公证通Factom白皮书：利用区块链真实地记录一切

[**顽石**](http://www.8btc.com/author/1843)**2015-03-30 14:51 发布在**[**竞争币**](http://www.8btc.com/digital-currency)[**18**](http://www.8btc.com/factombaipishu#comment)**17848**

**公证通Factom白皮书 1.0版**

**在区块链上建立不可更改的审计公证业务流程**

摘要：

Factom利用比特币的区块链技术来革新商业社会和政府部门的数据管理和数据记录方式。

利用区块链技术帮助各种各样应用程序的开发，包括审计系统，医疗信息记录，供应链管理，投票系统，财产契据，法律应用，金融系统等。开发者能够创造新的应用程序，并把数据保存在区块链上面，同时不用受到直接把数据写入比特币区块链的各种限制：例如写入的数据速度，成本，大小等限制。

Factom维护了一个永久不可更改的、基于时间戳记录的、区块链数据网络。大大减少了进行独立审计、管理真实记录、遵守政府监管条例的成本和难度。  
商业社会和政府部门可以利用Factom简化数据记录的管理，记录商业活动，并解决数据记录安全性和符合监管的问题。

Factom正在革新整个世界对数据的记录方式，并利用比特币区块链技术来保护您的数据安全。

一、概要

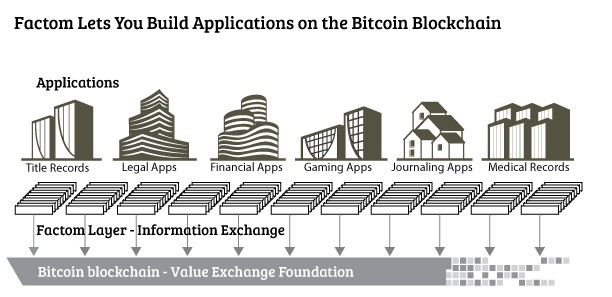
“诚信是具备颠覆性的” – 保罗·斯诺

 在当今的全球经济中，信任是稀有的。这种信任的缺乏，造成了大量资源的投入来进行审计和记录核查，从而降低了全球效率，投资回报率和经济繁荣。此外，如2010年美国的次贷危机等事件表明，目前的审计和核查流程是非常不准确和低效的，容易造成问题。Factom提供了世界上首个准确的，可核查的和不可更改的审计公证流程和方法，从此我们不再需要盲目信任。

 在过去，记录靠手工完成，数据的保护，同步更新和真实性的验证都非常困难。其中的一部分流程被计算机自动化以后，并没有使这些事情变得容易，反而是变得更加困难，因为电脑记录是如此容易被更改。权限和可靠性被四分五裂的分割到无数的独立系统中。

 区块链提供了一个分布式的机制进行数据锁定，使数据可以被核查和独立审计。比特币区块链是现存最值得信赖的不可变的数据存储，但这往往只限于比特币交易上。 Factom能使企业获得最新的区块链技术，而又无需在采用虚拟货币的问题上陷入麻烦。

 在本文中，我们将介绍Factom是如何创建一个分布式的，自治的协议，并把比特币区块链从Bitcoin网络中剥离。我们将讨论客户端自定义的记录链（Chains of Entries），客户端如何验证记录条目，分布式共识算法， 基于区块链锚定的安全机制等。

[](http://7fvhfe.com1.z0.glb.clouddn.com/%2540%252F%2F/wp-content/uploads/2015/03/111.jpg)

二、Factom设计目标

用Factom创建更快，更便宜，无膨胀的方式来开发区块链应用

 中本聪推出的比特币区块链彻底改变了交易记录的方式。在这以前从未存在过永久的，分散的，和无需信任的分类账。开发商纷纷开始开发建立在这个分类帐之上的应用程序。不幸的是，由于比特币一些初始的设计权衡， 他们已经碰到了几个核心的约束和问题：

 1）**速度** – 因为比特币的分布式设计和工作量证明的共识方法，工作量难度会被调整到以保持大致10分钟确认时间。对于希望更大的安全性的应用中，多个确认可能是必需的。一个常见的要求是要等待6确认，这可能会导致等待时间超过一个小时。

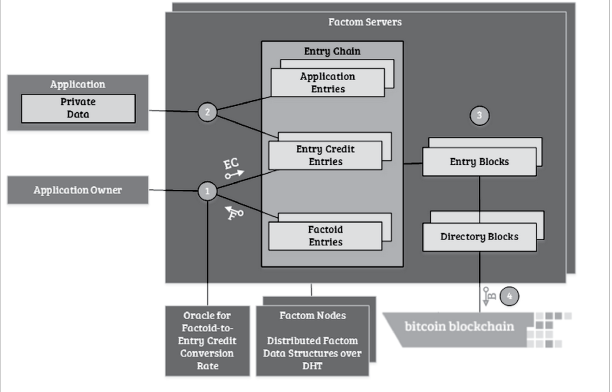
 2）**成本** – 默认的交易成本大约是0.01 mBTC（约0.003美元美元于2014年11月）。 BTC的交易价格一直不停的波动。如果BTC的价格上涨，则交易成本也上去了。这对需要管理非常大的数字交易数据的应用程序来讲， 是一个严重的成本障碍。此外，还有许多因素，如对区块大小的限制和奖励减半等，会导致交易费用增加。

3）**区块链膨胀** – 比特币区块大小限制目前是1MB，交易流量的上限是每秒7交易。任何应用程序想要使用区块链写入和存储信息都将会增加流量。这个问题已致使各方寻求增加区块大小限制。

 Factom是旨在解决这三大核心约束的协议。 Factom协议为应用程序提供的功能和特性超越了虚拟货币。 Factom构建了一个标准的，有效的，安全的基础，这将使应用程序的运行速度更快，更便宜，并且不会造成区块链膨胀。

三、Factom生态系统

Factom生态系统的几个主要部件，描述如下：

[](http://7fvhfe.com1.z0.glb.clouddn.com/%2540%252F%2F/wp-content/uploads/2015/03/121.png)

一旦系统建立后，Factoids（即Factom币）和用户帐户会被激活，用户将会交易Factom币。Factom和比特币具有以下主要的互动：

 1.应用程序用户使用Factom币（Factoids）购买数据条目信用（Entry Credit）

2.应用程序记录数据条目

3. Factom服务器创建条目区块和目录区块

4. Factom将目录区块的哈希值锚定到比特币区块链

*这些数据条目和区块链互动的部分请细看下面的几个章节。*

**安全性和证明**

**Factom如何保障条目的安全记录？**

 Factom把比特币的功能拓展到其货币属性之外的事件记录的功能。 Factom设有最小的规则集，用于永久记录数据条目。 Factom让客户的应用程序来执行大多数的数据验证任务。唯一Factom强制实施的验证是那些通过协议要求交易Factoids，购买条目信用，并确保条目正确付款和记录。

