我们的 PKI 系统的吞吐量 (可以注册/更新的项目数) 直接取决于底层区块链的吞吐量。每小时可以执行的新注册/更新操作的数量受底层区块链每小时可发送 (和确认) 的交易数的限制。同样, 如果底层区块链不能可靠且一致地执行操作, 则会影响 PKI 系统的可靠性。

**网络延迟峰值**: 作为比特币的分叉,域名币与比特币共享许多协议属性, 包括10分钟的平均领导选举时间 ("延迟目标") 和区块大小的1MB 带宽限制 (授予∼1000 交易的吞吐量每区块)。图 2 (a) 显示, 自从我们在2014年3月推出我们的 PKI 系统以来,域名币基本在网络延迟上表现良好。正如所期待的, 大多数新的区块是在10到40分钟内写的 (在比特币中也需要类似的时间[14])。图 2 (b) 显示2014年8月下旬发生的事件 (在区块数192000时),在几周内网络延迟数飙升 (构建1000 个区块需要大约一周)。在调查了问题并与域名币开发人员进行讨论之后, 我们发现延迟峰值是由域名币中的软件问题引起的。网络上的某个人正在发送许多交易，而且每个交易具有大量的数据字段。这给矿工们带来了很大的执行困难, 他们的域名币守护程序不断地崩溃。没有稳定的矿工节点, 区块没有得到及时的追加。这表明, 意外的协议或软件问题可能会触发网络延迟问题。在此期间, 我们注意到, 我们的 PKI 系统的新用户注册的速度会减慢, 用户投诉也会激增。

2域名币网络中的峰值延迟

**网络吞吐量下降**:在2014年9月初, 就在延迟峰值事件发生后, 我们注意到我们的交易没有被连续的多个区块接受, 而且在一段时间后, 将会在一个封装了大量交易的单一块中被接受。我们注意到很多新的区块都没有交易。这个问题持续了一个多星期, 图3绘制了我们试图发送的事务数 (显示为 "tx 目标") 对比网络接受的事务数。网络延迟是完全正常的 (如图3所示), 但由于新块中没有事务, 网络的吞吐量会下降。我们尝试升级软件和转播的交易, 但问题仍然存在。我们的结论是, 有一个大型的采矿池, 要么是故意拒绝, 要么是无法在其所写的区块中打包交易。只有当其他矿工被选为写新的区块时我们的交易才能得到包装,。在下一节中, 我们将更详细地讨论此问题。

**经验#2: 当前最大的公用区块链网络 (比特币) 的网络可靠性与有很长备用区块链的网络可靠性之间存在显著差异。**比特币网络的问题比域名币和其他较小的区块链影响更多的用户和企业。我们的工作是对除比特币之外的区块链网络的可靠性作第一次分析。

## 3.3 潜在的自私采矿

我们在事件中注意到矿工不接受我们的交易时 (3.2 节) 看起来类似于一个自私的采矿攻击 [22]。在一次自私的采矿攻击中, (a) 矿工需要拥有大量的采矿权 (超过 33%), (b)在很短的时间内人们就会注意到区块后面会有很长的延迟, (c) 而且会有很多被否决的区块。我们注意到了所有这些迹象, 并认为, 单个矿工异常高的计算能力导致了类似于自私开采的条件。那就是说, 矿工能够比其他人更快地计算出新的区块并且快速地将他们连接到区块链中。

**经验 #3: 自私采矿不仅是一种理论攻击, 而且类似的行为在生产区块链中也可以观察到。**这是第一次在网络生产中收集的数据显示了自私采矿行为的迹象, 不管矿工是否有意攻击网络.

## 3.4 打破共识的变化

对于主要更新, 如更改名称定价,域名币需要一个 "硬叉", 即网络上的每个人都必须升级其软件，而以前版本的节点不能再参与网络。有趣的证据表明, 很难让矿工们升级他们的软件, 因为他们没有足够的动力去维护一个像域名币这样小的密码货币，这也不是他们经营采矿池的主要原因。我们监测域名币网络的经验表明, 每当

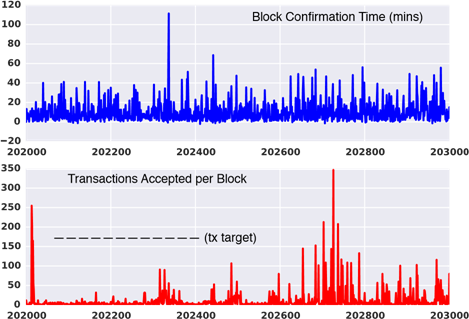


图 3: 域名币网络中的吞吐量下降。我们试图发送的交易数显示为 "tx target ".

