Randao 可证公平随机数白皮书

randao.org

September 11, 2017

Abstract

Randao 基于区块链技术，提供开源的、去中心化的、社交化的、可证公平的随机数生成服务。

Randao 继承了常用随机数发生器的不可控制性及不可预测性，同时具备其所不具有的可参与性及可证公平性。Randao 通过为每个利益相关的个体提供参与的通道，使个体可以观察到自己对随机数生成的影响。透明、不可逆的随机数生成过程保证了结果可证公平性。

利用 Randao 服务，用户可以针对各个使用场景，快速地构建可证公平的应用。这些场景包括但不限于公共管理、娱乐、体育、金融、企业内部管理等。

第一部分　问题及目标

日常生活中处处可见对随机性的需求和应用，比如游戏、博弈、抽样、公平分配等。人们为了产生随机数，也发明了各种方法，包括通过掷骰子、转转盘、抛硬币等统计方法产生随机数；通过调用计算机语言生成伪随机数；利用量子力学原理获取随机数等等。这类传统的随机数生成问题已得到了比较充分的研究与应用。

虽然上述随机数发生器很好地解决了随机数的随机性、不可控制性、不可预测性等方面的问题，但是却缺乏去中心性与可证公平性。一个真随机数并不天生是可证公平的，以 NIST Beacon 提供的随机数为例，即便其从宇宙背景辐射进行采样获得熵，NIST 仍然在他人之前提前获知最新的随机数，同时 NIST 具有对生成的随机数进行挑选和干涉能力。

自然地，人们希望找到一种更公平的随机数生成和发布机制。而区块链作为一个去中心化的平台，为可证公平的随机数生成提供了天然的基础。

但是在公有区块链上，设计一个可用的随机数发生器难度更大。除了基本的随机数统计学要求外，公有链上一个堪用的随机数发生器至少应该满足无法预测，不可操控，难以串谋，可证公平这几个特点。此外区块链本质上是一个状态复制机，不同计算设备对区块链的所有历史数据进行重放，并保持状态一致，如果进行重放的过程中需要节点自主生成随机数，这就要求该随机数发生器在每台设备上运行结果相同，还要求没有人可以提前知道结果，这本来就是矛盾需求。因此，在区块链上，随机性的引入必然和单台计算机的实现机制截然不同。

另外，区块链上的随机数发生器如果只有一个参与方，显而易见的是，该参与方相对其他人可以提前知道即将生成的随机数是什么，无法保证公平。如果是多人合作生成，则必然存在最后一个参与者，该参与者相比其他人拥有更大的权力：可以提前知道即将生成的随机数，可以根据情况改变提交的内容或者选择不提交，从而给予了最后参与者操控随机数的能力。此外多人合作生成随机数还有潜在的串谋可能，应该从机制上增加串谋难度，降低串谋的概率。

因此，如何在区块链上设计并实现可证公平的随机数发生器成为近年来一个重要的研究问题。自从 Randao 团队在 2015 年提出使用 Commit Reveal 方案后，又分别有 Vitalik Buterin 提出的 Randao++ 方案[1]、部分 DApps 使用 Oraclize 从链下服务获取随机数的方案，来实现区块链随机数生成。以太坊基金会也将链上随机数生成列为其接下来两到三年需要思考的一个重要问题[2]，邀请各方协作参与解决。

本白皮书详细描述了 Randao 如何通过 Commit Reveal、BLS 等随机数生成方案，实现了公平随机数生成与发布时须满足的随机性、不可控制性、多方参与性、串谋不经济等多方面的目标。

第二部分　随机数及随机数发生器

1　真随机数与伪随机数

根据随机数生成时所采用的方法及其可控制性和可预测性等，可将生成的随机数分为伪随机数与真随机数两类[3]。

伪随机数一般由确定的算法生成的，其分布函数与相关性均能通过统计测试。但与真随机数相比，它们由算法生成，而不是一个真实的随机过程。伪随机数也只是尽可能地接近其应具有的随机性，但是因为有“种子值”，所以伪随机数在一定程度上是可控可预测的。伪随机数可使用取中法、同余法、移位法、梅森旋转算法等方式产生。

目前编程语言一般都提供简单易用的编程接口或系统调用供用户生成随机数，例如 Java 语言中的 java.Math.Random() 及 java.util.Random() 方法。通过这些方式生成的随机数一般都是伪随机数。如果已知产生随机数时所使用的种子或已产生的随机数，则可以获得接下来随机数序列的信息。

真随机数的产生不可预计，也不可能重复产生两个相同的真随机数序列。真随机数一般使用物理现象产生，比如掷钱币、掷骰子、晃动鼠标、转转轮、使用电子元件的噪音、使用大气噪声、核裂变等。真随机数发生器的技术要求一般比较高，生产效率一般比伪随机数低。另外，如果信息熵的信息量很有限，不一定能产生真随机数。真随机可以进一步区分为统计意义上的随机以及量子效应上的随机。一般认为，由于量子力学内在的随机性，其产生的随机数比传统物理学通过统计产生的随机数更“真”。

Linux 内核提供了统计方式的真随机数生成器。它利用机器的噪音生成随机数，噪音源包括各种硬件运行时速，用户和计算机交互时速，比如击键的间隔时间、鼠标移动速度、特定中断的时间间隔和块 IO 请求的响应时间等。网站 http://random.org/利用大气噪音生成真随机数并对外提供。另外，通过监听真空内亚原子粒子量子涨落产生的噪音，澳大利亚国立大学的科学家们建造了随机数发生器并提供给互联网用户。

---

[1] https://www.reddit.com/r/ethereum/comments/4mdkku/could_ethereum_do_this_better_tor_project_is/d3v6djb/

[2] https://github.com/ethereum/research/wiki/Problems#21-on-chain-random-number-generation-63