*Factom具有关于激励运行网络和内部一致性的一些规则，但它不能检查用户记录的信息本身的真实性和有效性。*

比特币交易限制被限制为从一组输入值集合映射到到一组输出值集合。只要满足输入值条件的输入值集合（通常需要一定的签名）就可以保证系统输入的有效性。这是可以实现自动化一个验证过程，所以可以使审计过程是容易的。例如，为了记录的房地产的产权转让，Factom可以只记录转让发生那一刻的过程。现实世界中，房地产产权的转让的规则是非常复杂的。 例如，一个地方管辖机构可能对房地产购买者有不同的和特殊的要求，如果买家是外国人，农民，或兼职居民，则购买限制条件是不一样的。房地产的地段位置，房屋价格，或建筑类别等不同属性，也可以使房地产被归到不同的类别中。每个类别都可以有自己的规则来反映智能合约的验证和执行过程。在这种情况下，单独一个加密签名不足以充分验证所有权转移的有效性。 此时Factom更多是用来记录房地产所有权转移和交易的发生，而不是验证房地产所有权转移是否有效。

比特币矿工主要执行两个任务。首先，他们解决双重花费的问题。看到两次花同样的一笔资金的相互冲突的交易发生时，他们选择哪一个是可以受理的和接受的。矿工（连同其他全结点）的第二个任务是进行交易验证和审核。由于比特币矿工只受理有效的交易，一个被包含在区块的交易是可以被假定为已经被审核。轻型客户端并不需要知道比特币的完整历史来看接受的资金是否已经被花掉了。 （见SPV客户端）。

**Factom服务器和审核服务器如何验证数据条目？**

 Factom把比特币矿工做的事情分为两个任务：**按照顺序记录条目和审计条目的有效性**。

**1 – Factom服务器接受数据条目，并将它们装入到不同的区块，并修复条目的顺序。** 10分钟后，该条目的顺序通过插入到比特币区块链的一个锚定而变得不可逆转。 Factom通过对10分钟内收集的数据创建哈希值，然后把这个哈希值记录到比特币的区块链来实现这个功能。

**2 – 条目的审计是一个独立的过程，可以依靠信任第三方或不依靠信任第三方来完成**。审计是至关重要的，因为条目在被包括在Factom数据集之前， Factom是无法验证它们的真伪的。

 以信任为基础的审计，轻型客户端可以信任他们选择一个称职的审计师。一个项目被输入到系统中以后，审计人员将可以验证输入是否有效。审计师将提交自己的加密签名条目。签名会显示该条目通过了所有审计师认为需要做的的检查。审计要求的条件实际上可能是一个Factom链的一部分。以之前的房地产为例，审计师会仔细检查财产转移是否符合当地标准。审核员将公开证明，财产转移是有效的。

 不依靠信任的审计系统将类似于比特币网络。如果一个系统的有效性像比特币网络的数学定义一样完美，那么它可以实现程序化的审计过程。如果用于转移的规则能够由计算机进行审核，则应用程序可以下载有关的数据，并进行自我审计和审核过程。该应用程序可以通过下载数据条目，验证数据条目，并决定条目是否有效，从而使该应用程序建立起对系统的感知。

 Mastercoin, Counterparty, 和Colored Coin也有类似的信任模型。这些都是基于客户端的验证协议，这意味着交易被嵌入到比特币的区块链上。比特币矿工不审核其有效性；因此，基于这些协议的无效的交易如果被故意设计伪装成有效的交易也可以嵌入到比特币的区块链上。支持这些协议之一的客户端通过对区块链扫描并寻找潜在的交易，检查它们的有效性，并对控制这些数字资产的地方建立说明和解释（通常是一个比特币地址）。在这些协议中，都是由客户端来执行自我审计。

要把这些由客户端验证的协议搬到Factom上，将会只是一个如何定义协议中的交易并建立一个链来保存交易的问题。和比特币相比，交易协议在Factom下不会有太大的不同。不同之处在于信息在Factom上很容易表达，而不必以某种特殊方式对其进行编码并嵌入比特币的交易信息中。

**证明否定**

 比特币，土地登记，以及许多其他系统需要解决的一个根本性的问题：证明否定。他们证明了某个“东西”已经被转移到某个人，并证明这个“东西”还没有被转移给其他人。在无界系统里，否定的证明是不可能的，而在一个有界系统里它是很有可能的。 加密货币通过限制交易数据可以存在的地方来解决这个问题。比特币交易只能在比特币区块链里找到。如果某个交易没在比特币区块链里被找到，那它在比特币协议下就不存在，因此，该比特币就尚未被发送两次（双花）。

某些土地所有权记录系统是相似的。假设在一个系统里，土地转让要在政府登记，而且法律制度规定，未记录的转让是无效的（sans litigation）。如果一个人要检查某个产权是否明确（即，没有其他人声称这片土地所有权），答案就在政府登记处。利用政府记录可以证明为否定，土地不被第三方拥有。如果产权登记不是必须的，政府的注册表只能证明什么已被登记。私人转让很可能存在，注册表也就不能代表全部的转让情况。

 在上述两种情况下，否定可以在某个环境中得到证明。以万事达币（Mastercoin）为例，这个证明是强而有力的。而土地登记，则仅限于被注册的环境下，可能遭到一定的质疑和挑战。现实世界是复杂的。Factom的设计，针对的不只是精确的数字资产，而且是物质的世界中复杂的现实情况。

 在Factom中，数据分类有层次结构。 Factom只在链中记录数据条目；诸多用户定义的链在Factom执行的协议中没有互相依赖关系。这不同于比特币，每一笔比特币交易是都存在潜在的双重支付，因此它必须被验证。和把所有数据合并在一起成一个总账相比，Factom通过把条目放到多个链当中，可以让应用程序在较小的空间内搜索数据。

 如果Factom要用来管理土地转让，使用某个链来记录的应用程序可以安全地忽略在其他链上的条目，比如那些本来用于保安摄像机的记录的链就不与需要更新。如果政府法庭判决需要变更土地转让记录，那么和其相关的链将被更新，以反映上述判决的结果。但更改的历史不会丢失，并且如果这样的土地产权变更的更改从法律或其他角度来讲无效的话，它的记录的内容和顺序在Factom上都不能被更改或隐藏。

