比特币的学术谱系

加密货币的概念建立在那些被遗忘的研究文献和思想之上

Author	Arvind Narayanan and Jeremy Clark
Translator	Taosheng Shi
WeChat Contact	data-lake
Mail Contact	tshshi@126.com
Organization	NOKIA
Document category	Distributed System
Document location	https://github.com/stone-note/articles

Version	Status	Date	Author	Description of changes
0.1	Draft	10/2/2018	Taosheng Shi	Initiate
0.2	Draft	DD-MM-YYYY	YourNameHere	TypeYourCommentsHere
1.0	Approved	DD-MM-YYYY	YourNameHere	TypeYourCommentsHere

Contents

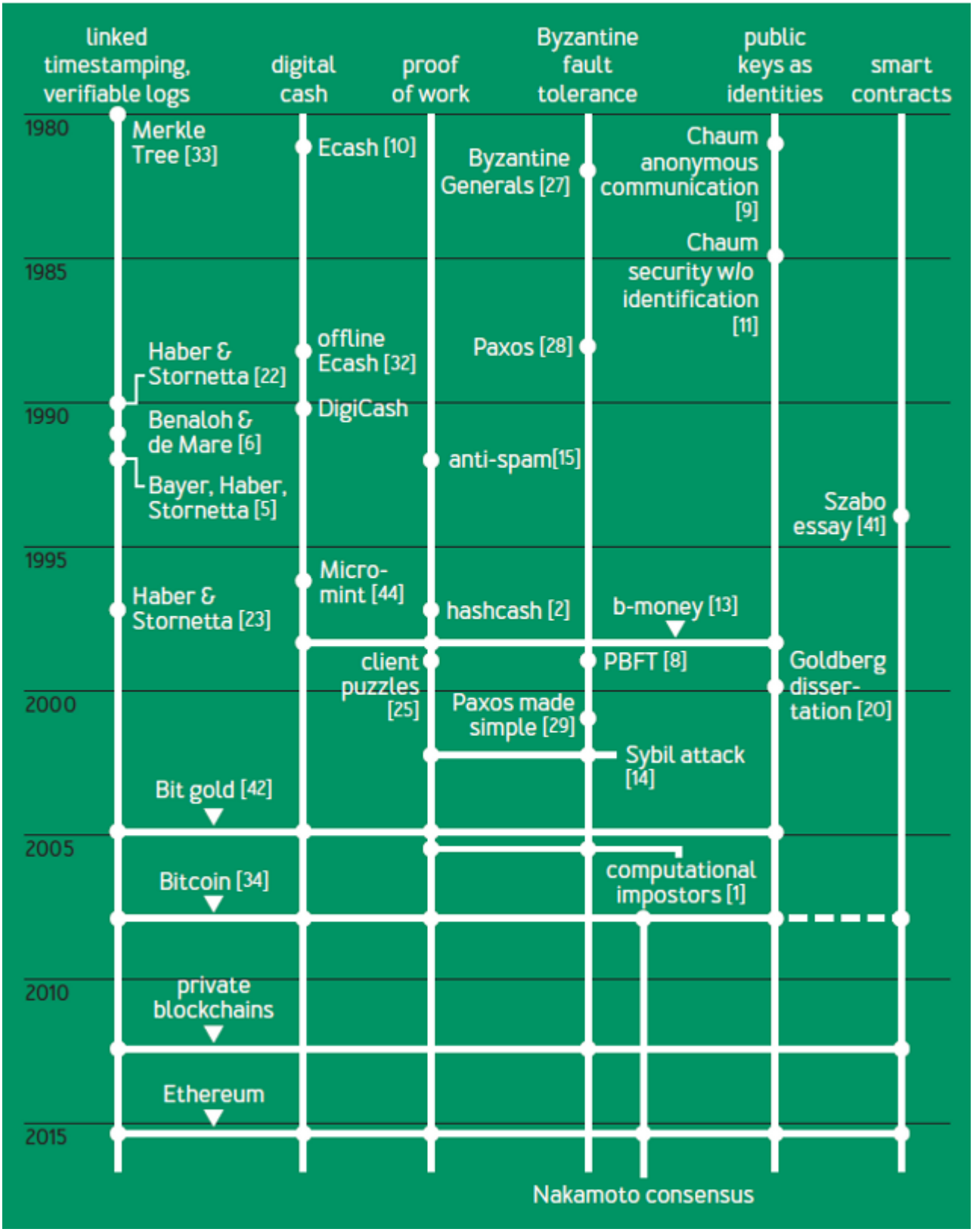
1 引言

如果你已经在媒体上看到过比特币，并且对密码学领域的学术研究有一定的了解，那么你可能会有如下的印象：从 David Chaum 开始(文献 10,12)，几十年来很多人对数字现金(digital cash)的努力研究，最终都没有获得商业上的成功。因为这些工作需要一个集中的，类似银行的服务器来控制系统，而没有任何一家银行愿意为其背书。随着比特币的出现，一种不需要银行的去中心化加密货币(cryptocurrency)方案被提出来，数字现金(digital cash)终于大获成功。比特币的发明者，神秘的中本聪，并不是一个学术界的人，比特币与早期的学术方案并无相似之处。

*译注：David Chaum，数字现金之父，早在 1983 年，在比特币出现的 25 年前，David Chaum 发明了数字现金(digital cash)。

本文认为，**几乎所有比特币的技术组件都起源于 20 世纪 80 年代和 90 年代的学术文献（见图 1）。这并不是为了削弱中本聪的成就，而是指出他站在巨人的肩膀上。**事实上，通过追溯比特币思想的起源，我们可以将中本聪洞察力的真正飞跃归结为一个特定的，复杂的方式——一种组合创新（把底层组件整合在一起）的方式。这有助于解释为什么比特币花了这么长时间才被发明。已经熟悉比特币工作原理的读者可以从这个历史回溯中获得更深入的了解（更多介绍，参见 Arvind Narayanan 等人的 Bitcoin and Cryptocurrency Technologies(文献 36)。**比特币的思想文化史也可以作为一个展示学术界、外部研究者和从业者之间合作关系的研究案例，并且为这些不同群体之间如何彼此合作获益提供经验教训。***译注：比特币可谓组合式创新的典范。

FIGURE 1: CHRONOLOGY OF KEY IDEAS FOUND IN BITCOIN



2 总账本(The Ledger)

如果你拥有安全的总账本，那么将其用于数字支付系统的过程将会非常简单。例如，如果 Alice 通过 PayPal 给 Bob 100 美元，则 PayPal 从 Alice 的账户中扣除 100 美元，并将 100 美元存入 Bob 的账户。这大体上和传统银行的业务差不多，虽然传统银行的业务之间并没有一个共享的总账本。