[3] 在计算物理学等领域，有时也将随机数分为三类，伪随机数、准随机数及真随机数。

2 代表性的随机数发生器

不同场景对随机数的要求不一样，一些场景下随机数需要高度保密并保证安全，比如生成密钥中使用的随机数。有些场景中，需要的是公开可见的随机数，在生成之前不可预测，生成后不可篡改或者可事后审计，比如现场掷骰子等。

2.1 中心化随机数发生器

在以区块链为代表的去中心化公共账本提法之前，数据都是私有的，中心化的。因此在此之前的随机数发生器也都是中心化的，其中的代表方案有 NIST Randomness Beacon 以及 random.org 等。

2.1.1 NIST 随机数生成器

NIST Randomness Beacon[4]用于实现公共随机数源。它使用两个独立、商用的随机数发生器，每个发生器配备一个独立的物理熵源和 SP800-90 认可组件。NIST 随机数生成器旨在提供不可预测、自主、一致的随机数源。不可预测是指：任何算法都无法预测该生成器将会给出的随机数。自主是指：能够抵抗不相关者介入或阻止分发随机数的过程。一致是指：一组用户访问该服务能够确实地获得相同的随机数。

2.1.2 random.org

random.org 使用大气噪音生成随机数。即先用录音设备获得大气中的声波，再检测其细微变化作为生成随机数的熵源。random.org 还提到了两类物理现象作为熵源的比较：量子现象和混沌现象。量子现象作为随机数源是利用了在原子尺度下粒子的行为具有随机性, 而且其本质还未被人类发现，因此可以将其看做一个具有良好不确定性的熵源。混沌现象是指在混沌系统中, 初始量的微小差异会导致未来的发展截然不同, 因此除非获得初始时刻的全部准确信息, 则无法预测未来的发展趋势。实际上应用这两种方法均能实现不可预测的随机数发生器，random.org 使用大气噪音生成随机数的方法就属于后者。

2.1.3 其他物理现象

除了使用量子效应以及射频噪音，还有使用原子衰变的甚至于熔岩灯的。这类方案尝试提供真随机数，瞄准的目标场景更多是科学领域。本质上是对骰子的数字化改进。由于随机数都由单一的组织机构提供，仍然是中心化的方案。

2.2 Bitcoin Beacon

斯坦福和普林斯顿的 Bonneau&Goldfeder 等[5]于 2015 年提出可将比特币的区块数据作为一个不依赖第三方的公开随机源，并分析了区块头所包含的熵，以及根据该熵所生成的随机数的安全性。

[4]https://beacon.nist.gov/home

[5]https://eprint.iacr.org/2015/1015.pdf

用比特币区块数据生成随机数的问题在于，其通用安全性不够高，无法防止“块保留攻击(Block Withholding Attacks)”，即参与者可以贿赂矿工丢弃不利于自己的区块，从而在博彩类的应用中获得相对优势。比特币单个区块的奖励是固定的，因此，所有依赖于该方法获得的随机数有一个固定的安全上限，不能根据应用的具体情况动态调整，导致其适用性有很大的限制。此外，Bitcoin Beacon 方案中，普通用户无法参与区块的生成，虽然串谋的成本较高，但矿工并不能避免嫌疑。因此该方案仍然不是一个可证公平的随机数方案。

2.3 Algorand

区块链共识算法主要解决两个问题，谁负责出块和出现分叉如何解决。难点通常在如何解决分叉上，即当出现两条或多条合法分叉的情况下，如何让所有人一致同意选择一条继续走下去, 这就需要一个统一的判断标准, Nakamoto 共识选择最昂贵的, 成本最高的那条分叉, 而昂贵的标准需要是客观可验证的, 也即工作量证明。但并不意味着解决分叉只有这一种办法, 比如有一个上帝的骰子, 所有人认同该骰子的公正性, 那么每当出现分叉的时候, 就由该骰子决定选择某个分叉，这样的共识也能保证系统的正常运作。Algorand 使用 Verifiable Random Functions (VRF) 来构造了一个 Common Coin，起到上帝骰子的作用，在系统各节点无法达成共识的时候，犹豫不决的节点可根据 Common Coin 给出的结果，快速站队，解决分歧。Algorand 中多处使用了 VRF，包括选择出块人及委员会成员等。

VRF 的流程和特点：考虑如下情形：有一个公共的难以找到原象的函数 F，现在 Bob 给出一条信息 x，要求 Alice 用她的秘钥 s 计算出 $F(s,x)=v$ 作为一个随机数输出，因为 s 对公众不可见，所以 Bob 收到 v 难以判断这个 v 是否是 Alice 诚实计算 $F(s,x)$ 得出的。VRF 实现了：让 Alice 在给出 v 的同时也给出一个 $proof$ 和一个对应 s 的公钥 p，Bob 可以用 p 和 $proof$ 来验证 v 是否是 $F(s,x)$。这个过程乍一看和数字签名相同，但 VRF 有额外的特点：不同于数字签名的一条信息可以有多个签名结果，VRF 的输出 v 有且只有一个。

VRF 和 Bitcoin Beacon 类似，都有相似的缺点，并不适合用来作为可证公平的随机数发生器。

2.4 Dfinity

Dfinity 的共识算法也是建立在随机数的基础上，但和 Algorand 的差异在于，其随机数的生成是由一组人通过 BLS 签名算法实现的，具有更好的安全性。在 Algorand，VRF 的签发是由一个参与者完成的，因此该参与者可以选择不发布对自己不利的签名。Dfinity 的方法由一组人产生签名，任何个人都无法预测签名结果，单个人无法阻止签名发布。