 尼克·萨博（Nick Szabo）在关于“房产俱乐部”的论述中，与本系统有许多重叠。以下是他在论文“安全产权与所有者权限”中的一个要点：

*虽然暴徒还可以通过暴力攫取物质资产，但延续存在的正确的所有权记录将是留在盗用者眼中的眼中钉。*

**应用程序如何验证Factom链？**

Factom不验证条目；条目是由用户客户端和应用程序来验证的。只要应用程序了解并熟知该条链应遵循的规则，那么无效条目的存在也不会引起不合理的干扰。在链里的不遵守规则的条目可以被应用程序忽略。

用户可以给他们的链设置任意规则，并且可以使用任意方法给这条链的用户传达这条链的规则。在这条链中的第一个数据条目可以被设计成这条链的一组规则，或某个审计程序的哈希值，等等。这些规则可以被应用程序理解，并通过运行Factom，在网络中忽略掉不符合这些规则的客户端应用程序。

 强制执行的序列可以在每条链的规则中指定。不符合这种顺序规定的条目将被应用程序拒绝。然而，那些被规则或审计程序拒绝的条目仍可能被Factom记录在这条链上。这种链的用户将需要运行审计程序，来验证这种序列。Factom服务器将不会使用审计程序来验证此类规则。

上面所述的应用程序验证（与用户自定义链相结合）提供了许多优点：

**1.应用程序可以在Factom上录入任何应用程序可以理解的条目**。比如，验证帐户报表的哈希列表，可以像资产的交易一样简单方便的被记录下来。

**2.规则的执行是非常高效的**。分布式网络必须执行验证规则，然后要求所有节点做所有验证。客户端验证只需要对这些规则关心的系统来执行验证。 Factom允许链使用任何一种语言来设计定义规则，可以使用任何外部数据，并选择在任何平台上运行。一个应用程序所做的决定，对另外一个应用程序不会产生任何影响。

**3. Factom服务器对记录下来的条目的内容没有多少认知。**我们用一个承诺方案来限制对条目的了解：在服务器承诺记录条目之后再透露该条目的内容。这使得Factom的记录条目的角色很简单，也使得各个服务器的工作流程很公开。 Factom服务器接受来自节点网络的信息，它们的决定和行为总是一目了然的。如不履行规则，就会被Factom自己的审核和来自Factom网络外面的审核所发现。所以，一个Factom服务器是否履行记录条目的责任就可以很容易被第三方核实;Factom不可能掩盖潜在的错误行为。

**4.写入记录的速度可以非常快**，因为由Factom服务器把所需检查的次数降至最低。

**5.针对Factom某个链的证明不需要管其它任何链的内容**。用户只需拥有他们使用的Factom的部分数据，可以忽略其余部分。

**Factom联合服务器如何管理链？**

究其核心，Factom是用一种去中心化的方式来收集，打包，安全保护数据，并把数据锚定到比特币的区块链上。 Factom以联邦服务器的网络来实现这个目标。这些服务器不断变换在系统中所承担的责任，永远不会只有一个服务器在控制整个系统，每个服务器都只是系统中的一部分。Factom的服务器每一分钟变换一次角色，没有服务器会永久控制系统的任何一部分。

在开始创建一个目录区块的时候，每个联邦服务器需要对某一部分的用户链负责。过程是这样的：

 1.所有服务器重设其进程列表（Process List）为空。

2.用户通与其条目信用的积分（Entry Credit）相关的公钥提交付款

3.根据用于支付的公钥，轮值服务器接受该付款。

4.该服务器向网络广播该支付被接受。

5.用户看到支付被接受， 然后提交条目。

6.根据条目的ChainID，其中一台服务器把条目加入其进程列表，并添加进入到相应链的区块中（如果这是该链的第一个条目, 那就创建这个新链）。

7.服务器对网络广播该条目的确认，内容含有条目在进程列表中的位置（Index)，条目的哈希值（链接到条目付款），以及最新进程列表的哈希值。

8.所有其他服务器更新该服务器的进程列表，验证该列表，并更新该链的区块。

9.只要用户可以验证到相关的进程列表中包含自己的提交的数据条目，那么他们就可以有相当的信心相信它会被成功地被录入到Factom上。

10.在一分钟结束时，所有服务器确认进程列表高度，揭示一个确定性的秘密数值（该值为一个反向哈希值，即一条较长的，连续的区块链哈希值的原像值），还有被处理区块的一系列哈希值（将与进程列表中的最后一项相匹配）。

11.那一分钟的目录区块（Directory Block）是由所有服务器中定义的所有条目区块（Entry Block）组合到一起建造而生成的。因此，每个服务器都拥有所有的条目区块（Entry Block），所有的目录区块（Directory Block），和所有条目（all Entries）。

12.使用反向哈希值的集合来创造一个种子，为下一轮的ChainIDs重新分配服务器。

13.在完成10个目录区块后，请执行以下操作：

a 对最后一分钟的条目块创建梅克尔根（Merkle Root），按ChainID排序。

b 创建最后一分钟的目录区块，并计算其梅克尔根（Merkle Root）。

c 用10个目录区块的梅克尔根（Merkle Root）创建一个锚定

d用服务器的反向哈希值集合来创建一个种子，再用其选择下一个服务器来把锚定写到比特币区块链上。

14.重复。 （又从第1部开始循环）

在一分钟里，联合服务器为其所负责的链建立进程列表，以及构建这些链的条目区块，这些将用于在一分钟结束时创建目录区块。进程列表很重要，是服务器用来向网络发布其对条目的处理的决定。

       联合服务器每四个小时重新排名。排名由用户投票决定，用户必须在Factom链上登记。登记信息包含任意数量的签名公共地址条目。一个用户的投票权重是由他们的个人资料的公共地址确定。计算一个公共地址投票权重总和的函数是：

●在过去六个月中购买的积分加权（当月乘6，上月乘5，以此类推）

●在过去六个月中使用的条目数加权（当月乘6，上月乘5，以此类推）

当我们说有n个服务器运行，排名前n个服务器是联合服务器，而另有n个为审计服务器。所有服务器都基于票数排名。n值最初指定为16，但这个数目是供社区内讨论，可以基于交易量浮动。