总账本的概念是理解比特币的起点。它记录了系统内发生的所有交易，并且对系统的所有参与者开放，并被他们信赖。比特币将系统的支付记录转换为货币。在银行业务中，账户余额代表了可以从银行取出来的现金，但一个比特币代表什么？就目前而言，我们只能说比特币代表包含固定价值的交易。

在互联网这种参与者之间可能互不信任的环境中，怎样才能如何建立一个总账本？让我们从简单的部分开始：数据结构的选择。这个数据结构必须满足一定的属性要求——总账本应该是不可变的。更准确地说，只能添加新的交易，不能修改删除，也不能对已有交易重新排序。除此之外，还需要获得总账本状态的密码摘要。摘要是一个简短的字符串，可以避免存储整个总账本。如果总账本被篡改，所产生的摘要必然会发生变化，从而可以检测到篡改。**需要这些属性的原因是：与存储在单个机器上的常规数据结构不同，总账本是由互不信任的一组参与者共同维护的全局数据结构。**这与去中心化数字总账本(decentralizing digital ledgers, 文献 7,13,21)的方法是不同的，在去中心化数字总账本中，参与者维护本地总账本，并且由用户查询这些总账本来解决冲突。

2.1 链式时间戳(Linked timestamping)

比特币的总账本数据结构有修改的借用了从 1990 年到 1997 年间由 Stuart Haber 和 Scott Stornetta 撰写的一系列论文（他们 1991 年的论文还有另一个合著作者 Dave Bayer,文献 5,22,23）。我们能够知道这些历史渊源是因为中本聪在他的比特币白皮书(文献 34)中如此提及的。Stuart Haber 和 Scott Stornetta 的主要工作是处理时间戳的文档化——他们的目的是建立一个“数字公证”的服务。对于专利，商业合同和其他文件，人们希望确定该文件是在某个时间点或者不迟于某个时间点创建的。Stuart Haber 和 Scott Stornetta 的文档概念非常泛化，可以是任何类型的数据。他们确实提到金融交易是潜在的应用，但金融交易不是他们关注的焦点。

在 Stuart Haber 和 Scott Stornetta 方案的简化版本中，文档被不断创建和广播。每个文档的创建者声明一个创建时间（并签名文档）、文档的时间戳和前一个广播文档。前一个广播文档友签署了自己的前一个，所以文档形成了一个很长的倒退链。外部用户不能改变时间戳的信息，因为它是由创建者签名的；创建者也不能在不改变整个信息链的情况下改变时间戳的信息。因此，如果通过可信源（例如，另一个用户或专门的时间戳服务）获得链中的某个项目，那么该时刻之前的整个链是锁定的，不可变，并且在时间上有序。进一步，如果你认为系统因为创建时间错误拒绝你的文档，那么你必须地

保证文档至少与其声称的一样久远。总之，比特币只是借用 Stuart Haber 和 Scott Stornetta 设计的数据结构，然后重新设计了其安全属性（通过增加工作量证明，本文稍后介绍）。

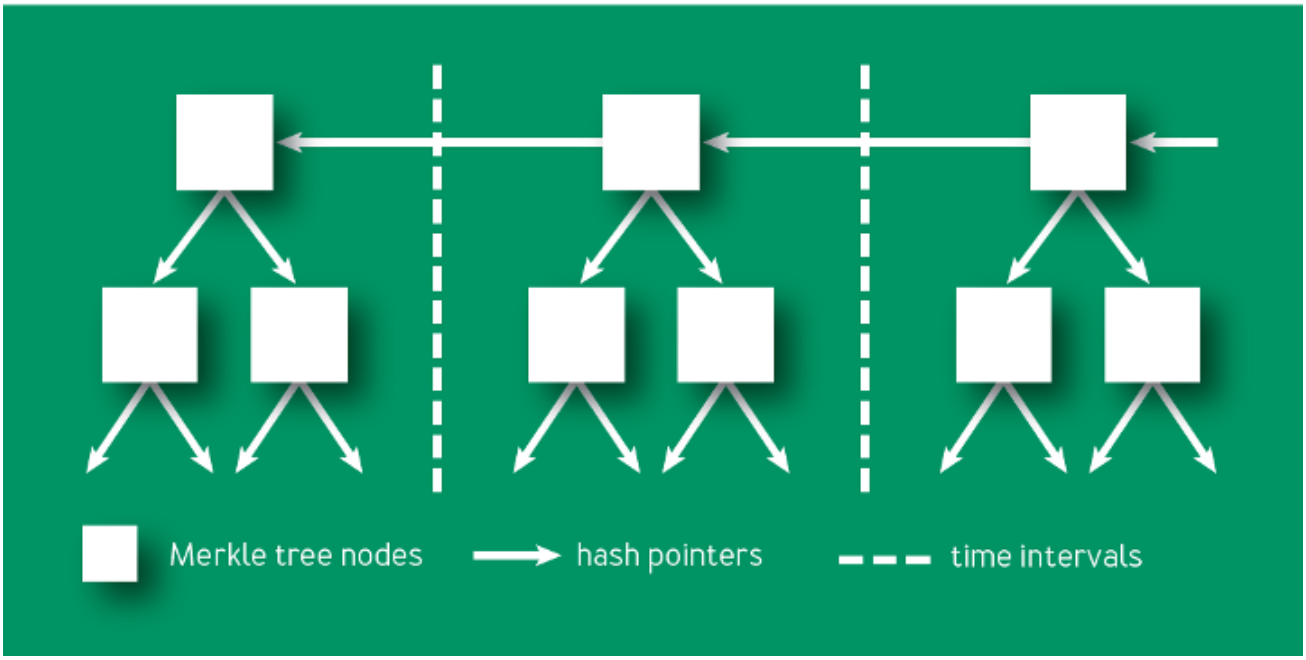
*译注：进一步，如果你认为系统因为创建时间错误拒绝你的文档，那么你必须地保证文档至少与其声称的一样久远。无原文：Further, if you assume that the system rejects documents with incorrect creation times, you can be reasonably assured that documents are at least as old as they claim to be.