Dfinity 的 BLS 是一种门限签名技术。首先将用户分组，第一轮由一个组产生一个随机数，之后每一轮选一个组对上一轮产生的随机数签名，作为这一轮的随机数输出，每个成员都无法提前预知签名结果。其中签名过程使用 BLS 签名机制，保证在签名过程中没有个体能够提前预知签名结果，因此无法操纵随机数。

BLS 很好的解决了 Withholding 攻击的问题，且生成随机数过程不可操纵，无法预测，很难串谋，是一个比较理想的随机数生成方案。

2.5 DAO

骰子是一种传统的公共随机数，一人掷骰子，所有利益相关方围观监督并认可结果。但因为骰子支持的围观人数有限且无法事后审计，因此骰子或者抽签这样的公共随机数在应用到更广泛场景的时候往往难以服众，而被质疑暗箱操作或流程不科学等。

这种质疑最著名的例子包括 1969 年美国实行的基于抽签的兵役系统。在记者、电视台摄像机、政府官员和公证员的监督下，义务兵役征兵官员分别从包含日期和包含数字的两个滚筒内抽出一个密封囊。该数字决定了在该日期出生的男子将接收入伍通知的顺序。例如，如果征兵官员抽出的日期是 4 月 22 日、数字是 42，那么当年 4 月 22 日度过 20 岁生日的男子将在第 42 批收到入伍通知。征兵官员会不断抽出所有日期和数字组合，直到将所有日期排好顺序，该顺序决定了适龄青年收到兵役通知的先后顺序。美国人很快发现了这个数字的分布并不均匀，11 月和 12 月出生的人的排名显著靠后，有人经过蒙特卡洛模拟后发现，出现接近该分布的概率只有 0.09%。出现这种低概率排序的原因可能有很多，比如密封囊在滚筒中混合不匀，但因为抽签结果无法审计，难以洗脱操纵嫌疑。

而在阴谋论者眼中，除他自己外的全世界所有人都是可能合伙来对付他的。一个他无法参与的随机数生成过程，无论理论上如何完美，实践上如何严密，都可能遭受串谋的质疑，而参与是破解不信任的良药。

传统的组织结构中，因为组织技术所限，代理委托关系是常用手段，但代理者是天然的结构中心，具有信息的非对称优势，其公正性只能通过其他制度约束进行修修补补。DAO 是一种无中心的组织结构，组织规则由代码描述并强制执行，任何人可自由加入退出，不同参与者间地位平等，刚好可以满足我们对于公共随机数的设计目标。因此区块链技术以及 DAO 理论，可以用来支撑并指导公共随机数的设计和实现。

Randao 正是基于区块链技术以及 DAO 理论，以低参与门槛为指导原则，破解了随机数生成及发布过程中的不透明与不信任，让随机数生成可证公平。

3 随机数发生器的评价

单机版的随机数发生器输出的随机数都是自己使用，因此只要考虑随机数的内在质量就可以，即是否满足统计学上的标准，不存在欺骗和信任的问题。但一个公共的随机源服务，尤其是供多个参与方共同使用的共享随机数，需要重新审视评价标准。

公共随机数利益相关方可分为两类，一类是生产者，一类是消费者。根据生产者的数量多少，还可以进一步分为单独生产模式和合作生产模式；根据使用场景不同，也可分为私有领域和公共领域。在私有领域，因为不存在信任问题，不在本白皮书的讨论范围。因此本文主要讨论公共领域的随机数方案的评价标准，以及各个方案的优缺点和使用场景。

一个理想的公共随机数首先应该是公平的，即所有的相关方在随机数生成过程中绝对平等，任何人都不具有相对优势；其次应该是公开的，包括生成的步骤是公开的，使用的方法是公开的，结果是公开的；然后还要保证整个过程和结果是可审计的，可以证明生成的随机数没有被篡改，经得起事后检查。

根据第一章问题及目标的描述，为了评价一个随机数生成方案，我们提出以下的标准：

1. 不可预测：不可预测是针对所有参与者的，不管是生产者和消费者，都无法根据历史数据

预测下一个随机数的可能值，即便是稍稍提高一点点预测的成功率都做不到，即具有马尔可夫性质。在公共随机数的方案中，还要求任何人根据任何公开信息也都不能提高预测概率，例如 Bitcoin Beacon 的方案中，即便知道区块的历史数据，矿池的公钥，待打包的交易列表等，也无法获得预测上的优势。

2. 不可串谋：在随机数的生成过程中，部分参与方联合起来，互相交换各自的私有信息，并不能影响随机数的生成过程或改变随机数的结果，或具有其他比较优势，比如相比其他人提前获得即将生成的随机数的结果。
3. 不可提前获知：随机数的参与方同时知晓该随机数，任何一方不能提前知道结果。
4. 不可篡改：即随机数的生产者不能伪造一个随机数出来，而当一个随机数生成好后，该随机数无法被任何人修改。
5. 不可选择：随机数的生产过程可能同时有很多个随机数生成，生产者无法只选择其中的某一个提供出去，或用其中一个替代另外一个。
6. 不可隐瞒：生产者在随机数生成完成后，不能拒绝公开该随机数。即生产好的随机数一定会被公开，无法被隐藏或者撤回。
7. 可参与：随机数的生成过程中，随机数的相关方可以容易的参与进来，随机数生成方案应该为一般人的广泛参与提供便利，降低或消除参与门槛，参与的权力不应该被剥夺。
8. 可审计：在随机数生成过程结束后，其整体过程是可以被时候审计的。
9. 成本：随机数的生产成本应该尽可能低
10. 响应速度：随机数的生成过程应该足够快