所有服务器必须在每一个心跳周期播出心跳条目。 （一个条目确认时间可作为一次心跳）如果服务器在超时了还没有收到心跳或条目确认，服务器就会广播服务器故障消息（SFM）。如果关于某个联合服务器的SFM数目超过一半，该联合服务器就会被认为是“故障”，并降格为审计服务器，被最高排名的审核服务器接管。升级的服务器将做完当前4小时任期。之后服务器重新排序，但发生故障的服务器必须再等另一个4小时任期。

联合服务器的多数可以在设置链上修改心跳周期和超时规定。参照比特币的传播时间，心跳应该为4秒，超时时间为8秒。

*更多Factom共识机制的细节，以及我们正在开发的算法，可在“Factom共识机制”的文档中找到。*

四、Factom系统概述

**Factom由一组分层数据结构所构成**

Factom由分层结构的区块(Block)组成，根部是目录区块(Directory Block)。这些区块(Block)构成了一个微型链, 链上存储着压缩过的引用(reference). 为了避免数据规模过大，目录块(Directory Block)中的引用(refrence)只是Entry Block 和 ChainID 的哈希值。这些Entry Block包含了引用(refrence). 这些引用(reference)指向了特定时间段内所有Entry. Entry Block也是微链的一部分。在Factom系统里大部分的数据存储在叶子节点上，也就是那些Entry。这些分层数据结构由比特币的算力维护。它们可以被概念化为不同的层。

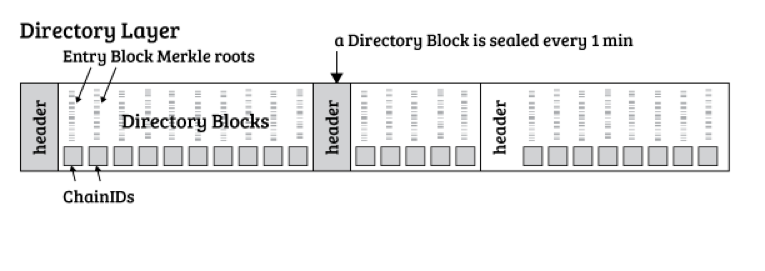
 Factom系统里不同的层和概念：

 1）目录层(Directory Layer) – 负责管理Entry Block的梅克尔根(Merkle root)

 2）Entry Block Layer – 组织指向Entry的引用(Reference)

 3）Entry – 包含应用程序的原始数据或私人数据的哈希值

 4）链(Chain) – 属于应用程序的一组Entry

**目录层(Directory Layer) ：如何管理Entry Block的梅克尔根(Merkle root)**  
[](http://7fvhfe.com1.z0.glb.clouddn.com/%2540%252F%2F/wp-content/uploads/2015/03/1211.png)

目录层(Directory Layer)是Factom分层结构中的的第一级。它定义了哪些 Entry ChainID被更新过, 更新发生在那个目录块(Directory Block)负责的时间段。 （ChainID用于识别用户的Entry属于那个Chain; ChainID的生成将在后面讨论）。目录层(Directory Layer)含有ChainID和对应链块里Entry Block的梅克尔根(Merkel root)。

在目录块(Directory Block)中说引用的每个Entry Block占用64个字节（分别是两个32字节的哈希值，ChainID和Entry Block的梅克尔根）。包含一百万个这样Entry Block的一个目录块(Directory Block)的大小约是64 MB。如果平均每个Entry Block有5个Entry，64 MB的目录块(Directory Block)将管理者500万不同的Entry。

应用程序通过目录块(Directory Block)可以定位到特定的Entry Block, 无需下载所有的Entry Block。一个单独的应用只会对一部分ChainID有兴趣。这极大地减小了使用Factom系统时的带宽需要。例如，一个监控房地产转让的应用程序可以放心地忽略摄像机安全日志。

Factom服务器收集Etnry Block的梅克尔根(Merkle Root), 然后把它们打包到一个目录块(Directory Block)。十个连续目录块(Directory Block)再算出一个梅克尔根(Merkle Root)，这个梅克尔根(Merkle Root)会被记录到比特币链块。这给比特币链块带来最小的负担, 确又足以报账Factom数据的安全. 把梅克尔根(Merkle Root)写入比特币链块的过程被称为Anchor(锚定)。详情参见“附录：Timestamping into Bitcoin (在比特币上加盖时间戳)”。

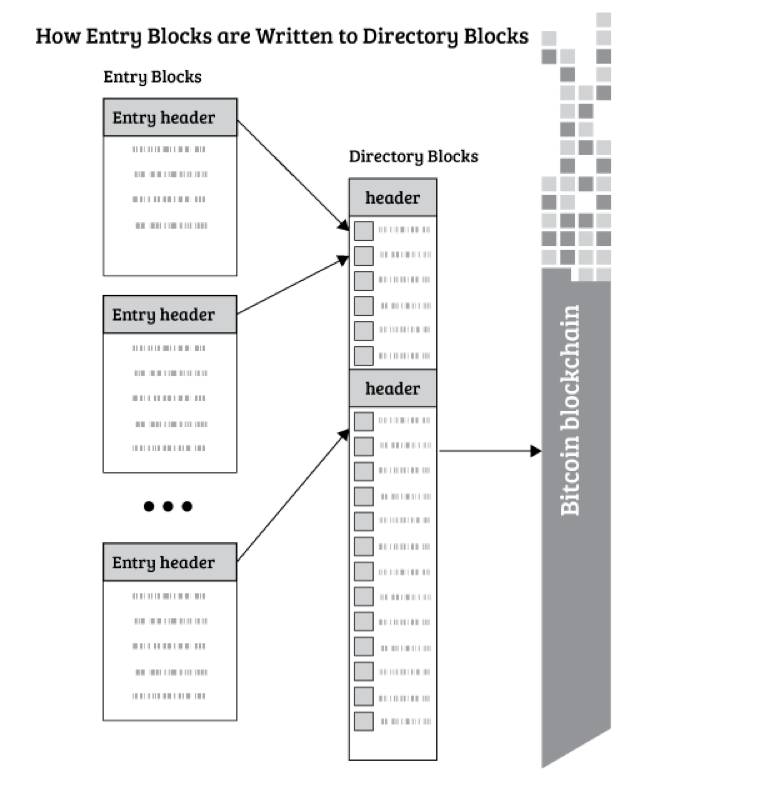
Factom系统里, 目录块(Directory Block)的数据读写需要最多的带宽和存储。用户在建立了自己的Chain以后, 需要保存这个时间点以后所有的目录块(Directory Block)。

创建一个链(Chain)和对链(Chain)的第一次更新会增加目录块(Directory Block)的大小。为了避免目录块(Directory Block)过度膨胀, 把数据写入目录块(Directory Block)所需付的Entry Credit会高于把数据写入 Entry Block.