在 Stuart Haber 和 Scott Stornetta 的后续文献中，他们介绍了使这个数据结构更加高效的其他方案（其中一些在第一篇论文中有暗示）。首先，可以使用哈希而不是签名来创建文档之间的链接；因为哈希更简单，计算速度更快。这样的链接被称为哈希指针。其次，不是单独对文档进行线程化处理（如果几乎同时创建多个文档，效率可能会很低），它们可以分为批处理组或区块，每个区块中的文档具有基本相同的时间戳。第三，在每个区块内，文档可以用一个哈希指针的二叉树连接在一起，称为 Merkle 树，而不是一个线性链。顺便提一下，在 Stuart Haber 和 Scott Stornetta 的第一篇论文发表的 6 年之后，即 1991 年，Josh Benaloh 和 Michael de Mare 独立地提出了上述三个方案。

2.2 梅克尔树(Merkle trees)

比特币本质上使用 Josh Benaloh 和 Michael de Mare 在 1991 年和 1997 年提出的数据结构（中本聪大概不了解 Josh Benaloh 和 Michael de Mare 的工作），图 2 以简化的形式表示。当然，在比特币中，交易取代了文档。在 Merkle 树的每一个区块中，叶节点代表交易，每个内部节点由两个指针组成。这个数据结构有两个重要的属性。首先，最新区块的哈希作为摘要。对任何交易（叶节点）的更改都需要将更改一直传播到区块根，以及所有后续区块的根。因此，**如果你知道最新的哈希值，你可以从不可信源下载余下的总账本，并验证是否改变**。类似的观点建立了数据结构的第二个重要属性——也就是说，**某人可以简单有效地向你证明某个特定的交易是否包含在总账本中**。这个用户只须向你发送该交易区块中的少量节点（这是 Merkle 树的特点），以及每个后续块的少量信息。高效地证明交易的包含能力对于性能和可伸缩性是非常需要的。

FIGURE 2: **THE LEDGER DATA STRUCTURE IN LINKED TIMESTAMPING**

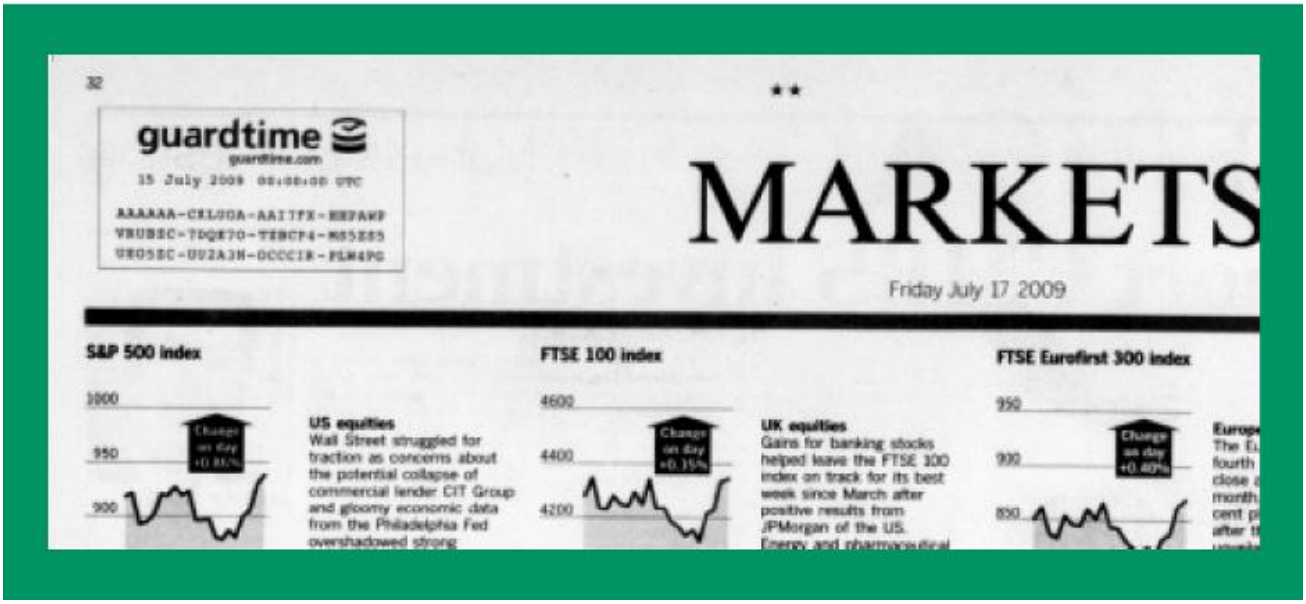


顺便说一下，Merkle 树是由对称密码学的先驱 Ralph Merkle 命名。Ralph Merkle 在 1980 年的论文(文献 33)中提出了这个想法。他的目标应用是生产数字签名证书的公共目录摘要。例如，当一个网站向你提供一个证书时，它也可以提供证书显示在全局目录中的简短证明。只要你知道证书目录中 Merkle 树的根哈希，就可以高效地验证证明。**这个想法在密码标准中是古老的，但它的力量只有在最近才被认可。**它是最近实施的证书透明系统(文献 30)的核心。2015 年的一篇论文提出了 CONIKS，将 Merkle 树应用于端到端加密电子邮件的公钥目录(文献 32)。**对全局状态的部分进行高效的验证是总账本在新的加密货币(cryptocurrency) “以太坊(Ethereum)”中提供的关键功能之一。**

*译注：Ralph Merkle，生于美国，计算机科学家，对于公钥加密技术有重大贡献。后来研究方向转至于纳米科技以及人体冷冻技术。

比特币可能是 Josh Benaloh 和 Michael de Mare 数据结构在真实世界中最著名的应用实例，但它并不是第一个。至少有两家公司 ——从 90 年代中期开始的 Surety，从 2007 年开始的 Guardtime —— 都用到了文档时间戳服务。这些服务都有一个有趣的交集(An interesting twist)是 Bayer, Haber 和 Stornetta(文献 5)提到的一个想法，这个想法是在报纸上以广告的形式定期刊登 Merkle 根。图 3 显示了由 Guardtime 发布的 Merkle 根。

FIGURE 3: **GUARDTIME MERKLE ROOT PUBLISHED IN NEWSPAPER (TOP LEFT)**



2.3 拜占庭容错(Byzantine fault tolerance)

当然，没有中心权威的互联网货币要求更为严格。分布式账本将不可避免地存在分叉，这意味着一些节点会认为 A 区块是最新的块，而其他节点会认为 B 区块是最新的块。这可能是因为攻击者试图破坏总账本的操作；也可能仅仅是因为网络延迟，不同的节点不知道对方的区块，偶尔会几乎同时产生区块。仅仅依靠链式时间戳解决分叉是不够的，这由 Mike 在 1998 年的文章中证明(文献 26)。

一个不同的研究领域——容错分布式计算——已经研究了这个问题，其中包括状态复制（state replication）在内的不同名称。解决这个问题的方法是使一组节点以相同的顺序应用状态转换——通常，精确的顺序无关紧要，只要所有的节点都是一致的。对于数字货币(digital currency)，要复制的状态是一组余额，交易即是状态转换。早期的解决方案，包括由图灵奖得主 Leslie Lamport 于 1989 年(文献 28,29)提出的 Paxos——当通信信道不可靠时，少数节点可能会出现某些“realistic”的故障，例如永远离线或重新启动，收到最初离线时发送的过时消息等——会考虑状态复制。随后发表的大量文献，主要是应对更为复杂（敌对/不利）的环境，以及针对效率的权衡（tradeoff）。