考虑前述代表性的随机数发生器，根据上面的评价标准，可得到如下评测结果：

	random.org	Bitcoin Beacon	Algorand	Dfinity	DAO
生产者	独自	独自	独自	合作	合作
不可预测	无法预测	无法预测	无法预测	无法预测	无法预测
不可提前获知	可以	难	难	难	难
不可串谋	可以	难	很难	很难	很难
不可篡改	容易篡改	不可篡改	不可篡改	不可篡改	不可篡改
不可选择	可选择	可选择	可选择	不可选择	不可选择
不可隐瞒	可隐瞒	可隐瞒	可隐瞒	很难隐瞒	很难隐瞒
可参与	不可参与	不可参与	不可参与	可参与	可参与
可审计	无法审计	无法审计	可审计	可审计	可审计
成本	低	低	中	中	高
响应速度	快	慢	中	中	慢

第三部分　Randao 协议设计及评价

Randao 是区块链上的随机数服务，其底层实现可接入不同的技术方案。在 Randao 项目规划中，至少有两个随机数发生器方案。随着研究的深入、技术的进步以及市场需求的变化，Randao 将进一步实现其它的随机数生成方案。目前规划中的方案包括 Commit Reveal 方案、BLS 方案等。

1　Commit Reveal 方案

使用 Commit Reveal 方案生成随机数的基本流程分为三个阶段。

1.1　第一阶段：收集有效 $sha3(s)$

所有希望参与随机数生成的生产者，在指定的时间窗口期内（例如以太坊的 6 个出块周期，大约 72 秒），向合约 C 发送 m 个以太的保证金，同时附上其任意挑选的数字 s 的 $sha3(s)$。

1.2　第二阶段：收集有效 s

之前成功提交 $sha3(s)$ 的所有生产者，在第一阶段结束后，在第二阶段指定的时间窗口期内，向合约 C 发送各自在第一阶段选中的数字 s。合约 C 检查数字 s 是否合格的。如果合格，保存该 s 到最终随机数生成函数的 $seeds$ 中。

1.3　第三个阶段：计算随机数，发放保证金及奖励

全部的 s_i 收集完成后，以 $f(s_1, s_2, ..., s_n)$ 作为最终的随机数，把随机数写入到 C 的存储空间中，向所有请求该随机数的其他合约返回结果。把生产者在第一阶段发送的保证金退还，同时把本期随机数生成过程中，其他合约支付给 C 的手续费作为奖励发送给本期的所有生产者。

1.4　方案约束

为了保证随机数结果不被操纵，兼顾安全和效率，合约 C 有如下这些额外规则约束：

1. 第一阶段中，如果先后有两个同样的 $sha3(s)$ 被提交，只接受第一个。
2. 第一阶段中，有一个最小参与人数的设定，如果在窗口期时间内未能达到预设人数，则本期随机数构造失败。
3. 如果提交的 $sha3(s)$ 结果被合约 C 接受，则在第二阶段必须提交 s。
   (a) 如果在第二阶段的窗口期中某个参与者没能提交 s 的话，则在第一阶段中发送的 m 个以太会被没收，不再返还。
   (b) 如果在第二阶段未能收集到全部的 s，则本期随机数生成失败，退还其他合约支付的手续费后，将本期收集的保证金分发给在第二阶段成功发送 s 的其它生产者。

2 BLS 方案

2.1 BLS 签名机制

2.1.1 BLS 签名思想

为介绍 BLS 方案，需要先介绍 BLS 签名。BLS 签名的思想为：N 个成员组成一个组，一个组有一个逻辑上的私钥 S（使用 S 对消息签名的结果记为 SIG，S 对应公钥记为 P），而每个成员 i 只拥有这个秘钥的一部分 s_i，完整的秘钥 S 不存在于任何人手中，没人能直接计算出 SIG。签名时，每个人用自己的秘钥 s_i 对一个信息签名得到 sig_i，只有收集了 k 个（某个预先设定的值）成员的签名，才能够计算出签名 SIG。

2.1.2 BLS 签名特点

1. 秘钥 S 由全网共同决定，但 S 的真实值不会出现在任何计算过程中，除非全组所有成员（当部分成员承担秘钥生成任务时，则为这些成员）串通，否则无法得到 S。
2. 每个人都无法预先知道使用 S 进行签名的结果 SIG。无法操控 SIG 的值。

2.1.3 初始化

要让 BLS 签名机制运行起来，需要一次初始化，生成成员的私钥和全组公钥。初始化过程如下[6]：

1. 每个成员 i 生成自己的随机数组 ran_i（对其他人保密，包括同组成员）。
2. 使用一个（BLS 签名机制给出的）函数 f 计算出 $send_{ij} = f(ran_i, j)$，通过私密通道发送 $send_{ij}$ 给成员 j（对不同的 j，$send_{ij}$ 不同；f 有单向性，用 $send_{ij}$ 和 j 无法得到 ran_i）。对所有 $j(= 1, 2, ..., N)$，成员 i 要对成员 j 发送 $send_{ij}$（对于 $i = j$，则计算出 $send_{ii}$ 记录在本地）。
3. 每个成员 j 收集数据 $send_{1j}, send_{2j}, ..., send_{Nj}$，用（BLS 签名机制给出的）函数 v 验证 $send_{ij}(i = 1, 2, ..., N)$ 的合法性（$v(send_{ij}) = 1$ 则合法，$v(send_{ij}) = 0$ 则非法）。
4. 全部合法后，用它们计算出自己的秘钥 s_j，s_j 对应的公钥 pub_j，和 S 对应的公钥 P。
5. 将 pub_j 以及 P 广播。

2.1.4 执行步骤[7]

1. 每个成员 i 用秘钥 s_i 对消息 M 签名得到 sig_i 并广播 sig_i。
2. 每个成员 j 收集其他成员广播的 sig_i，并用 pub_i 验证，通过验证则作为有效签名接受。当收集到 k 个有效签名（包括自己的签名 sig_j），则计算出最终签名 SIG 并广播 SIG。