在域名币上发布软件更新时, 算力就会有很大的波动。事实上, 我们注意到, 最近域名币核心升级后 [42],（这是一个主要域名币守护进程的升级）许多矿工退出并再也没有回归。

**经验 #4: 除工程问题之外, 由于所涉各方的基本激励结构, 打破共识的变化是复杂的。在密码货币之前, 系统设计者从未处理过打破共识的变化，这对他们来说是一个新的挑战。**用于密码货币网络的软件升级我们应该: (a) 将其他升级和打破共识的升级分离作为为软件工程原则 (比特币最近开始在其代码库中执行此操作 [11]) (b) 尝试给矿工对软件升级付出的成本 (工程时间)同等的奖励。对升级的阻力也存在于比特币中, 但这种阻力对于小区块链的长链来说就会被放大。在4节中, 我们描述了 Blockstack 如何解释这些激励措施, 并介绍了不需要矿工升级软件的新功能。

## 3.5 合并采矿的失败

区块链的安全性取决于矿工的相对计算能力, 以及单个方雇佣超过剩余网络的算力的成本。然而, 新的、较小的区块链有一个引导问题: 在新的区块链形成的最初几天,单个方比较容易接管它, 因为区块链的总计算能力还不够大, 无法防止这种情况。为了解决这个问题, Satoshi Nakamoto (比特币的创造者) 介绍了 "合并的采矿" [39], 在那里一个交替的区块链可以允许比特币矿工参与新的网络, 无需他们花费额外的计算周期。矿工们可以在新的区块链上赚取额外的利润而不增加计算开销。通过合并开采的密码货币,区块链的安全性通常是 "主区块链" 的一个子集, 因为实际上并非所有主区块链的矿工都经历了设置合并采矿的麻烦。

域名币转换为与比特币合并采矿来增加它的区块链的安全 [31]。域名币是最早和最大的合并开采的密码货币并启发了其他密码货币。我们最主要的发现之一是：合并采矿目前在实践中是失败的: 领先的合并开采的区块链——域名币,易于遭受51% 攻击 (第3.1 节)。此外, 合并采矿提供了一种虚假的安全感。F2Pool 控制了比特币的30-35% 算力, 但通过合并开采, 他控制了超过 60%的域名币的算力, 使得域名币易受51% 攻击。除非合并开采的密码货币能始终吸引主区块链的较多矿工来支持他们的软件, 否则合并开采将不会它保证免于51% 的攻击。

**经验 #5: 在区块链的当前发展阶段, 没有足够的计算周期专门用于挖掘以支持多个区块链的安全。**区块链相对于比特币的相关金融资本支持这一论点: 截至 2016年2月, 比特币拥有59亿美元的市场上限, 占所有500 多个区块链市值的 89%, 而第二和第三大市场仅是比特币的3.2% 和2.6% [6]。虽然技术成熟后可能会有多种安全的区块链, 但在最近, 比特币会是唯一攻击成本高昂的区块链。

## 3.6 总结

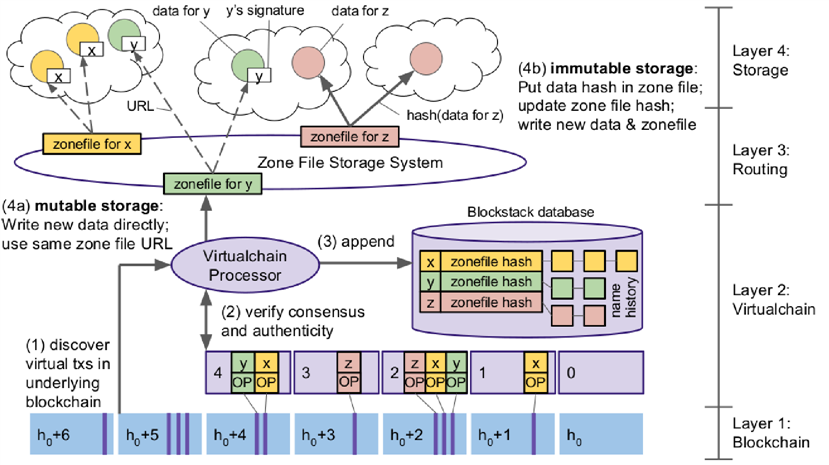
域名币可以说是最早用区块链解决命名问题的。但在考虑上述所有因素后, 将我们的 PKI 系统从域名币转移到比特币是一个很合理的决定。通过我们的经验, 我们坚信分布式的应用程序和服务需要在最大、最安全、最积极维护的区块链上。目前, 在这**些安全要求方面, 还没有其他区块链比比特币更符合要求。

图4 Blockstack 的体系结构概述。区块链记录给出 (名称、哈希) 映射。在路由层查找哈希值以发现到数据的路径。由名称所有者的公钥签名的数据存储在云端中。

# 4 Blockstack 的设计

Blockstack 的目的是在区块链的一层上搭建一个命名系统。在本节中, 我们将描述 Blockstack 如何使用底层的区块链, 并介绍它如何应对当代区块链的技术局限。

## 4.1 挑战

用区块链构建系统带来了挑战：

**· 对数据存储的限制：**单个区块链记录通常按千字节 [49] 的顺序排列, 并且无法保存大量数据。此外，区块链的日志结构意味着所有状态更改都记录在区块链中。参与网络的所有节点都需要保留区块链的完整拷贝, 这就将区块链的总大小限制在当前的商品硬件所能支持范围内。截至 2016年5月, 比特币节点需要将69GB 的总磁盘空间奉献给区块链数据, 以便与网络保持同步。

**·书写速度慢：** 交易处理速率由区块链的写入传播和领导选举协议限制, 并且与前导节点即矿工[14]宣布新块的速率挂钩。新的交易可能需要几分钟到几个小时才能被接受。

**· 有限的带宽：**每个块的交易总数受限于区块链中的区块大小。为了保持公平, 让所有节点都有机会在下一轮中成为领导者, 所有节点都应该在大致相同的时间接收一个新发布的区块。因此,块大小通常受节点 [14] 的上行链路的平均带宽限制。对于比特币, 当前带宽为每个新块 1MB (1000个交易)。