一系列相关工作研究了网络大多可靠的情况（消息以有限延迟传递），但是“故障”的定义被扩展为处理与协议的任何偏离（any deviation）。这种拜占庭式故障包括自然发生的故障以及恶意制造的行为。早在 1982 年(文献 27)，Lamport，Robert Shostak 和 Marshall Pease 发表了一篇论文：《拜占庭将军问题》。之后的 1999 年，Miguel Castro 和 Barbara Liskov 发表了一篇里程碑式的论文引入 PBFT（practical Byzantine fault tolerance）同时容纳了拜占庭故障和不可靠的网络(文献 8)。**与链式时间戳相比，容错有关的文献数量是很多的，包括 Paxos，PBFT 和其他重要协议的数百种变体和优化。**

中本聪在最初的白皮书中没有引用 BFT 的文献或者使用其语言。他使用了一些概念，将协议作为一种共识机制，并以攻击者的形式，以及节点加入和离开网络的方式来考虑故障问题。**这与他明确声明参考了链式时间戳的文献（包括工作量证明，下文讨论）形成鲜明对比。**当被问及关于比特币与拜占庭将军问题（一个需要 BFT 解决的思想实验）的邮件列表讨论时，中本聪声称工作量证明链解决了这个问题(文献 35)。

在接下来的几年中，其他学者从分布式系统的角度研究了中本聪的共识机制——这仍然是一个正在进行的工作。有人表示，比特币的属性是相当弱的(文献 43)；而另外一些人则认为，对于比特币的一致性属性来说(文献 40)，BFT 的观点并不公平。另一种方法是定义已经充分研究的性质变体，并证明比特币满足它们(文献 19)。最近，这些定义大大加强了，以提供一个更为标准的一致性定义，且该定义为消息传递保留更多现实性假设(文献 37)。然而，所有这些工作都假设部分参与节点的行为是“诚实”的（例如，协议兼容），而中本聪则认为，没有必要盲目地假设节点行为是诚实的，因为行为是被激励的。**对中本聪的激励共识机制的全面分析并不适合过去的容错系统模型。**

3 工作量证明(Proof of Work)

几乎所有容错系统都假定系统中的大多数或绝大多数（如超过一半或三分之二）节点都是诚实和可靠的。在一个开放的对等网络中，没有节点的注册，节点可以自由地加入和离开。因此，攻击者可以创建足够多的 Sybils 或 sockpuppet 节点来打破系统的一致性保证。Sybil 攻击是由 John Douceur 在 2002 年正式形式化，并提出借助于密码学基础设施——工作量证明——来化解它。

3.1 起源(The origins)

为了理解工作量证明，我们来看看它的起源。工作量证明概念是由 Cynthia Dwork 和 Moni Naor 于 1992 年首次提出和创建的。他们的目标是阻止垃圾邮件。请注意，垃圾邮件，Sybil 攻击和拒绝服务都是大致类似的问题：与常规用户相比，攻击者通过网络增大其破坏力。工作量证明适用于三方防御。在 Cynthia Dwork 和 Moni Naor 的设计中，**电子邮件收件人只会处理那些附带证明——发件人执行了适量计算工作——的电子邮件，即“工作量证明”。**计算工作量证明在普通计算机上可能需要几秒钟的时间。因此，对普通用户来说不会造成任何困难，但是对于垃圾邮件发送者，在使用等效硬件的条件下，发送一百万封电子邮件则需要几周的时间。

请注意，**工作量证明（也称为难题解决）必须特定于电子邮件以及收件人。**否则，垃圾邮件发送者将能够向同一个收件人发送多个邮件（或者向多个收件人发送相同的邮件），而成本和一对一发送一样。第二个重要的特点是它应该给收件人仅造成最小的计算负担；难题解决方案应该是易于验证的，无论他们计算多么困难。此外，Cynthia Dwork 和 Moni Naor 认为带有后门的功能——这是中心权威机构所知道的一个秘密——可以让权威机构在不做工作量证明的情况下解决问题。一个可能的应用程

序后门是为权威机构开放一个不产生成本发送邮件的邮件列表。Cynthia Dwork 和 Moni Naor 的提案包含三个满足其性质的候选难题，并启动了整个研究领域，我们将再次回到这个主题。

3.2 哈希现金(Hashcash)

一个非常类似的名为 hashcash 的想法是在 1997 年由当时是 cypherpunk 社区的博士后研究员 Adam Back 独立发明的。Cypherpunk 社员是反政府和反中心机构力量的活动家，并致力于通过密码学推动社会和政治变革。Adam Back 是注重实践的人：他首先发布的是 hashcash 软件，五年后的 2002 年才发布 Internet 草案（标准化文件）和论文(文献 4)。

Hashcash 比 Cynthia Dwork 和 Moni Naor 的想法简单得多：它没有后门，也不需要中心权威，它只使用哈希函数而不是数字签名。**Hashcash 基于一个简单的原理：哈希函数在某些实际用途中表现为随机函数，这意味着找到哈希到特定输出的输入的唯一方法是尝试各种输入，直到产生期望的输出为止。而且，找到哈希到任意一组输出的输入的唯一方法是再次逐个尝试对不同的输入进行哈希。**所以，如果让你尝试找到一个输出哈希值以 10 个零开始的输入（二进制），你将不得不尝试大量的输入，你会发现每个输出从 10 个零开始的机会都是（1/2）^10，这意味着你将不得不尝试（2）^10 个输入的顺序，或大约 1000 个哈希计算。

顾名思义，在 hashcash 中，Adam Back 把工作量证明看作一种货币形式。在他的网站上，他把这种货币定位为 David Chaum 的 DigiCash 实现选择之一——一个由银行向用户发放无法追踪的数字现金(digital cash)的系统。他甚至在技术设计上做了一些权衡设计，使其显得更像一种货币。后来，Adam Back 评论认为比特币就是 hashcash 的直接扩展。但是，hashcash 并不是现金，因为它没有防止双重支出（双花）的保护。Hashcash 的令牌不能在对等同伴之间交换。

同时，在学术领域中，研究人员发现，**除了垃圾邮件之外，工作量证明还有很多应用场景**，例如防止拒绝服务攻击(文献 25)，确保网络分析的真实性(文献 17)，密码在线猜测的速率限制(文献 38)等。顺便说一句，工作量证明这个词是由 Markus Jakobsson 和 Ari Juels 在 1999 年撰写的一篇论文中首次提出来的，这篇论文也是到那时为止对这一研究的很好综述(文献 24)。值得注意的是，这些研究人员似乎并不知道到 hashcash，各自独立地朝着基于哈希的工作量证明的方向汇集，这在 Eran Gabber 等人的论文以及 Juels(文献 18)和 Brainard(文献 25)的论文中都有提及（本文中使用的许多术语是在有关论文发表后很长时间才成为标准术语的）。

侧边栏：Sybil-对抗网络(Sybil-resistant networks)