[6] Distributed Key Generation: https://en.wikipedia.org/wiki/Distributed_key_generation

[7] https://en.wikipedia.org/wiki/Shamir%27s_Secret_Sharing

2.2 BLS 签名产生随机数

使用 BLS 签名产生随机数的基本思想为：将全网分为 h 个组，第一组生成一个随机数 SIG_1，使用 SIG_1 选取下一个组，被选中的组签名这个随机数得到 SIG_2，使用 SIG_2 选取下一个组，再对 SIG_2 签名得到 SIG_3。如此重复，过程中已经参与过随机数生成的组不再参与随机数生成，直到第 h 个组生成 SIG_h 作为最终的随机数输出。具体流程如下：

1. 将全网节点分为 h 个组 $G = g_1, g_2, ..., g_h$
2. g_1 生成一个随机数 SIG_1 并广播，将 g_1 从 G 中剔除。
3. 对 $i = 2, 3, ..., h$，执行：取 $g_{work} = G\,(sig_{i-1} mod |G|)$，将 g_{work} 从 G 中剔除，g_{work} 对 SIG_{i-1} 签名得到 SIG_i 并广播。
4. 最终得到的 SIG_h 为最后输出的随机数。

这个过程保证每一步的签名都是一个随机数，每个组都无法预测后续组的签名结果。只有最后一个组有操纵 SIG_h 的优势，当最后一个组有不超过 k 个非法成员，则 SIG_h 是安全的，但是无法预测哪个组会担任 SIG_h 的生成任务。

2.3 BLS 安全性讨论

上述执行过程中可能发生的问题：

1. 在 BLS 签名初始化时，一个组的成员如果串通，则会让 S 泄露，成员可以预先知道签名结果。若要做到这一点需要所有成员对组内公开自己的秘钥 s_i，若有一个人不参与，则无法得到 S，也就无法预知签名 SIG。但是除了这个方法，有一种更简单的攻击方法：组内 k 个人串通，k 个人使用 BLS 签名机制生成这 k 个人的秘钥并共享，那么在运用 BLS 签名就可以把剩余的 $N-k$ 个人排除在外，独占这一组的签名权。
2. 在 BLS 签名执行的第二步，如果一个成员收集了 $k-1$ 个其他成员的签名，并与自己的 (还未广播的) 签名组合计算出 SIG，发现 SIG 对自己不利，则可以选择不广播自己的签名。如果因此造成全网只有 $k-1$ 个成员签名被广播，则签名会失败。当全网有多于 $N-k$ 个成员可能做出此行为，则这个情况有发生的可能性。
3. 假设当 $k < N/2$，若全网有 k 个成员对一个并非全网大多数成员同意的消息进行签名，则这 k 个成员能够“伪造”集体意愿。
4. 如果生成的随机数与全网大多数成员有直接的利益关系，那么在生成随机数的最后一步，很可能有超过一半的成员拒绝给出正确的签名 SIG_h 使得随机数生成失败。

综合 1, 2 , 3 点，BLS 系统可以抵御 $max(k-1, N-k)$ 个非法成员。在 DFINITY 中，$N = 400$，$k = 201$，当某个组内超过一半成员为非法成员，才会出现签名失败或伪造签名的情况。事实上 BLS 签名方法在将全网节点分组时使用 VRF 进行随机分组，在这个过程没有作弊的前提下能提供较高的容错率：以全网有 $10,000$ 用户为例，当 $3,000$ 人为恶意用户时，分组时某一组会有超过一半恶意节点的概率不足 $1e^{-17}$，4000 人为恶意用户时，这一概率为 $1e^{-5}$，而且概率会

随着 N 提高 ($k = N/2 + 1$) 而进一步降低，但是无论 N 取到多高，都无法做到 50% 容错，当恶意用户上升到 50% 左右时，上述概率会急速上升到 50% 附近。因此可知这一随机数生成机制并不适合用于与全网利益直接相关的问题之中，而对于只与全网部分节点 ($< 50\%$) 利益相关的问题中，上述 4 个问题发生的可能性很低。

2.4 防止串谋与贿赂

2.4.1 初始化阶段

- 有兴趣参与提供 BLS 方案服务的生产者向系统发起申请，由系统确定入围帐号
- 设定 Feldman 秘密分享的参数，通过分布式密钥分发机制形成一个 BLS 服务提供组。
- 该 BLS 服务组的每个成员向获得的私钥对应的以太坊地址转入一笔押金，该押金在 BLS 服务组存续期间，不得转出。如果在 BLS 服务组到期之前，有人提前退出，则扣除其部分押金。

2.4.2 提供服务阶段

- 随机数消费者向该 BLS 服务对应的智能合约 R 发送请求，请求包括待签名的内容 (如果为空则为当前块的 $hash$) 以及支付的手续费。智能合约 R 记录下待签名内容。
- BLS 服务组检测到区块链上的随机数生成请求后，链外协调服务组一起签名并合并签名得到随机数。该随机数由一指定用户写入到智能合约 R 中。

2.4.3 审计监督

- 为了防止消费者和某些服务者成员勾结，提前获取某些内容的签名结果，然后有选择的提供待签名内容给 BLS 服务，获得比较优势从而获益，可以通过以下的惩戒条例来避免串谋，并使得服务者不敢接受贿赂提供私下签名。
- 如有人出示某成员私钥对内容 s 的签名结果，且该内容 s 未出现在智能合约的已签名的历史记录中，则可判定该成员私自签名，罚没押金。
- 考虑到区块链重组的可能性，该成员可以申辩，申辩内容应该包括包含内容 s 的区块。
- 成员必须保有一定的押金在系统中，该押金的私钥拥有者可以转移走押金，这样可避免成员向特定人群公开自己的私钥。
- 为了防止成员在 BLS 服务组到期之前，私自转走押金，增加一条规则，如在默认到期之前提前转走押金，则扣除 20% 的押金给坚持到期的其他成员。
- 每个 BLS 服务组会有一个健康指标，该指标由押金状况，签名响应速度，节点在线率等指标构成。健康指标对外公示，消费者可挑选服务组的服务。服务组的健康度可能会影响其服务的收费标准。