**·无穷尽分类帐：**区块链的可信性取决于任意人审查区块完整性的能力。随着系统的不断进步和新的问题的出现, 审计成本随着时间的推移呈线性增长, 这使得新节点的启动时间越来越长。我们将此称为无穷尽分类帐问题。比特币区块链当前有约41.3万块, 新节点需要1-3 天的时间从比特币对等方下载区块链, 验证它并启动。

## 4.2 建造概要

Blockstack将命名系统作为一个单独的逻辑层置于底层区块链上运作。Blockstack 使用底层区块链在这个命名系统上达成共识, 并将名称与数据记录绑定。具体地说, 它使用底层的区块链作为发布状态更改的通信通道, 因为对名值对状态的任何更改只能在新的区块链块中公布。依赖于底层区块链 的共识协议, Blockstack 可以为命名系统支持的所有操作 (如名称注册、更新和传输) 提供总命令。

**控件和数据平面的分离：**Blockstack 将通过分隔控件和数据平面, 从与名称关联的数据的可用性中获得名称注册和名称所有权的安全性。

控制平面规定用于注册可读名称, 创建(名称,

哈希值) 绑定, 并创建绑定以拥有加密密钥对的协议。

控制平面由一个区块链和一个逻辑上独立的层组成, 称为 "虚拟链"。

数据平面负责数据的存储和可用性。它的组成包括 (a) 区域文件, 用于通过哈希值或 URL 来查找数据; (b) 用于存储数据的外部存储系统 (如 S3、IPFS[29] 和Syndicate [30])。数据值由各自的名称所有者的公钥签名。客户端从数据平面读取数据值, 并检查数据的哈希值是否在区域文件中, 或者数据是否包含具有名称所有者公钥的签名, 从而验证其真实性。

我们认为, 这种分离是超过域名币的一个显著的改善, 它实现了控制和数据平面的区块链水平。我们的设计不仅大大提高了系统的数据存储容量, 而且还允许每个层独立地进行进化和改进。

**基础区块链的不可知性：**Blockstack 的设计并没有对可以与之一起使用的区块链施加任何限制。任何区块链都可以使用, 但安全性和可靠性属性与底层的区块链直接相关。我们认为,从一个区块链到另一个的迁移能力是一个重要的设计, 因为它可以使其有更大的系统可以运行, 即使底层的区块链被破坏。目前, Blockstack 的核心开发人员决定了哪些底层区块链支持软件的哪个版本。如果他们不愿意迁移的话，单个应用程序可以决定运行他们选择的软件版本, 并在特定的区块链上保留它们的名称空间。第5节提供了有关迁移过程的详细信息。

**构造状态机的能力：**Blockstack 的一个关键贡献是在区块链上引入一个逻辑上独立的层, 它可以在处理底层区块链的信息后构造一个任意的状态机。我们将此层称为虚拟链(4.3.2节)。虚拟链将底层区块链中的交易视为状态机的输入并且有效的输入触发状态的更改。在任何给定时间, 在时间由区块号定义的情况下, 状态机可以处于一个完全的全局状态。随着时间推移，底层区块链中写入新块并更新全局状态。**虚拟链可以引入新类型的状态机, 而不需要底层区块链的任何更改。**在区块链中直接引入新的状态机需要对等方进行升级。升级可能会破坏共识并导致分叉。事实上, 将他们协调很困难[14]。现在，Blockstack 引入了一个表示命名系统全局状态的状态机, 包括谁拥有特定名称以及哪些数据与名称关联。此外, 也可以使用虚拟链概念定义其他类型的状态机.

## 4.3 Blockstack的分层

Blockstack 通过规定一组不受区块链支持的新操作, 在区块链上引入了新功能。Blockstack 有四层, 在控制平面中有两层 (区块链层和虚拟链层), 在数据平面中有两层 (路由层和数据存储层)。

### 4.3.1 第一层：区块链层

区块链占据最低层, 并有两个目的: 它存储 Blockstack 操作的顺序, 并就操作的编写顺序提供共识。Blockstack 操作在底层区块链的交易中进行编码。

### 4.3.2第二层: 虚拟链层

在区块链的上面是一个虚拟链, 它规定了新的操作, 而不需要更改底层的区块链。只有 Blockstack 节点可以感知到这个层, 底层的区块链节点是不可知的。Blockstack 的操作在虚拟链层中规定, 并在有效的区块链事务中作为附加元数据进行编码。区块链节点确实可以看到原始事务, 但处理 Blockstack 操作的逻辑只存在于虚拟链层。

接受或拒绝 Blockstack 操作的规则也在虚拟链中规定。接受的操作由虚拟链处理, 以构造一个数据库将信息存储在系统的全局状态以及任何给定 区块链区块上的状态更改上。虚拟链可以用来制造各种状态机。目前, Blockstack 只规定了一台状态机——一个全局的命名和存储系统。