**记录区块层(Entry Block Layer)：记录区块层如何管理哈希值和数据？**

记录区块(Entry Block)是这个系统的第二分层。个体应用将会关注各种各样的ChainID. 在寻找记录的应用会需要记录区块, 可以从一个ChainID搜索到所有可能相关的记录。

每一个记录区块(Directory Block)内都为每个有更新的ChainID记录下一个Entry Block. Entry Block包含着Entry的哈希值。记录的哈希值同时证明了数据的存在和在分布式散列表（DHT）网络中找到记录的钥匙。（*详细信息请查看“Factom点对点网络”章节*）。

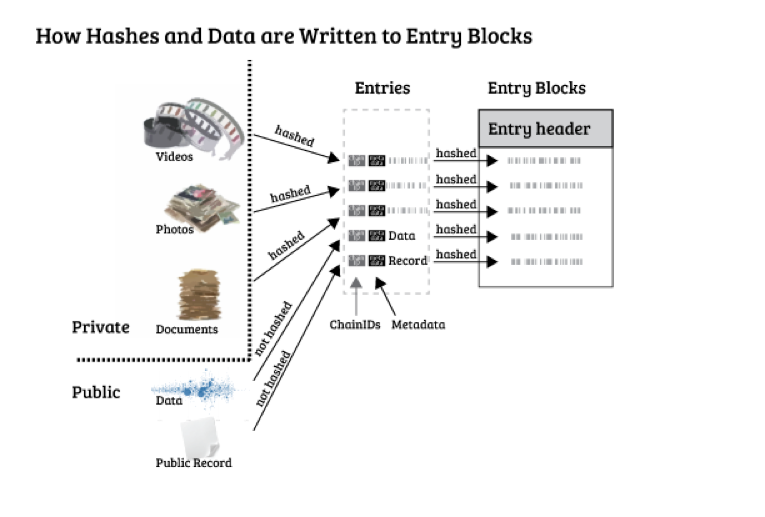
[](http://7fvhfe.com1.z0.glb.clouddn.com/%2540%252F%2F/wp-content/uploads/2015/03/131.png)

记录区块(Entry Block)包含了和一个链ID有关的全部记录(Entry)。如果某个记录(Entry)不是关联到某个记录区块(Entry Block)的话，那么可以认为这个记录(Entry)并不存在。这样的设计能让应用程序很容易的证伪, 方便的识别哪些记录(Entry)是真实可靠的.

记录区块(Entry Block)并不包含记录(Entry)。比起所有数据都被集合起来放入区块中，这种方式会让记录区块(Entry Block)的体积更小。把记录从记录区块中分离出来，也会让数据更加容易审计. 审计员可以在一个单独的链中发布记录，用来批准或拒绝一个普通链中的记录。审计员可以添加拒绝的理由在Entry中。如果一个应用程序信任这个审计员，那么就可以直接采用这个审计员对记录的决定(批准或拒绝), 不用再去重新审核一次记录(Entry)。然后这个应用就可以只下载那些已经被审计通过的记录(Entry)。多个审计员可以引用相同的记录，单个记录只会存在于分布式散列表(Distributed Hash Table, DHT)中一次。这些记录的容量应该会比刚开始的32字节大很多。忽略列表不需要让应用知道已经被忽略的完整对象。

一个记录会详述一宗土地转让的细节，会根据此类交易的类型决定记录在链的什么位置。一个或多个审计员会在他们自己的链上引用这条土地转让的哈希值记录，并添加加密签名来表明此条记录有效或无效。土地转让的文件只需要存储一次，之后就会在多条不同的链上可以被引用。

**记录(Etnry)：记录是如何被创建的？**

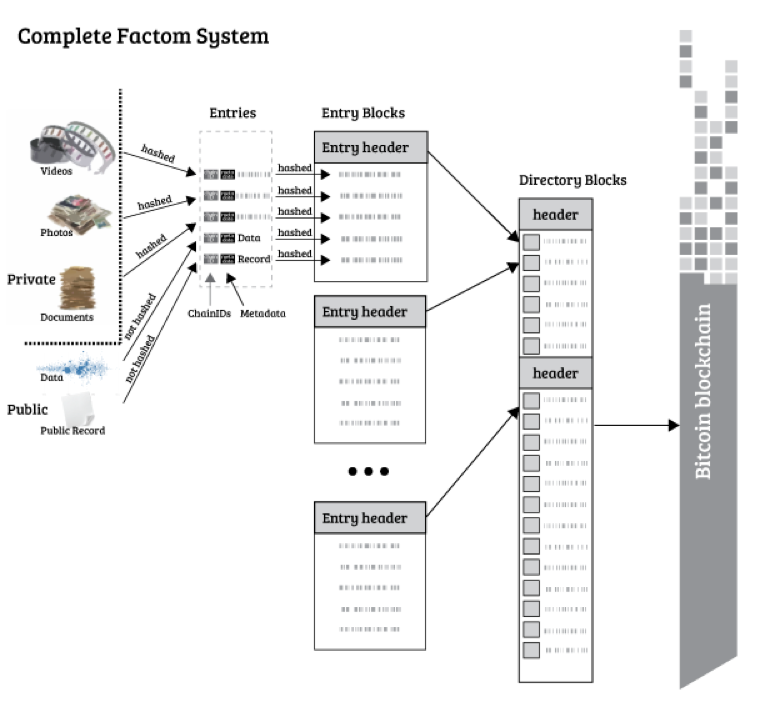
[](http://7fvhfe.com1.z0.glb.clouddn.com/%2540%252F%2F/wp-content/uploads/2015/03/122.png)

记录是被用户创建并提交到Factom的。通过散列和编码信息，用户可以确保记录的隐私性。如果编码或隐藏数据是不必要的话，那么记录可以替换成为纯文本。通过记录一份文档的一段哈希值，Factom可以提供基本的发布证明(proof of publication)。稍后, 人们可以生成文档的哈希值, 并和之前链块记录的哈希值进行比对, 来判断文档是否是当初发布的那个版本。