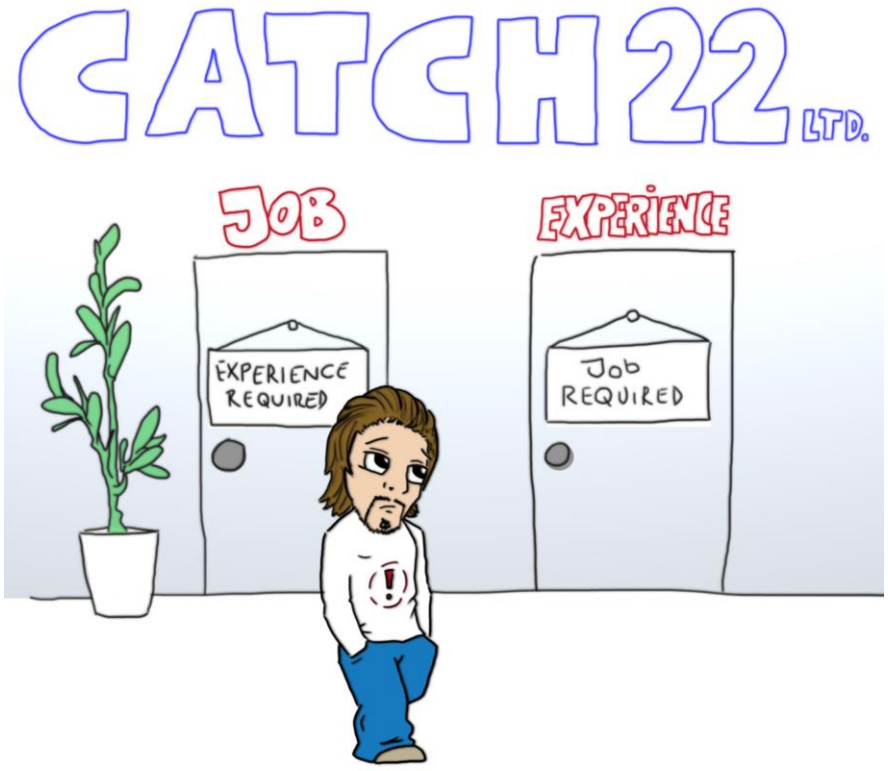
John Douceur 在他关于 Sybil 攻击的论文中提出，所有参与 BFT 协议的节点都需要解决 hashcash 难题。如果一个节点伪装成 N 个身份，将无法及时解决 N 个难题，其伪造的身份将被清除。然而，恶意节点仍然可以获得比只声称单一身份的诚实节点有更多的优势。2005 年发布的后续文章(文献 1)中

提出，诚实的节点应该反过来模仿恶意节点的行为，并声称其计算能力能够承担的尽可能多的虚拟身份。**利用这些虚拟身份执行 BFT 协议，原来的假设“最多只有部分 f 节点故障”可以用“由故障节点控制的总计算能力的份额至多为 f ”来代替。**因此，不再需要验证身份，并且开放的对等网络可以运行 BFT 协议，比特币恰好使用了这个想法，但中本聪提出了进一步的问题：用什么来激励节点执行昂贵的工作量证明计算呢？答案需要进一步的飞跃：数字货币(digital currency)。

3.3 工作量证明和数字现金：双环困境(Proof of work and digital cash: A catch-22)

你可能知道，作为反垃圾邮件措施，**工作量证明没有成功应用于其发源的应用。**一个可能原因是不同设备解决难题的速度有巨大差异。这意味着垃圾邮件发送者可以用小额的投资来定制硬件，就可以将滥发垃圾邮件的速率提高几个数量级。**在经济学中，对生产成本不对称的自然反应是进行贸易——即，工作量证明的交易市场。**但是，这里就是一个**双环困境**（catch-22），因为这将需要一个可以工作的数字货币(digital currency)。事实上，正是因为缺乏这样的货币，导致了工作量证明使用的最大动机不足。这个问题的一个粗暴的解决办法是宣布难题解决方案是现金，正如 hashcash 试图做的那样。

*译注：工作量就是货币，而工作量又需要货币激励，这就是双环困境。



*译注：双环困境，一般指互相抵触之规律或条件所造成的无法脱身的困窘；或者是不合逻辑的或矛盾的问题。例如这就是一个相互矛盾的困窘：没有人想要支持你除非你已经成功了，但是如果没有人支持，你怎么可能成功呢？

在比特币之前的两篇文章中发现了更为清晰的方案来将难题解决作为现金来处理，文章分别描述了 b-money(文献 13)和 bit gold(文献 42)。这些方案提供时间戳服务，用来签署钱的创建（通过工作量证明），并且一旦创建了钱，就可以签署转账。但是，如果服务器或节点之间出现总账本不一致的情况，文章则没有给出明确的解决办法。依靠多数原则来决定似乎是两位作者的文章的隐含之意，但是由于 Sybil 问题，这些机制并不是很安全，除非有一个 gatekeeper 控制网络的引入，或者 Sybil 对抗本身是通过工作量证明来实现的。

4 组合创新(Putting it all together)

通过了解所有这些贡献了比特币设计细节的前辈，你会体会到中本聪真正天才般的创新。在比特币中，难题解决方案不能自我构建成为现金，相反，他们只是用来保护总账本。而工量作证明的解决是由专门的实体称为矿工来完成的（虽然中本聪没有估计到专业挖矿将会成为什么样子）。

矿工们需要不断地相互竞争，寻找下一个难题解决方案。每个矿工都要解决这个难题的一个稍微不同的变体，因此成功的机会与矿工控制的全球采矿能力的一部分成正比。解决难题的矿工贡献了基于链式时间戳的总账本的下一个批次，或者区块（即下一个交易）。通过维护和交换总账本，贡献一个区块的矿工将会获得一个新挖到货币的一份奖励。很有可能的是，如果一名矿工贡献了一个无效的区块或交易，将会被大多数其他贡献下一区块的矿工拒绝，从而使无效区块的奖励失效。这样，由于金钱上的激励，确保矿工们彼此都遵循同样的协议。

比特币巧妙地避免了困扰“工作量即现金(proof-of-work-as-cash)”机制的双重支出（double-spending,双花）问题，因为它避开了难题解决方案本身的价值。**事实上，比特币实现了难题解决方案与经济价值的两次解耦：生产一个区块所需的工作量是一个浮动参数（与全球采矿能力成正比），而且更进一步每个区块发放的比特币数也不是固定的。**区块奖励（也就是新比特币如何挖出）每四年设定一半（在 2017 年，奖励是 12.5 比特币/块，从最初 50 比特币/块减半而来）。比特币包含了一个额外奖励计划——即交易发起者向具有包含该交易的区块的矿工支付交易费用，并且期望由市场决定交易费用和矿工的报酬。