### 4.3.3 第三层：路由层

Blockstack 将路由请求 (即如何发现数据)与实际数据存储区分开。这就避免了系统从开始就采用任何特定的存储服务的需要, 而是允许多个存储提供商共存, 包括商用云存储和对等系统。

Blockstack 使用区域文件存储路由信息, 它们与 DNS 区域文件的格式相同。虚拟链将名称绑定到各自的哈希值(区域文件) ,并将这些绑定存储在控制平面中, 而区域文件本身存储在路由层中。用户不需要信任路由层, 因为区域文件的完整性可以通过在控制平面中检查哈希值(区域文件) 来验证。

在 Blockstack 的当前运作中, 节点形成一个基于DHT的对等网络 [36] 用于存储区域文件。DHT只存储哈希值（区域文件）在区块链中已经预先广播过的区域文件。这样可以十分有效地将数据列入可以存入DHT的白名单中。由于空间限制, 我们省略了本文中的 DHT 存储的大部分细节;与 Blockstack 设计相关的重要方面是, 路由 (无论从哪里获取) 都可以被验证, 因此也不能被篡改。此外, 因为区域文件的大小相对较小 (每个文件4KB) ，大多数服务器都保留了所有区域文件的完整副本。保存路由数据的完整拷贝只会在存储区块链数据的上引入最低限度的存储成本。

### 4.3.4 第4层：存储层

最顶层是存储层, 它承载了名值对的实际数据值。所有存储的数据值都由名称的各自所有者的密钥签名。通过在区块链之外存储数据值, Blockstack 允许任意大小的值, 并允许各种存储后端。**用户不需要信任存储层**, 因为他们可以从控制平面中验证数据值的完整性。

有两种使用存储层的模式, 它们在验证数据值的完整性方面有所不同;Blockstack 同时支持两种存储模式。

(a) **可变存储区**是存储层的默认操作模式。用户的区域文件包含指向数据的 URI 记录, 并构造数据以包含用户私钥的签名。写入数据涉及签名和复制数据 (但不包括区域文件), 读取数据涉及获取区域文件和数据, 验证哈希值(区域文件) 是否匹配Blockstack 中的哈希值, 并使用用户的公钥验证数据的签名。这使得写入速度与签名算法和底层存储系统允许的一样快, 因为更新数据不会改变区域文件, 因此不需要任何区块链交易。但是, 读取器和写入器必须采用数据版本控制方案, 以避免使用陈旧的数据。

(b) **不可变存储**与可变存储类似, 但额外在包含哈希值(数据)的区域文件中放置 了TXT 记录。读取器通过获取数据并检查哈希(数据) 是否在区域文件中验证数据的完整性, 以及验证数据的签名和区域文件的真实性。此模式适用于不经常更改的数据值, 以及有必要验证读取器是否看到最新版本的数据值时。对于不可变存储, 对数据值的更新需要底层区块链上的新交易 (因为区域文件必须修改为包含新的哈希值), 这就使得数据更新比可变存储慢得多。

图5状态和名称的转换

## 4.4 命名系统

Blockstack 使用它的四层来实现一个完整的命名系统。名称由底层区块链及其关联私钥的加密地址拥有(例如, 比特币 [14] 中使用的基于ECDSA的私钥)。与域名币一样, 一个用户预订, 然后用两步注册一个名称,在不向世界显示名称的情况下获得一个名字, 并允许攻击者在获取名称时与用户竞争。第一个成功写入预定和注册交易的用户拥有该名称。此外, 当注册名称后, 任何以前的预订都将无效。注册名称后, 用户可以通过发送更新交易并将新值上传到存储层来更新名值对, 从而更改名值绑定。名称转让操作只更改允许对后续交易进行签名的地址, 而撤消操作来禁用名称的更多操作。

命名系统是通过定义状态机和虚拟链中的状态转换规则来实现的。图5显示了名称可以处于的不同状态, 以及状态转换的工作方式。名称被归入到命名空间中, 命名空间是 DNS 顶层范围中的功能等量, 它们规定了名称的成本和更新率。与名称一样, 命名空间必须预定, 然后注册。过期的名称可以重新注册, 并且名称可以被吊销使得它们不能在一定时间内重新注册。

### 4.4.1 命名空间的定价函数

命名空间中任何人都可以创建命名空间或注册名称, 因为没有中央方阻止某人这样做。定价函数规定了在命名空间中创建命名空间或注册名称的成本。明确智能定价函数是防止 "土地争夺" 的一种方法, 它可以阻止人们注册许多他们不打算实际使用的命名空间或名字。Blockstack 使人们能够创建具有复杂定价函数的命名空间。例如, 我们在我们的 PKI 系统中使用 .id命名空间并创建了具有定价功能的 .id 命名空间。在命名空间里： (a) 名称的价格随名称长度的增加而下降, (b) 在名称中引入字母字符也会降低价格。使用这个定价函数, 在价格上john id > johnadam. id > john0001. id。这一函数是受观察到推特用户名中带有字母的短名字更受青睐的现象启发得来的。也可以创建注册名称免费的命名空间。此外, 我们预计, 未来将有一个售卖名字的市场，就像有 DNS一样。对定价函数的详细讨论超出了本文的范围, 我们鼓励读者查看 [31] 以了解定价函数的详细信息。