在数据的处理可以有很大的灵活性。可以出现类似超链接的东西。数据还可以更庞大，但不能过于庞大. 数据越大需要付的费用也越多。这和比特币比较相似。超过100兆字节的比特币转账数据是可能发生的，但需要支付更多的转账费用。Factom可以处理比比特币网络里大得多的数据。由于比特币的完整节点需扫描完整的区块链数据，所以区块链体积不能太大。在Factom中完整节点只需要扫描最高级的目录区块(Directory Block), 并不需要扫描全部的链块数据。如果人不对链数据感兴趣的话，完全可以忽略它。

用一个类似微博的系统进行举例说明。一个人可以制作一条文本记录(Entry)。他会用私钥进行签署，来展示这条记录确实来自与他。粉丝们可以追踪他发布数据的那条链, 随时获取更新。任何有他签署的记录就会被粉丝们的应用软件识别出来，这记录就相当于一条微博。别人可以通过添加记录到名人的链上来转发这条名人的微博。

**链(Chain)：记录是如何被编组到链中的？**

[](http://7fvhfe.com1.z0.glb.clouddn.com/%2540%252F%2F/wp-content/uploads/2015/03/124.png)

Factom中的链是记录的序列，反应了与一个应用相关的事件。这些序列是比特币2.0链的核心部分，用文件证明了这些事件序列，并提供了针对一个发生的事件序列的一个审计线索。另外通过加密签名，这些事件可以被证明是来自于一个已知的来源。

链是对于数据的逻辑诠释，放置在目录区块和记录区块中。目录区块表明了哪些链被升级，记录区块表明了哪些记录已经被添加到了链上。这有点类似于比特币完整客户端如何在本地保持未花费的交易输出（UTXO）数据。未花费的交易输出数据（目前）并不在存在于区块链中，但它会被完整客户端所解读。

**Factom点对点网络**

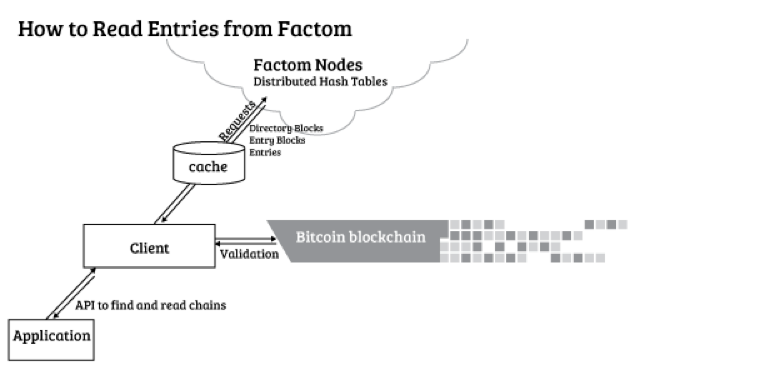
 Factom将会拥有点对点（P2P）网络，会实现两个目标：通信和数据保存

**Factom点对点通信**

Factom将会拥有类似与比特币的P2P网络。它会由完整节点构成，会拥有Factom全部的数据。完整节点会创建一个网状网络，会将有效数据填满整个网络。联合服务器组会是完整节点，但所有完整节点不会全是联合服务器组或审计服务器。这和比特币非常相像，矿工是完整节点，但所有完整节点不都是矿工。这会降低DDOS攻击联合服务器组的风险。它们可以在网络中连接到任何地方，去获取建立数据架构的所需数据。这些服务器是用来获得共识和广播签署数据的，他们会通过P2P网络来公布数据。P2P网络也限制了联合服务器组根据IP地址来禁止访问的能力，由于有效的通信数据已经被他们所连接的节点进行了混合。这也同样会帮助避免审查制度，由于审计服务器可以看到哪些记录应该被包含在记录区块中。审查服务器有将不良行为转化为良好行为的激励措施，所以他们才会被联合服务器组投票选出。

**数据保存和传播**

 Factom数据架构（目录区块、记录区块、记录）是为了让Factom能够被使用而建设的。它们是公开的而且将会被保存在两个地方。联合服务器组和审查服务器需要保存数据来做出正确的添加新纪录的决定。由于这些服务器拥有这些数据，所以它们提供出来作为一种服务，可作为身为完整节点的一个补充。而不完全节点可以共享数据，这些数据仅是和它们特定的应用相关联。不完全节点的节点发现是由分布式散列表（DHT）处理的。

[](http://7fvhfe.com1.z0.glb.clouddn.com/%2540%252F%2F/wp-content/uploads/2015/03/125.png)

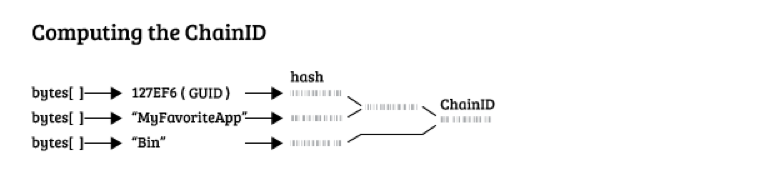
这个设置考虑到高效的节点数据分配，就算整个Factom数据集成长到很臃肿的体积，也能正常工作。分布式散列表（DHT）还使得数据可以被任何联合服务器组或完整节点独立保存。就算所有完整节点被从互联网中移除，数据依然可以通过很多对特定数据子集的团体进行分享。

五、对**Factom深度探讨**

**如何命名Factom链？**

Factom群组都使用链ID进行记录。链ID通过链名称演算而来。链ID是链名称的一段哈希值。链名称是一段任意长度的字节数组。由于从链名称转化成为链ID是通过哈希运算，因此这是一个简单的过程。但从链ID推算出链名称则不那么简单，因此需要一个查询表单。

用户必须提供一个链名称，这样链ID才能经由哈希值展现出来。这样可以避免未经运算的数据成为链ID，这部分数据被一直储存在目录区块中。这个公约消除了将不良的纯文本文档插入区块架构中的可能性。

[](http://7fvhfe.com1.z0.glb.clouddn.com/%2540%252F%2F/wp-content/uploads/2015/03/211.png)

链名称基本上可以是任意组合。它可以是一组随机数字、一串字符或一个公钥。单个个体应用从不同的链名称中获知意思。

一种可能的公约会是使用人类可读的字符当做链名称。这会让链架构成为一种有逻辑的层级，即使该链天生并不是分层的。用户甚至可以使用相同的命名规范，但通过制定简单的修改方式，可以确保不会在链和链之间出现偶然的重名。考虑到下面的路径：

MyFavoriteApp/bin

ChainID “space” with another, at the expense of just a few more bytes in the Chain creation.