那么，中本聪的天才并不是比特币的任何单个组成部分，而是创造了一种复杂方式——把各种技术融合在一起为整个系统注入生命活力。客观的说，时间戳和拜占庭协议的研究人员没有触及节点激励问题，直到 2005 年，也没有使用工作量证明来消除节点身份问题。反过来说，**hashcash，b-money 和 bit gold 的作者并没有吸收共识/一致性算法的思想来解决双重支出（双花）问题。**在比特币中，一个安全的总账本才能防止双重支出（双花）问题，从而确保货币有价值。有价值的货币才能奖励矿

工，然后保证采矿力量的强度才能保证总账本的安全。如果没有足够采矿力量，一个对手可能会占据全球 50%以上的采矿能力，从而能够比网络的其他部分更快地生成数据块，然后双重支付交易，并有效地重写历史记录，使得整个系统赤字。**因此，比特币是自举的，在总账本，货币，矿工这三个组件之间有一个闭环的依赖关系。中本聪面临的挑战不仅仅在于设计，而是能够说服最初的用户和矿工社区一起，面向未知的时代，纵身一跃：那个时候比萨的价格超过了 10000 比特币，网络的采矿能力还不到今天的万亿分之一。**

侧边栏：智能合约(Smart contracts)

一个智能合约就是把数据放在一个安全的总账本中，并将智能合约扩展到计算。换句话说，它是一个公开指定程序正确执行的共识协议。用户可以调用智能合约程序中的功能，并服从程序指定的任何限制，并且功能代码由矿工串联执行。用户可以信任输出而不必重做计算，并可以编写自己的程序来处理其他程序的输出。通过与加密货币(cryptocurrency)平台相结合，智能合约尤其强大，因为上述程序可以处理资金——拥有，转让，销毁，在某些情况下甚至可以打印。

比特币实现一种限制性编程语言作为智能合约。一个“标准”交易（即将货币从一个地址转账另一个地址的交易）就是用这种语言实现的简短脚本。以太坊(Ethereum)提供更宽容和强大的语言。

智能合约的想法是由 Nick Szabo 在 1994 年提出的(文献 41)，因为可以类比于法律合同（智能合约比法律合同多了自动执行的功能），所以命名为智能合约。**Nick Szabo 早就预见性（这个观点已经被 Karen Levy (文献 31)和 Ed Felten 所批评(文献 16)）的提出了智能合约作为数字现金协议的扩展，并且认识到拜占庭协议和数字签名（等等）可以作为构建模块。**加密货币的成功使智能合约成为现实，对这个话题的研究也开始兴起。例如，编程语言研究人员已经调整了他们的方法和工具，以自动发现智能合同中的错误并写出可校验正确的智能合约。

4.1 公钥即身份(Public keys as identities)

本文基于这样一个理解：一个安全的总账本使创建数字货币(digital currency)更加容易。让我们再回顾这个断言。当 Alice 希望支付 Bob 时，她将交易广播给所有的比特币节点。一个交易只不过是一个字符串：一个声明“Alice 希望支付给 Bob 一些钱”并由 Alice 签名。最终，这个声明被矿工记入总账本，交易即成为现实。请注意，这个过程中并不要求 Bob 以任何方式参与。但是让我们把注意力放在这次交易的缺席者：显然缺席的是 Alice 和 Bob 的身份；相反，交易只包含他们各自的公钥。**这就是比特币的一个重要的概念：公钥是系统中唯一的身份**。交易向公钥传入或者传出价值，这个公约就称为地址。

*译者注：从这个地址概念的引入，对比传统的分布式系统，中本聪的创新很巧妙。

为了能够“说出”一个身份，你必须知道相应的密钥。你可以随时创建一个新的身份——方法是生成一个新的密钥对——而不需要中心机构或注册机构。你不需要申请用户名或通知其他人你已选择了特定的名称——这是去中心化身份管理的概念——比特币没有指定 Alice 如何告诉 Bob 她的化名（Pseudonyms）是什么，这是系统外部的。

与当今大多数其他支付系统截然不同，这些想法相当“古老”，可以追溯到数字现金(digital cash)之父 David Chaum。实际上，David Chaum 也对匿名网络做了开创性的贡献，正是在这个背景下，他发明了这个“数字化名”（ Digital Pseudonyms）这个创意。在他 1981 年的论文“Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms(文献 9)” 他说：“数字化名”是一个公钥，用来验证由相应私钥的匿名持有者是否取得签名。

现在，**只有通过公钥才知道收件人是一个明显的问题：无法将邮件路由到正确的计算机。这导致了 David Chaum 方案的效率极低：匿名交易，不能消除。**与集中支付系统相比，比特币同样效率极低：包含每个交易的总账由系统中的每个节点维护。无论如何，比特币选择了安全而同时选择了低效，从而实现了“免费”的匿名性（即公钥作为身份）。David Chaum 在 1985 年的论文(文献 11)，把这些想法推进了一步，他提出了一种基于普遍“化名”隐私保护电子商务愿景，以及数字现金(digital cash)背后的关键思想——“盲签名(blind signatures)”。

“公钥即身份”的思想也存在于前面讨论过的比特币的先驱文献中：b-money 和 bit gold。但是，大部分工作都建立在 David Chaum 的基础上，而 David Chaum 自己后来的工作（包括电子现金）都偏离了这个想法。Cypherpunk 社区对隐私保护的通讯和商务有浓厚的兴趣，他们拥抱了他们称之为 nyms 的“化名”。但对他们来说，nyms 不仅仅是密码身份（即公钥），而是通常与公钥相关的电子邮件地址。同样，伊恩 • 戈德堡（Ian Goldberg）的论文——后续匿名通讯工作的基础——赞成 David Chaum 的观点，但是认为“化名”nyms 应该是人们容易记住的用证书绑定的绰号。因此，比特币被证明是 David Chaum 思想最成功的例子。

5 区块链(The Blockchain)

到目前为止，本文没有提到区块链。如果你相信炒作，区块链是比特币的主要发明。可能会让你大吃一惊的是，中本聪压根没有提到这个词。实际上，区块链这个技术术语没有标准的技术定义，但是被各方用来指称与比特币和总账本有不同程度相似的系统。

讨论受益于区块链的示例应用程序将有助于澄清该术语的不同用途。首先，考虑一个银行财团之间交易的数据库后端，交易在每天结束时呈网状结构，账户由中央银行结算。这样一个体系有少数的明确的各方，所以中本聪的共识将是矫枉过正的。也不需要区块链上的货币，因为账户是以传统货币计价

的。另一方面，链式时间戳显然是有用的，至少可以确保在网络延迟的情况下实现一致的全局事务排序。状态复制也是有用的：一家银行会知道，其本地数据副本与中央银行用来结算账户的数据是一样的。这使银行摆脱了目前必须执行的代价高昂的协调过程。