类似于名称, 命名空间也具有定价函数[13]。为了**在Blockstack 上启动第一个命名空间 .id 命名空间, 我们在比特币中向网络支付了1万美元。这表明, 即使是这个分布式系统的开发商, 也必须遵循 Blockstack 的规则, 并支付适当的费用。**

## 4.5 简单的名称验证

Blockstack 节点可以在任何区块链的区块上独立计算共识哈希值。共识哈希值帮助 Blockstack 节点确定它们在任何给定块中是否具有全局状态的相同视图。每个共识哈希CH(h) 是从区块h的虚拟链操作的序列中的Vh空间构建的，以及一系列的集合共识哈希值Ph定义的：

CH (h)= hash(Vh+Ph)

当Ph= {CH(2i)| i ∈N ,h −2i > = h 0}

h0是第一个块。除了检测两个 Blockstack 节点具有相同的全局视图之外,共识哈希值还处理无穷分类帐问题(在4.1 节中定义)。随着底层区块链的增长, 新的 Blockstack 节点需要在启动之前处理越来越多的区块。

一个新的 Blockstack 节点可以通过在给定的区块编号上使用不受信任的状态信息数据库来引导, 并与可信的共识哈希值CH (h)相结合相同的区块编号。区块编号也称为文献中的区块高度, 它会随着新区块的生成而增加。一个新的 Blockstack 节点可以从不受信任的数据库重建虚拟链, 并在每个区块链区块上重新处理虚拟链操作, 将每个CH(h)重新计算。如果最终共识哈希与hn上的可信的共识哈希值相同，则与hn相关联的数据库是可信的, 并且该节点可以在hn之后开始处理区块。这比从第一块h0开始并获取所有交易的传统方法要快得多, 即使其中的大多数都将被丢弃。

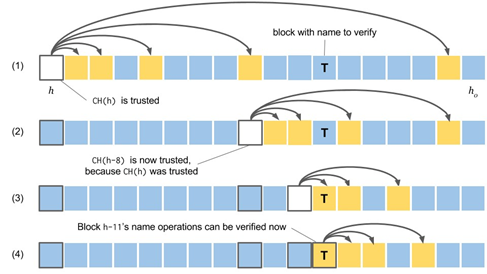
使用后面的可信共识哈希验证先前名称操作的真实性的过程称为简化名称验证 (SNV)。SNV 支持 "瘦客户端", 它可以在不运行 Blockstack 节点或访问完整的区块链历史记录的情况下查询系统的过去状态。支持瘦客户端的使用对于移动设备上的用户非常重要。

图6 SNV的概述。SNV在区块T中查询记录的示例。

因此, 如果用户信任CH (h), 那么她可以查询并验证虚拟链操作Vh和区块h的共识哈希值Ph。CH(h)的结构允许用户从高度h前 < h的区块中验证任何虚拟链操作的真实性仅使用对数个查询数。图6显示了一个 SNV 查询的示例。每行代表区块链, 从左到右递减区块高度顺序(h > h0)。在这里, 用户能够验证目标块中的名称操作的真实性 (用T进行标记)。

在当前的商品硬件上, 用SVN启动新的 Blockstack 节点可能需要1-2 小时,而没有SVN则需要24天。我们的系统可能需要进一步的程序上的改进。

## 4.6 读和写的表现

我们通过 Blockstack 评估读写的性能, 以演示它以具有竞争性的速率读取和写入底层存储的文件。Blockstack 为每个文件添加了一个可忽略的恒定存储空间开销 (压缩后大约占较大文件的5%压缩)。用于加密和压缩的CPU需要开销, 但是由于文件大小差异很小, 因此读写的网络执行与直接访问底层存储服务类似。

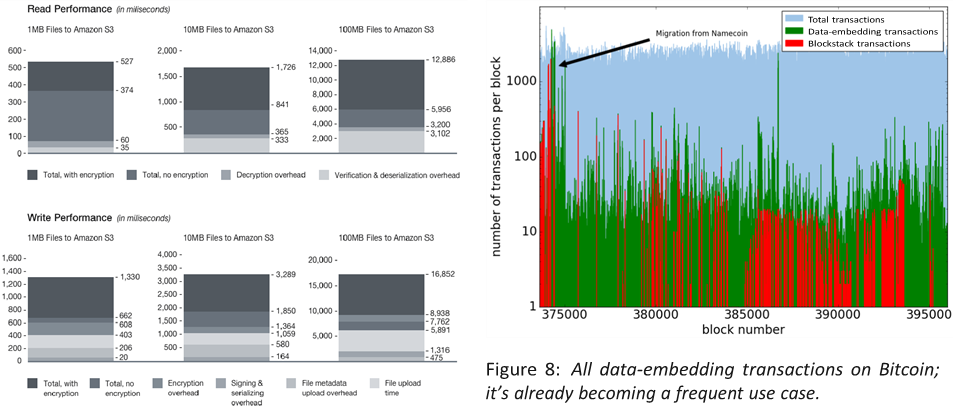
将1、10和100兆字节文件上传到亚马逊 S3 ****的写入性能和开销显示在图7中 (每个试验执行了25次)。我们看到, 对于大型 (100MB) 文件, CPU 绑定开销的顺序是2秒。在我们实现的过程中, 许多性能优化仍然保持不变。类似地, 从 Blockstack 中读取加密文件用S3 作为存储后端, 与直接读取 S3 (图 7) 相比更有竞争力。我们省略了文件下载时间来强调图中的开销。开销的来源、验证签名和解密数据都是受 CPU 限制的, 而在实践中, 性能将在很大程度上取决于广域网的使用范围。