斜线是另一分层的公约。斜线将ASCII字符串“MyFavoriteApp”和“bin”分开，代表了过渡到一个更深的层次。这两个字符串必须转化为字节，会有很多种方式来实现。字符串可以用UTF-16, UTF-32, ASCII或甚至IBM的EPCDIC进行编码。每种编码方式都会使得同一字符串产生完全不同的链ID，由于对链ID的运算已经从字节方面完成。此外，应用程序可以利用全局唯一标示符（GUID）作为他们命名规范中的第一段字节数组。这会消除一个应用程序的链ID“空间”和别的程序重叠的可能性，这仅仅在创建链的时候使用很少几个字符就可以搞定。

**使用Factoids购买条目信用（Entry Credit）**

Factoids是可调节和激励系统执行者的主要稀缺代币，条目（Entries）转化为Factom的权利可由条目信用进行表示。Factom设有两个价值承担机制，以执行不同的功能。Factoids可转化为条目信用，但条目信用则无法转化为Factoids。

Factoids交易方式和比特币交易方式基本相同，均支持多重签名安全保障、多项输入、多项输出等等。Factoid交易由特定（Factoid链）Factoid Chain进行管理。Factoid链相比其它Chains限制性条件更多。Factoid链条目必须是有效Factoid交易，否则Factom服务器将拒绝接受条目（Entries）。

协议中包含的Factoids功能可减少Factom和比特币膨胀和垃圾邮件。基于协议，Factoids可转化为条目信用并付款至Factom服务器。交易过程信息必须保持一致性，否则Factoids价值将会缩水。

Factoid转化为条目信用将通过Factoid链特定购买交易完成。该购买交易包括：

●转化的Factoid币值数额输出

●接受条目信用的公钥

一旦购买了数据条目信用，该数据条目信用便无法转移到其它公钥；购买的数据条目信用只能用于支付条目记录或进行联合服务器投票。由于条目信用无法重新出售，这样大大降低了资产被偷窃的可能性。条目信用私钥即使被保存到低安全区，偷窃风险也降低到了最低程度。

**使用条目信用购买条目记录权力（Entries）**

Factom条目添加需要给予稀缺资源-信用条目（由Factoids获取）。Factom条目添加需进行两个步骤操作。首先支付（委托）条目，该支付过程涉及两个事项：减少用户公钥关联的条目信用；指定条目哈希值。条目付款之后，服务器等待未经运算的条目，直到条目被展示出来。

**1. 条目付款**

○用户条目信用减少

○指定条目哈希值

**2. 插入条目**

○用户公布条目区块所包含的条目。

以上两个步骤具有很多优势。其中一个优势就是避免记录数据超额付款。这样，用户则无须被迫下载付款事项所生成的数据，只需下载系统激活数据即可，安全轻松地跳过付款信息。

另一项优势就是审计过程。掌握信息内容之前接受条目便于Factom服务器进一步进行审计。Adam Back发布的文章《blind symmetric commitment for stronger byzantine voting resilience》中提倡了比特币类似机制。如果用户或审计服务器正确显示已付款条目，但是没有联合服务器接受条目的话，审计制度便可进行验证。

扣  除条目信用交易将记录在类似于Factoid链的特定链中。联合服务器将通过有效条目信用交易填充链。

**通过中央服务器Oracle设置条目价值**

Factoids转化为条目信用兑换率由oracle进行决定。电脑系统中，oracles是将信息提供到未经系统验证或激活系统的过程。本案例中，oracle将Factoids转化成Factom条目信用的兑换率设为相对比特币交易值的1/10-1/100。

最初，兑换率将由中央服务器oracle进行执行。当我们将记录Factoid兑换率和交易量足量交易转入Factom链，我们将进行兑换率分散运算。合法来源链（交易记录）由各类交易Factoids兑换率和交易量进行决定。

完成付款订购之后，条目信用链将分配条目信用至相应的公钥。Entry包括：

●公钥

●购买的条目信用数量

**使用不带Factoids的Factom功能**

众多Factom用户可能不想使用钱包功能以及持有cryptocurrency资产，只想创建链（分类账）和添加条目。两个记录过程可分离Factom交易货币Factoids功能，Factom条目添加由条目信用进行表示。服务器和Factom代币接受者可接受用户通过比特币、常规信用卡付款等方式进行条目信用出售。用户可提供该条目信用的公钥。卖方可将相应数量的Factoids转化为条目信用，并将这些条目信用分配到用户公钥中。如此一来，用户便无须持有Factom服务器支持的Factoids进行条目信用购买。

基于管理层面，该功能非常强大。服务器可通过协议赚取Factoids。交易事项仅涉及服务器和协议。服务器可出售条目信用给用户，用户最终将反馈Factoids进行回报。条目信用不可转移，所以用户无法将条目信用分配给其它用户公钥。而且，私钥出售行为无法得以实现。交易中，可交易货币（Factoid）只能由交易双发进行转让。

**条目链第一个条目：类似“宅基地”（Homesteading）的先到先到的方案**

Factom将每个链的第一个条目保留给首先申请付款的用户。依照惯例，第一个条目记录着链审计规则包括归档URL、文本格式规则、链审计程序哈希值。该惯例可被称为“宅基地”。

通过信息展示，Factom可确保创建链的用户拥有第一个条目内容的决策权。

新链提交请求需支付10个条目信用为基础费用，如果后面想继续添加，需要支付每1024个字节空间再支付1条目信用（1 Credit per 1,024 bytes）。较高的10个条目信用的基础费用是为止防止大量新链的创建，新链的创建需要提交三个参数：

●ChainID哈希值

●（ChainID+条目哈希值）哈希值

●条目哈希值

以上三个哈希值被放置到条目信用链中。

一旦新链提交被接受，新链将在网络上进行公布，ChainID将立刻公开显示。为了确保新链提交信息的有效性，发布内容必须提供生成ChainID哈希值的新链名称。条目必须进行哈希运算来产生条目的哈希值。最后，该条目哈希值ChainID相关联信息必须匹配最终提交的信息。

如果以上条件未满足，Factom服务器将无法记录第一个条目，新链创建失败。

六、总结

Factom是基于比特币区块链协议而构建的另一层分布式的、匿名的、数据协议。赋予把比特币区块链技术拓展到无限应用场景中的能力。另外，用户无需持有加密货币也可以使用Factom的系统功能。