3 Commit Reveal 及 BLS 方案评价

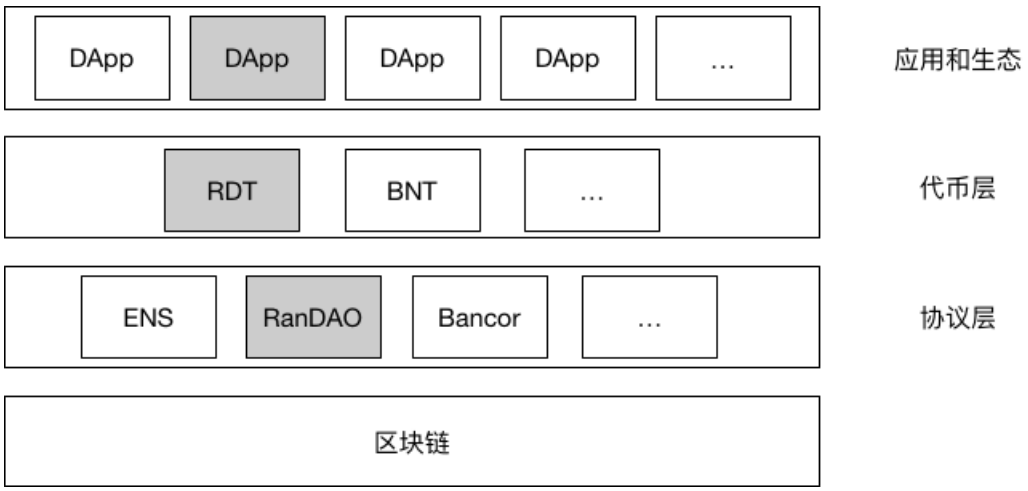
第一种 Commit Reveal 方案中，其缺点主要是生成随机数的速度较慢，从接到随机数生成请求，到生成随机数，至少需要 10 个块以上的组织协调时间，在以太坊不缩短出块间隔的情况下，耗时在 3 分钟以上；因为需要参与者多次发送交易提交数据，其生产和使用成本较高。但该方案的优势在于，其参与门槛基本为零，任何人都可以随时加入一个随机数的生成过程，在防止串谋和可证公平方面拥有非常特殊的优势。

BLS 签名方案刚好是对第一种方案的一种很好的补充，因为生成的过程在链外组织，响应速度快，通常只需要一个区块的时间就能生成随机数；消费者发起随机数生成请求，生产者在下一个块写入随机数，只需要发送两次交易就可以完成随机数的生成和调用，生产和使用成本都很低，适合用于高频，同时对于防串谋的需求不那么苛刻的场景。

第四部分 Randao 技术架构

1 概述

Randao 整体架构于区块链上，并致力于打造成区块链上标准的可证公平的随机数协议。所有部署在区块链上且对随机数有使用需求的 DApp 都可以通过 Randao 为其用户提供基于随机数的各种服务。同时结合 Randao 提供的具有经济激励机制的代币，为协同产生随机数的生产者提供收益。此外，围绕随机数协议，Randao 还将通过各种技术形式推动 Randao 的生态系统建立和发展，下图给出了 Randao 在以太坊上的整体架构：



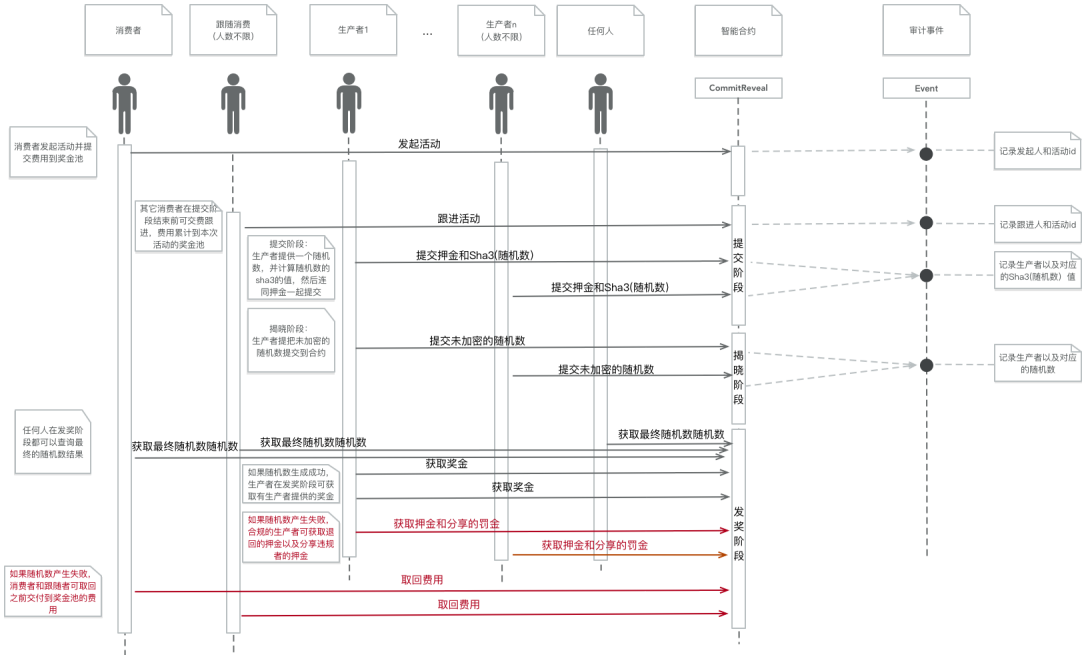
其中协议层包含了支撑 Randao 服务运行的智能合约；代币层是支撑 Randao 经济系统运行的代币实现；应用和生态层包含了运营和推广 Randao 服务的必要 DApp 产品。