图7 Blockstack的性能开销

# 5.迁移到比特币中获得的经验

图8

我们在40344行 Python 代码中构造了 Blockstack [13], 并且当前使用比特币作为底层区块链来实现。在 2015年9月, 我们成功将我们开发的 PKI 系统[43]中的3.3万用户从域名币迁移到 Blockstack/比特币。这些用户已从域名币上的u/ 命名空间上迁移到Blockstack上的. id命名空间。

Blockstack 使用专用于包含任意数据的特殊字段 [12]在比特币交易中嵌入额外的数据。在比特币交易中嵌入额外数据已经是在比特币上面定义 高级协议的通用方式, 如对手方协议[20]、开放资产协议[44] 等。图8显示了比特币区块链上的数据嵌入协议的最近带宽使用情况。在我们向比特币的迁移过程中, 交易数激增到一万以上, 产生近375000个区块。我们的生产系统 [43] 当前占多数 Blockstack 交易, 占所有在比特币中数据嵌入的交易的26.9% [45]。以下是我们在与比特币网络合作时所做的一些观察：

**网络吞吐量：** 比特币当前支持每秒3到7个交易, 并会产生1MB 的区块大小。即使经过一年的激烈辩论 [52] 在比特币开发商和更广泛的社区之间, 区块大小并没有增加。当我们控制我们的交易量时时, 我们注意到了这些限制, 以便我们的交易不会超过20-30% 的比特币块, 这反过来大大增加了完成登记所花费的时间。当用户扩展到数百万而不是数千个时, 8MB的区块甚至都不够用, 于是社区需要研究在多个链 [10] 中执行注册, 以及在单个交易中打包多个名称的新操作(我们未来的工作领域).

**网络攻击：**在我们向比特币迁移期间, 一家名叫 CoinWallet的英国公司正在对比特币网络进行压力测试 [17]。压力测试包括大量的小型交易, 其中的交易金额太低, 矿工无法打包在一个区块内 (由于旨在防止垃圾信息的协议规则)。这导致在网络上的未经证实的交易数量极高, 我们最终支付了2-3 倍高的交易费用, 才使得我们的交易被矿工打包。这个事件展示了单个参与者如何在网络的其余部分强制征收高昂的挖掘费用 (尽管在这种情况下会有附加的攻击成本)。我们认为区块链受到网络攻击的事件, 像我们之前经历过的或其他 DDoS 攻击一样[37], 很可能会变得更加频繁。对此类攻击的保护是未来研究的一个重要领域。

# 6相关工作

将名称绑定到命名系统中的值是一个值得探索的问题。UIA [24]对全球命名系统及其重要性作了很好的概述。我们鼓励读者给UIA [24] 详细的背景命名系统。不像域名币[41] 或 Blockstack, UIA不尝试提供全球唯一的名称。在如InCommon [27]、OpenID [47] 和 Web 的证书颁发机构这样的身份验证系统中,权威的联盟证明了绑定。但是, Blockstack 不需要联盟。

除域名币之外,区块链如Ethereum [7] 和 BitShares [5] 也支持可读的名字。此外, 侧链[10] 允许允许命名系统作为一个替代区块链,链接到主要的比特币区块链上。所有这些涉及到较小的, 可替代区块链和 Blockstack 的设计都直接使用了最安全的区块链 (比特币)。不基于区块链的 PKI 系统, 如 Keybase [34] 和 CONIKS [38], 实现了一些与 Blockstack 自动化密钥管理相同的目标。主要的区别在于, Blockstack 既为用户提供了对其数据的直接访问和控制, 掌握到全局状态而不用信任任何特定的中央。

在网络系统中, 如果不涉及中央信任方, 就很难获得全局状态 [33], 而Blockstack 能够提供全局状态 (而不仅仅是近似的全局状态)。我们的系统是开放的(“无需许可”), 而现有的宽区域系统象如OceanStore [21]和Bonafide [16] 有一个封闭的对等方 (“需要权限”) 使用 BFT 协议使整个系统发展。Blockstack 不同于允许开放会员的分步式存储系统, 但通过注重分散提供了强于最终数据的共识(比如Shark[9],Pond [48] 和Scatter [25])，同时支持各种各样的外在数据的共识。

面向存储的密码货币区块链, 如 Filecoin [23], Permacoin [40], 和Storj [50] 寻找替代通过将文件作为区块链内的交易集分发的方法，并为矿工的存储证明奖励他们 (而不是工作量证明)。Blockstack 与这些系统不同的是, 通过将承载数据与底层区块链的操作分离, 使开发人员能够使用适合其问题域的存储系统。Blockstack 当前使用简单的 Kademlia [36] 基于DHT 作为发现层, 但其他协议如Chord [51] 或缓存优化如Beehive [46] 也是可行的。

# 7总结

我们在域名币（除比特币之外最早最大的密码货币区块链之一）运行一个网络产品的经历，展示了单个矿工如何持续拥有超过51% 的哈希算力, 以及它的网络可靠性为何远远低于比特币。我们的数据显示, 目前正在使用的数百个区块链[6], 甚至更稳定更流行的区块链像域名币这样，都不适合我们的产品运行。目前, 比特币的安全性远远超过其他区块链。