一个分布式的、不可篡改的、分类总账技术是比特币区块链技术所代表的本质创新和技术突破。很多人的梦想是把有数学法则保证的分类账本技术的诚实性和不可欺诈的特点应用到现实生活中。Factom通过允许基于区块链技术来创造新的分类总账，从而把区块链技术的好处和优势带到现实世界中。

**参考文献**

1.“Bitcoin: A Peer-to-Peer Electronic Cash System” Nakamoto, Satoshi. Web. 16 Nov. 2014<https://bitcoin.org/bitcoin.pdf>

2.”Can Blocks Remain Capped to 1MB Forever?” Transactions. Web. 15 Nov. 2014.<http://bitcoin.stackexchange.com/questions/18101/can-blocks-remain-capped-to-1mb-forever>

3.”Thin Client Security.” – Bitcoin. Web. 15 Nov. 2014.<https://en.bitcoin.it/wiki/Thin_Client_Security#Simplified_Payment_Verification_.28SPV.29>

4.”Evidence of Absence.” Wikipedia. Wikimedia Foundation, 11 July 2014. Web. 15 Nov. 2014.<http://en.wikipedia.org/wiki/Evidence_of_absence>

5.”Recording (real Estate).” Wikipedia. Wikimedia Foundation, 14 Nov. 2014. Web. 15 Nov. 2014.<http://en.wikipedia.org/wiki/Recording_(real_estate)>

6.”Secure Property Titles with Owner Authority.” Secure Property Titles with Owner Authority. Web. 15 Nov. 2014. <http://szabo.best.vwh.net/securetitle.html>

7.”Patent US4309569 – Method of Providing Digital Signatures.” Google Books. Web. 15 Nov.2014.   <http://www.google.com/patents/US4309569>

8.”Block Timestamp.” – Bitcoin. Web. 15 Nov. 2014.  <https://en.bitcoin.it/wiki/Block_timestamp>

9.”OP\_RETURN and the Future of Bitcoin.” – Bitzuma. Web. 15 Nov. 2014. <http://bitzuma.com/posts/op-return-and-the-future-of-bitcoin/>

10.”Goblin/chronobit.” GitHub. Web. 15 Nov. 2014.   <https://github.com/goblin/chronobit>

11. “How Can One Embed Custom Data in Block Headers?” Mining. Web. 15 Nov. 2014.[http://bitcoin.stackexchange.com/questions/18/how-can-one-embed-custom-data-in-block- headers](http://bitcoin.stackexchange.com/questions/18/how-can-one-embed-custom-data-in-block-%20headers)

12 .”Headers-First Synchronization Coming Soon to Bitcoin Core – CryptoCoinsNews.” CryptoCoinsNews. Web. 15 Nov. 2014.  [https://www.cryptocoinsnews.com/headers-first- synchronization-coming-soon-bitcoin-core/](http://bitcoin.stackexchange.com/questions/18/how-can-one-embed-custom-data-in-block-%20headers)

13.“Enabling Blockchain Innovations with Pegged Sidechains – Block Stream ” Web. 15 Nov. 2014.<http://www.blockstream.com/sidechains.pdf>

14. “[Bitcoin-development] 2-way pegging (Re: is there a way to do bitcoin-staging?)” / Mailing Lists. Web. 27 May. 2014.

<http://sourceforge.net/p/bitcoin/mailman/message/32108143/>.

15.”Could the Bitcoin Network Be Used as an Ultrasecure Notary Service?” Computerworld. Accessed 27 May. 2014. [http://www.computerworld.com/s/article/9239513/Could\_the\_Bitcoin\_network\_be\_used\_as\_an\_ ultrasecure\_Notary\_service\_.](http://www.computerworld.com/s/article/9239513/Could_the_Bitcoin_network_be_used_as_an_%20ultrasecure_Notary_service_)

16.”Proof of Existence.” Proof of Existence. Web. 27 May. 2014. <http://www.proofofexistence.com/>.

17.”Virtual-Notary.” Virtual-Notary. Web. May 27. 2014.  [http://virtual-notary.org](http://virtual-notary.org/)/.

18. “Commitment Scheme” Web. 16 November. 2014. <http://en.wikipedia.org/wiki/Commitment_scheme>

19. “Foundations of Cryptography: Volume 1, Basic Tools, (draft available from author’s site).” Cambridge University Press. ISBN 0-521-79172-3. 16 November. 2014. (see also<http://www.wisdom.weizmann.ac.il/~oded/foc-book.html> ) :224

20. “Real-World Sybil Attacks in BitTorrent Mainline DHT Wang Liang. Jussi Kangasharju. University of Helsinki. Web. 17 Nov. 2014. <http://www.cs.helsinki.fi/u/lxwang/publications/security.pdf>

21. “Sybil-resident DHT routing” University of Cambridge. Danezis George. Chris Lesniewski-Laas. Kaashoek M. Frans. Anderson Ross. Web. 17 Nov. 2014. <https://www.cl.cam.ac.uk/~rja14/Papers/sybildht.pdf>

22. “A Sybil-proof one-drop DHT” Lesniewski-Laas Chris. Web. 17 Nov. 2014.<http://pdos.csail.mit.edu/papers/sybil-dht-socialnets08.pdf>

23.“Art Provenance: What It Is and How to Verify It” Web. 17 Nov. 2014.<http://www.artbusiness.com/provwarn.html>

24.“Equine Appraisal: The Value of our Horses” Web. 17 Nov. 2014.<http://www.hgexperts.com/article.asp?id=7366>

25.“Proof of work” Web. 17 Nov. 2014. <https://en.bitcoin.it/wiki/Proof_of_work>

26. “Why one time passwords using nested hash chain are not used” Web. 17 Nov. 2014.

[http://security.stackexchange.com/questions/35135/why-one-time-passwords-using-nested- hash-chain-are-not-used](http://security.stackexchange.com/questions/35135/why-one-time-passwords-using-nested-%20hash-chain-are-not-used)

27. ”Proving Your Bitcoin Reserves” Web. 17 Nov. 2014. <https://iwilcox.me.uk/2014/proving-bitcoin-reserves>

28. “Distributed Consensus from Proof of Stake is Impossible” Web. 17 Nov. 2014.      <https://download.wpsoftware.net/bitcoin/pos.pdf>

**作者：**保罗·斯诺 (Paul Snow)，布莱恩·迪尔里 (Brian Deery)，吕旭军 (Jack Lu)，大卫·约翰斯顿(David Johnston)，彼得·柯比 (Peter Kirby)