2 协议层 - 随机数协议

此部分是 Randao 的核心，它将以一个或多个区块链上智能合约的形式封装了各种随机数生成的算法和对应的业务逻辑，这些随机数服务的多样性确保我们的协议能够覆盖到更广泛的业务场景，以满足更多 DApp 的需求。比如 Commit Reveal 是其中一个专门为多人参与且具有多个阶段操作过程的随机数生成业务流程，时效性较差，而 BLS 方式则提供了一种快速生成随机数的机制，更适需要快速获得随机数的业务系统对接需求。随着平台的发展和推进，Randao 将结合市场需求封装更多的随机数生成模式。为了把 Randao 打造成区块链上的标准随机数服务。随机数服务都将经过严格的测试以及安全审计，确保服务上线的质量。

2.1 Commit Reveal 合约

Commit Reveal 是一种在规定时间内可以有多人一起参与产生随机数的流程，下面的时序图给出了智能合约的程序处理流程：

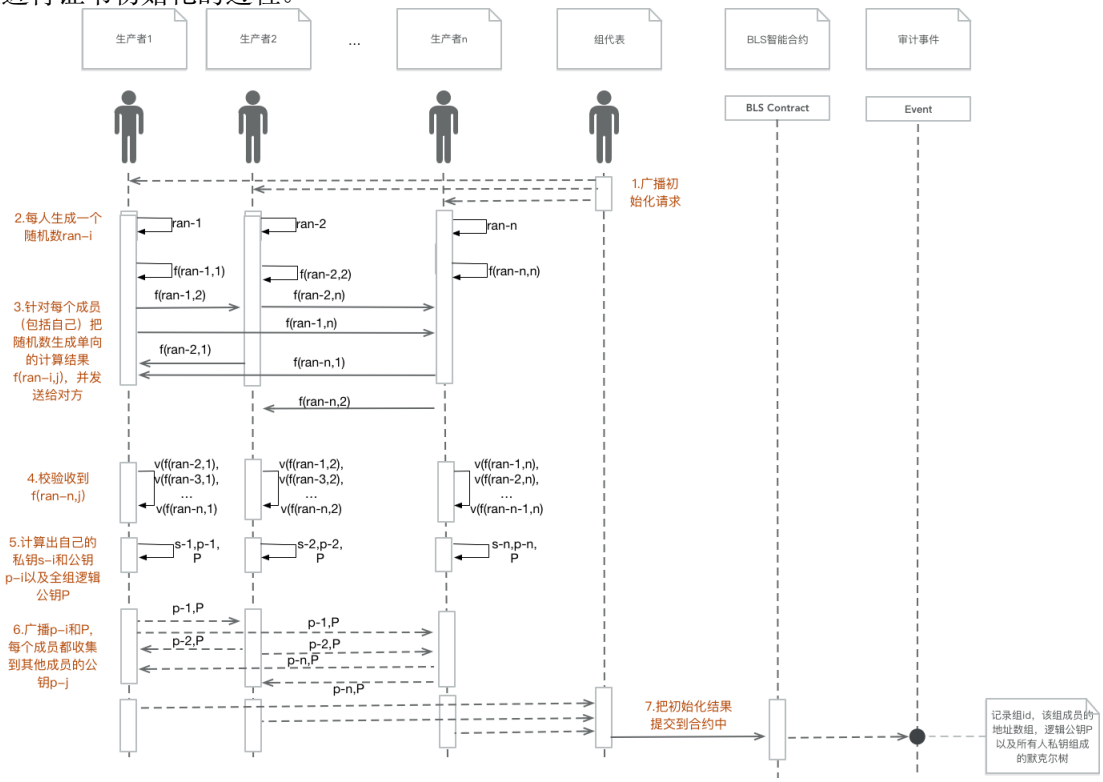


2.2 BLS 合约

BLS 也是一种多人参与生成随机数的方式。但其目标是给消费者在可证公平的前提下以一种更高效的方式来提供随机数。首先 BLS 生成随机数涉及分组和初始化。分组就是把所有参与人随机分成 M 组，每组 N 人。初始化是分组结束后根据 BLS 的签名机制为每个组员生成自己的私钥以及全组的公钥 P。为了提高效率分组为线下完成，且分组结果会运行一段时间，然后定期重新分组，以保证公平性。下面我们通过时序图的方式，分别描述 BLS 的成员组初始化和随机数产生流程。