其次，考虑一个资产管理应用程序，例如追踪金融证券，房地产或任何其他资产所有权的文档登记册。使用区块链可以提高互操作性，降低进入门槛。我们希望有一个安全的全球文档登记册，最好能让公众参与。这本质上是 20 世纪 90 年代和新千年时代的时间戳服务所提供的。公共区块链提供了一个特别有效的方法来实现这一点（数据本身可能被存储在链外，只有元数据存储在链上）。其他应用程序也受益于时间戳或“公告板”抽象，最显著的是电子投票。

让我们继续资产管理的例子。假设你想通过区块链执行资产交易，而不是仅仅保存交易记录。如果资产本身在区块链以数字形式发行，并且区块链支持智能合约，则可以进行交易。在这种情况下，**智能合约解决了确保只有在资产转移时才进行支付的“公平交换”问题。**更一般地说，智能合约可以对复杂的业务逻辑进行编码，只要所有必要的输入数据（资产，价格等）都在区块链上表示出来。

这种区块链属性与应用程序的映射使我们不仅能够欣赏其潜力，而且还能够使我们注入极为需要的怀疑态度。首先，很多提议的区块链应用，特别是在银行业，并没有使用中本聪的共识机制。相反，他们使用总账本数据结构和拜占庭协议（这些技术，如前文所述，可以追溯到上个世纪 90 年代）。这就暗示了区块链是一种新的革命性技术。相反，围绕区块链的嗡嗡声(buzz)已经帮助银行发起集体行动来部署共享总账本技术，正如“石头汤(stone soup)”的隐喻。比特币也是去中心化总账本工作的一个非常明显的概念证明，并且比特币核心项目提供了一个便利的代码库，可以根据需要进行调整。

*译注：《石头汤》是一本根据法国民间故事改写的作品，但琼·穆特把故事的场景设定在古代中国。三个和尚来到一个饱经苦难的村庄，村民们长年在艰难岁月中煎熬，心肠变得坚硬，不愿接纳任

何人。可是，和尚们用煮石头汤的方法，让村民们不知不觉地付出了很多，明白了分享与幸福在真谛。

琼·穆特，美国图画书作家和画家曾在日本学习，醉心于日本和中国传统文化。创作了很多具有东方哲学智慧的图画书，比如《禅的故事》。

其次，有一个误导性的说法：区块链通常比传统的文档登记更安全。要明白为什么，必须把系统或平台的整体稳定性与终端安全（即用户和设备的安全性）分开。诚然，**区块链的系统性风险可能低于许多中心机构，但区块链的端点安全风险远远高于传统机构相应的风险。**区块链交易几乎是即时的，不可逆转的，而且在公共区块链中，设计为匿名交易。在基于区块链的股票登记中，如果用户（经纪人或代理人）失去对其私人密钥的控制权——只要手机丢失或在计算机上安装了恶意软件——则用户将丧失其资产。比特币黑客、盗窃和诈骗的非凡历史并不会给人们带来多大的信心，据估计，至少有6%的比特币在流通中被盗过一次(文献 39)。

侧边栏：许可区块链(Permissioned blockchains)

虽然这篇文章强调私人和许可区块链并没有使用比特币的大部分创新，但这并不意味着这个领域发生的有趣工作就很少。许可区块链限制谁可以加入网络，写交易或挖矿（区块）。特别是，如果矿工被限制在一个值得信赖的参与者名单，则可以放弃工作量证明，以利于更传统的 BFT 方法。因此，大部分的研究都是 BFT 算法的重生，并且可以提出如下问题：我们可以使用哈希树来简化共识算法吗？如果网络只能以某种方式出现故障呢？

此外，围绕身份和公共密钥基础设施，访问控制以及存储在区块链上的数据的机密性等话题，还有一些重要的考虑因素。这些问题很大程度上不在公共区块链中出现，也没有被传统的 BFT 文献所研究。

最后，还有一项工程性工作是提高区块链的吞吐量，并将其应用于各种业务：比如供应链管理和金融技术。

6 经验总结(Concluding Lessons)

这里描述的历史为从业人员和专业学者提供了丰富（和互补）的经验教训。从业者应该对革命性技术的主张持怀疑态度。如前文所示，比特币中那些引起企业兴奋的大多数想法，例如分布式账本和拜占庭协议，可以追溯到 20 年以上。**认识到你的问题可能不需要任何突破性创新——在研究论文中可以找到那些长期被遗忘的解决方案。**

学术界似乎有相反的问题，至少在这种情形下：抵制激进的，外来的想法。比特币白皮书的很多想法尽管都可以回溯其谱系，但比大多数学术研究更新颖。而且，中本聪不关心学术同行评议，也没有完全把它与学术历史联系起来。因此在数年里，学术界几乎完全忽视比特币。许多学术团体非正式地认为，尽管比特币事实上在实践中运行的很好，但比特币不可能基于过去系统的理论模型和经验的来运行。

我们一再看到，研究文献中的创意可能会逐渐被遗忘或被忽略，特别是如果这些想法是超越它们的时代，甚至在流行的研究领域之外。**从业者和专业学者都应该回顾旧的创意，收集当前系统的见解。比特币的非凡和成功之处不在于它处于任何组件研究的前沿，而在于它整合了许多不相关领域的旧创意。**要做到这一点并不容易，因为它需要弥合不同的术语，假设等，但这是创新的宝贵蓝图。

从业者应该能够识别过度炒作的技术并因此受益。**识别技术炒作有一些指标：难以确定其技术创新；由于企业急于把自己的产品附加到流行趋势上，所以难以确定所谓技术术语的含义；难以确定正在解决的问题；最后，要求技术解决社会问题或者制造经济/政治动荡。**

相反，学术界却难以推销其发明。例如，不幸的是，最初的工作量证明研究人员没有得到比特币的信贷（credit for bitcoin），可能是因为这项工作在学术界以外并不为人所知。在学术界，诸如发布代码和与从业者合作等活动没有得到充分的奖励。事实上，**迄今为止，学术工作量证明的原始分支仍然不承认比特币的存在！**与现实世界接触不仅有助于获得信贷（credit），而且还会减少轮子再造，并且是找到新创意。

7 致谢(Acknowledgements)

The authors are grateful to Adam Back, Andrew Miller, Edward Felten, Harry Kalodner, Ian Goldberg, Ian Grigg, Joseph Bonneau, Malte Möser, Mike Just, Neha Narula, Steven Goldfeder, and Stuart Haber for valuable feedback on a draft.

8 参考文献(References)

1. Aspnes, J., et al. 2005. Exposing computationally challenged Byzantine imposters. Yale University Department of Computer Science; http://cs.yale.edu/publications/techreports/tr1332.pdf.