我们提出了Blockstack——一个基于区块链的命名和存储系统。Blockstack 引入了单独的控制和数据平面, 通过这样做, 它可以在不修改底层区块链的情况下引入新的功能。Blockstack 的设计经历了从最大的基于区块链生产系统生产的一年的时间。我们进行了几项新的改进 (例如, 引入了执行跨链迁移的能力, 更快地引导新节点, 以及防止在慢速度的区块链网络上进行数据更新), 从而使它更容易用公开使用的基础结构建造分布式系统。我们的执行结果表明, Blockstack 可以提供与底层存储服务相当的性能, 并且只增加了少量的 CPU 开销。我们已经将Blockstack 作为公开资源发布[13]。

# 致谢

我们感谢比特币开发者Gavin Andresen首先鼓励我们在比特币之上建立一个类似域名币的系统。 感谢Matt Corallo的反馈协议设计, 和Peter Todd讨论延迟矿工奖励和相关的安全影响。感谢域名币开发者Daniel Kraft帮助我们调试域名币的问题, Ryan Castellucci帮助我们获得采矿统计, Guy Lepage 为我们生成性能曲线图, 和Riccardo Casatta收集数据嵌入交易的统计信息。我们还感谢Larry Peterson, Lakshmi Subramanian, Andrea LaPaugh, 我们的指导人Mohit Aron, Jon Howell, 和 无名的USENIX ATC评论者们对我们的有用建议和反馈。最后, 我们要感谢整个 Blockstack 开源社区 (http://blockstack.org) 的各种讨论, 错误报告, 和在Blockstack产品准备发布阶段Github的邀请。

参考文献

[1] Bitcoin hashrate. https://blockchain.info/

charts/hash-rate.

[2] Bitcoin transactions per blocks. https://blockchain.

info/charts/n-transactions-per-block.

[3] Blockstack ID format, version 2. https:

//blockstack.org/docs/blockstack-profiles.

[4] Crypto-currencies stats – active nodes. https://

bitinfocharts.com.

[5] Bitshares namespaces, 2016. http://docs.

bitshares.eu/namespaces/index.html.

[6] Cryptocurrency market cap, Jan. 2016. http://www.

coincap.io.

[7] A next-generation smart contract and decentralized ap-

plication platform, 2016. https://github.com/

ethereum/wiki/wiki/White-Paper.

[8] Adam Back. Hashcash - A Denial of Service Counter-

Measure. Tech report, 2002. http://www.hashcash.

org/papers/hashcash.pdf.

[9] S. Annapureddy, M. J. Freedman, and D. Mazieres.

Shark: Scaling file servers via cooperative caching. In

Proc. 2nd NSDI, Boston, MA, 2005.

[10] A. Back, M. Corallo, L. Dashjr, M. Friedenbach,

G. Maxwell, A. Miller, A. Poelstra, J. Timon, and

P. Wuille. Enabling Blockchain Innovations with Pegged

Sidechains. White paper, Blockstream, 2014. https:

//blockstream.com/sidechains.pdf.

[11] Bitcoin Core Developers. Bitcoin core version 0.10.0

release notes: Consensus library, Feb. 2015. https:

//bitcoin.org/en/release/v0.10.0.

[12] Bitcoin.org: Bitcoin developer guide, 2015. http://

bitcoin.org/en/developer-guide.

[13] Blockstack source code release v0.10, 2016. http://

github.com/blockstack/blockstack-server.

[14] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll,

and E. W. Felten. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In 2015 IEEE

Symposium on Security and Privacy, SP 2015, San Jose,

CA, USA, May 17-21, 2015, pages 104–121, 2015.

[15]Chainpoint white paper. https://tierion.com/

chainpoint.

[16] B.-G. Chun, P. Maniatis, S. Shenker, and J. Kubiatowicz.

Tiered fault tolerance for long-term integrity. In FAST,

pages 267–282, 2009.

[17] Coindesk. Bitcoin network stress test could occur next

week, Sep 2015. http://coinde.sk/1Ku5oWc.

[18] Coindesk. State of bitcoin 2015: Ecosystem grows

despite price decline, 2015. http://coinde.sk/

1tJDDvv.

[19] Coindesk. State of blockchain q1 2016: Blockchain

funding overtakes bitcoin, May 2016. http://www.

coindesk.com/state-of-blockchain-q1-2016/.

[20] Counterparty protocol specifications. http:

//counterparty.io/docs/protocol\_

specification/.

[21] P. Eaton, H. Weatherspoon, and J. Kubiatowicz. Ef-

ficiently binding data to owners in distributed content-

addressable storage systems. In Security in Storage Work-

shop, 2005. SISW’05. Third IEEE International, pages

12–pp. IEEE, 2005.

[22] I. Eyal and E. G. Sirer. Majority is not enough: Bitcoin

mining is vulnerable. CoRR, abs/1311.0243, 2013.

[23] Filecoin: A Cryptocurrency Operated File Network. Tech

report, 2014. http://filecoin.io/filecoin.pdf.