2. Back, A. 1997. A partial hash collision based postage scheme; http://www.hashcash.org/papers/announce.txt.

3. Back, A. 2001. Hash cash; https://web.archive.org/web/20010614013848/http://cypherspace.org/hashcash/.

4. Back, A. 2002. Hashcash—a denial of service counter measure; http://www.hashcash.org/papers/hashcash.pdf.

5. Bayer, D., Haber, S., Stornetta, W. S. Improving the efficiency and reliability of digital time-stamping. Proceedings of Sequences 1991; https://link.springer.com/chapter/10.1007/978-1-4613-9323-8_24.

6. Benaloh, J., de Mare, M. 1991. Efficient broadcast timestamping; http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.38.9199.

7. Boyle, T. F. 1997. GLT and GLR: Component architecture for general ledgers; https://linas.org/mirrors/www.gldialtone.com/2001.07.14/GLT-GLR.htm.

8. Castro, M., Liskov, B. 1999. Practical Byzantine fault tolerance. Proceedings of the Third Symposium on Operating Systems Design and Implementation; http://pmg.csail.mit.edu/papers/osdi99.pdf.

9. Chaum, D. 1981. Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM 24(2): 84-90; https://dl.acm.org/citation.cfm?id=358563.

10. Chaum, D. 1983. Blind signatures for untraceable payments. Advances in Cryptology: 199-203.

11. Chaum, D. 1985. Security without identification: transaction systems to make Big Brother obsolete. Communications of the ACM 28(10): 1030-1044; https://dl.acm.org/citation.cfm?id=4373.

12. Chaum, D., et al. 1988. Untraceable electronic cash. Advances in Cryptology: 319-327; https://dl.acm.org/citation.cfm?id=88969.

13. Dai, W. 1998; http://www.weidai.com/bmoney.txt.

14. Douceur, J. R. 2002. The Sybil attack; https://dl.acm.org/citation.cfm?id=687813.

15. Dwork, C., Naor, M. 1992. Pricing via processing or combatting junk mail; https://dl.acm.org/citation.cfm?id=705669.

16. Felten, E. 2017. Smart contracts: neither smart nor contracts? Freedom to Tinker; https://freedom-to-tinker.com/2017/02/20/smart-contracts-neither-smart-not-contracts/.

17. Franklin, M. K., Malkhi, D. 1997. Auditable metering and lightweight security; http://www.hashcash.org/papers/auditable-metering.pdf.

18. Gabber, E., et al. 1998. Curbing Junk E-Mail via Secure Classiffication. http://www.hashcash.org/papers/secure-classification.pdf.

19. Garay, J. A., et al. 2015. The bitcoin backbone protocol: analysis and applications. Advances in Cryptology: 281-310; https://eprint.iacr.org/2014/765.pdf.

20. Goldberg, I. 2000. A pseudonymous communications infrastructure for the Internet. Ph.D. dissertation, University of California Berkeley; http://moria.freehaven.net/anonbib/cache/ian-thesis.pdf.

21. Grigg, I. 2005. Triple entry accounting; http://iang.org/papers/triple_entry.html.

22. Haber, S., Stornetta, W. S. 1991. How to timestamp a digital document. Journal of Cryptology 3(2): 99-111; https://link.springer.com/chapter/10.1007/3-540-38424-3_32.

23. Haber, S., Stornetta, W. S. 1997. Secure names for bit-strings. In Proceedings of the 4th ACM Conference on Computer and Communications Security: 28-35; http://dl.acm.org/citation.cfm?id=266430.

24. Jakobsson, M., Juels, A. 1999. Proofs of work and bread pudding protocols; http://www.hashcash.org/papers/bread-pudding.pdf.

25. Juels, A., Brainard, J. 1999. Client puzzles: a cryptographic countermeasure against connection completion attacks. Proceedings of Networks and Distributed Security Systems: 151-165; https://www.isoc.org/isoc/conferences/ndss/99/proceedings/papers/juels.pdf.

26. Just, M. 1998. Some timestamping protocol failures; http://www.isoc.org/isoc/conferences/ndss/98/just.pdf.

27. Lamport, L., et al. 1982. The Byzantine Generals Problem. ACM Transactions on Programming Languages and Systems 4(3): 382-401; https://dl.acm.org/citation.cfm?id=357176 .

28. Lamport, L. 1989. The part-time parliament. Digital Equipment Corporation; https://computerarchive.org/files/mirror/www.bitsavers.org/pdf/dec/tech_reports/SRC-RR-49.pdf.

29. Lamport, L. 2001. Paxos made simple; http://lamport.azurewebsites.net/pubs/paxos-simple.pdf.

30. Laurie, B. 2014. Certificate Transparency. acmqueue 12(8); https://queue.acm.org/detail.cfm?id=2668154.

31. Levy, K. E. C. 2017. Book-smart, not street-smart: blockchain-based smart contracts and the social workings of law. Engaging Science, Technology, and Society 3: 1-15; http://estsjournal.org/article/view/107.

32. Melara, M., et al. 2015. CONIKS: bringing key transparency to end users. Proceedings of the 24th Usenix Security Symposium; https://www.usenix.org/system/files/conference/usenixsecurity15/sec15-paper-melara.pdf.

33. Merkle, R. C. 1980. Protocols for public key cryptosystems. IEEE Symposium on Security and Privacy; http://www.merkle.com/papers/Protocols.pdf.

34. Nakamoto, S. 2008. Bitcoin: a peer-to-peer electronic cash system; https://bitcoin.org/bitcoin.pdf.

35. Nakamoto, S. 2008. Re: Bitcoin P2P e-cash paper; http://satoshi.nakamotoinstitute.org/emails/cryptography/11/.

36. Narayanan, A., et al. 2016. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Princeton University Press; http://bitcoinbook.cs.princeton.edu/.

37. Pass, R., et al. 2017. Analysis of the blockchain protocol in asynchronous networks. Annual International Conference on the Theory and Applications of Cryptographic Techniques; https://link.springer.com/chapter/10.1007/978-3-319-56614-6_22.

38. Pinkas, B., Sander, T. 2002. Securing passwords against dictionary attacks. Proceedings of the Ninth ACM Conference on Computer and Communications Security: 161-170; https://dl.acm.org/citation.cfm?id=586133.

39. Reuters. 2014. Mind your wallet: why the underworld loves bitcoin; http://www.cnbc.com/2014/03/14/mind-your-wallet-why-the-underworld-loves-bitcoin.html.

40. Sirer, E. G. 2016. Bitcoin guarantees strong, not eventual, consistency. Hacking, Distributed; http://hackingdistributed.com/2016/03/01/bitcoin-guarantees-strong-not-eventual-consistency/.

41. Szabo, N. 1994. Smart contracts; http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html.

42. Szabo, N. 2008. Bit gold. Unenumerated; https://unenumerated.blogspot.com/2005/12/bit-gold.html.

43. Wattenhofer, R. 2016. The Science of the Blockchain. Inverted Forest Publishing.

44. Rivest, R. L., Shamir, A. 1996. PayWord and MicroMint: Two simple micropayment schemes. International Workshop on Security Protocols.