[24] B. Ford, J. Strauss, C. Lesniewski-Laas, S. Rhea,

F. Kaashoek, and R. Morris. Persistent personal names

for globally connected mobile devices. In Proceedings

of the 7th USENIX Symposium on Operating Systems De-

sign and Implementation (OSDI ’06), Seattle, Washing-

ton, Nov. 2006.

[25] L. Glendenning, I. Beschastnikh, A. Krishnamurthy, and

T. Anderson. Scalable consistency in scatter. In Proceed-

ings of the Twenty-Third ACM Symposium on Operating

Systems Principles, pages 15–28. ACM, 2011.

[26] M. Gronager. Namecoin was stillborn, i had to switch off

life-support, Oct 2013. https://bitcointalk.org/

index.php?topic=310954.

[27] Incommon federation. https://www.incommon.org/

federation/.

[28] D. Johnson, A. Menezes, and S. A. Vanstone. The elliptic

curve digital signature algorithm (ecdsa). Int. J. Inf. Sec.,

1(1):36–63, 2001.

[29] Juan Benet. IPFS - Content Addressed, Versioned, P2P

File System. Draft, ipfs.io, 2015. https://github.

com/ipfs/papers.

[30] Jude Nelson and Larry Peterson. Syndicate: Virtual cloud

storage through provider composition. In ACM BigSys-

tem, 2014.

[31] H. Kalodner, M. Carlsten, P. Ellenbogen, J. Bonneau,

and A. Narayanan. An empirical study of Namecoin and

lessons for decentralized namespace design. WEIS ’15:

Proceedings of the 14 th Workshop on the Economics of

Information Security, June 2015.

[32] D. Kaminsky. Spelunking the triangle: Exploring aaron

swartzs take on zookos triangle, Jan 2011. http://

dankaminsky.com/2011/01/13/spelunk-tri/.

[33] S. Keshav. Efficient and decentralized computation of ap-

proximate global state. SIGCOMM Comput. Commun.

Rev., 36(1):69–74, Jan. 2006.

[34] Keybase. https://keybase.io.

[35] J. A. Kroll, I. C. Davey, and E. W. Felten. The eco-

nomics of bitcoin mining, or bitcoin in the presence of

adversaries. In WEIS 2013.

[36] P. Maymounkov and D. Mazières. Kademlia: A peer-

to-peer information system based on the xor metric. In

Revised Papers from the First International Workshop on

Peer-to-Peer Systems, IPTPS ’01, pages 53–65, London,

UK, UK, 2002. Springer-Verlag.

[37] J. McGovern. Official statement on the last weeks ddos-

attack against ghash.io mining pool, 2015. http://bit.

ly/1nu49vR.

[38] M. S. Melara, A. Blankstein, J. Bonneau, E. W. Fel-

ten, and M. J. Freedman. CONIKS: bringing key trans-

parency to end users. In 24th USENIX Security Sympo-

sium (USENIX Security 15), pages 383–398, 2015.

[39] Merge mining specification. https://en.bitcoin.

it/wiki/Merged\_mining\_specification.

[40] A. Miller, A. Juels, E. Shi, B. Parno, and J. Katz. Per-

macoin: Repurposing bitcoin work for data preservation.

In Security and Privacy (SP), 2014 IEEE Symposium on,

pages 475–490. IEEE, 2014.

[41] Namecoin. https://namecoin.info.

[42] Namecoin core, required software for miners, 2015.

https://github.com/namecoin/namecoin-core.

[43] Onename. https://onename.com.

[44] Open assets protocol. http://www.openassets.org.

[45] Statistics of usage for bitcoin op return. http://

opreturn.org.

[46] V. Ramasubramanian and E. G. Sirer. Beehive:

O(1)lookup performance for power-law query distribu-

tions in peer-to-peer overlays. In Proceedings of the 1st

Conference on Symposium on Networked Systems Design

and Implementation - Volume 1, NSDI’04, pages 8–8,

Berkeley, CA, USA, 2004. USENIX Association.

[47] D. Recordon and D. Reed. Openid 2.0: A platform

for user-centric identity management. In Proceedings of

the Second ACM Workshop on Digital Identity Manage-

ment, DIM ’06, pages 11–16, New York, NY, USA, 2006.

ACM.

[48] S. C. Rhea, P. R. Eaton, D. Geels, H. Weatherspoon, B. Y.

Zhao, and J. Kubiatowicz. Pond: The oceanstore proto-

type. In FAST, volume 3, pages 1–14, 2003.

[49] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic

cash system. Tech report, 2009. https://bitcoin.

org/bitcoin.pdf.

[50] Shawn Wilkinson and Tome Boshevski and Josh Brandof

and Vitalik Buterin. Storj: A peer-to-peer cloud storage

network. Tech report, storj.io, 2014. http://storj.

io/storj.pdf.

[51] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. F.

Kaashoek, F. Dabek, and H. Balakrishnan. Chord: A scal-

able peer-to-peer lookup service for internet applications.

IEEE Transactions on Networking, 11, Feb. 2003.

[52] A. V. Wirdum. A closer look at bip100: The block size

proposal bitcoin miners are rallying behind, Aug. 2015.

http://bit.ly/1VbyoXP.

[53] P. R. Zimmermann. The Official PGP User’s Guide. MIT

Press, Cambridge, MA, USA, 1995.