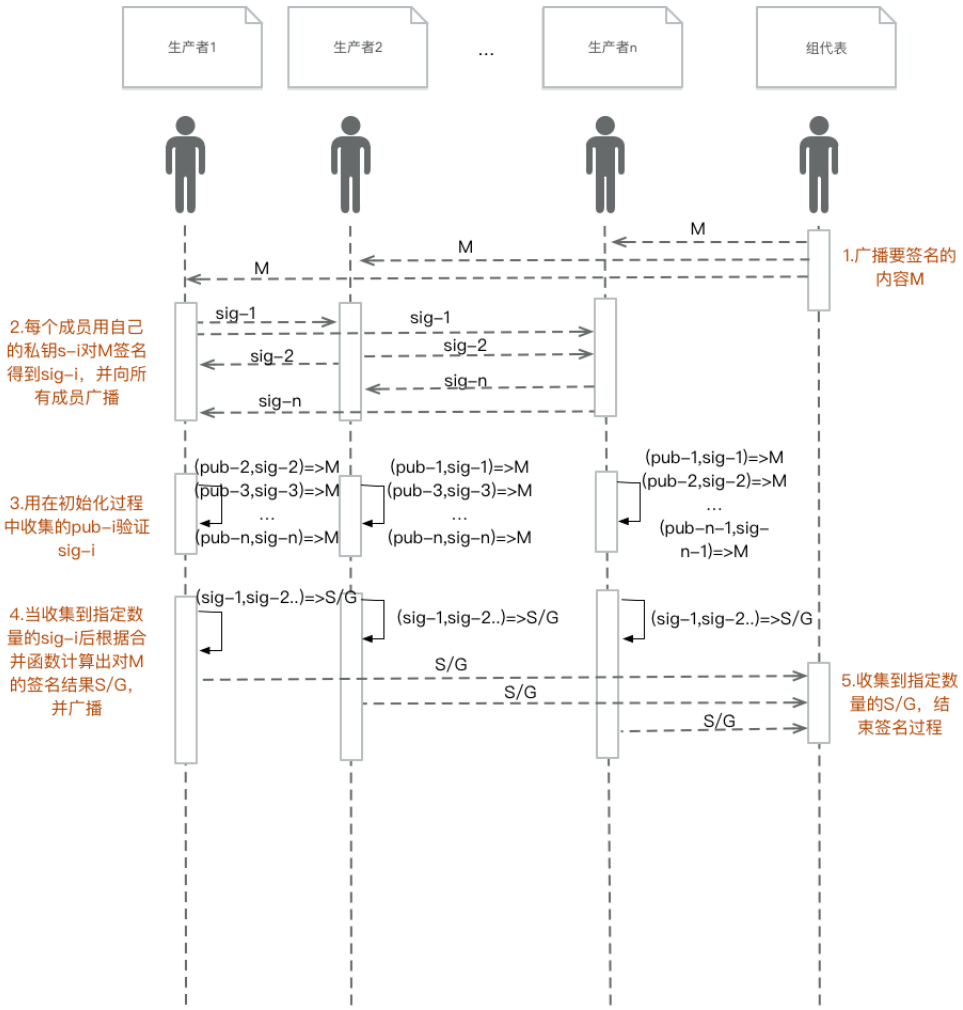
2.2.1 成员组初始化流程

每个组由成员和组代表组成，组代表只负责做内容和信息的传递和记录，不干预随机数的生产过程。成员是实际承担产生随机数的实体，因此也称生产者。下图描述了分组结束后，整个组进行证书初始化的过程。



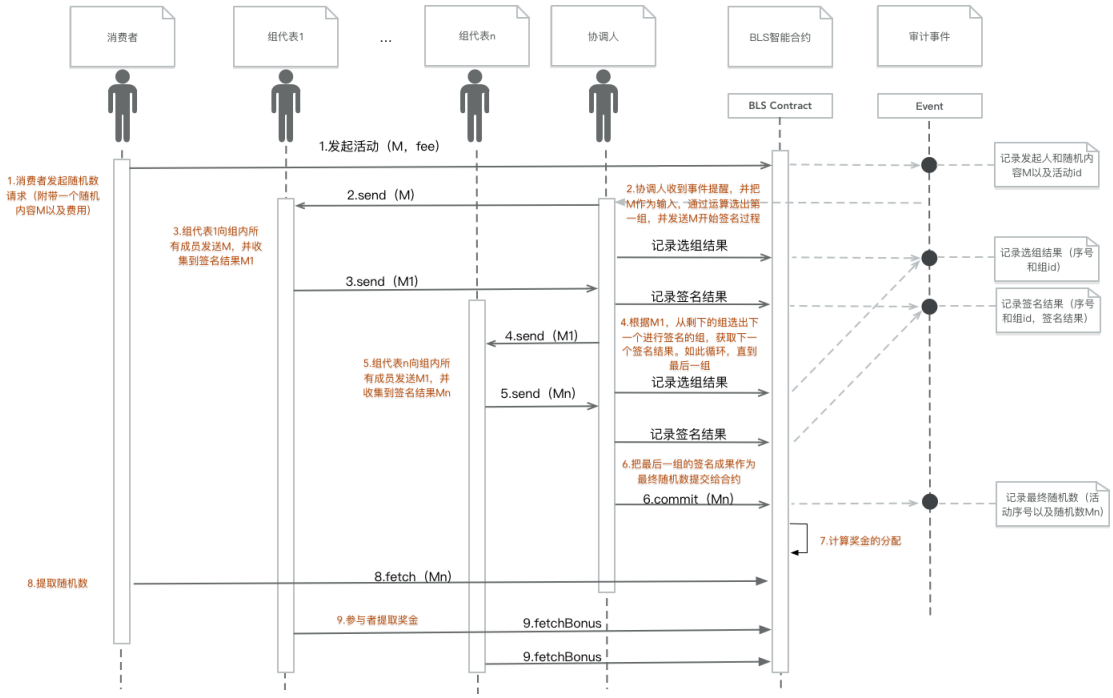
2.2.2 组内对随机内容进行签名的流程

分组结束后，当外部需要利用本组对一个指定内容进行签名时，组内成员将按照下图的方式完成签名过程。



2.2.3 BLS 产生随机数流程

了解组内初始化和签字的方式后，下图将从整体来描述多个分组如何一起配合完成随机数的产生。



3　代币层 - 代币

Randao 的代币符合以太坊的 ERC20 标准，兼容各种以太坊的钱包应用。

4　应用和生态 - 随机数接入生态

现实世界中随机数服务具有广阔的需求。如前所述，在目前的互联网上有不少提供随机数的中心化服务商，比如 random.org，NITS Random Beacon 等，他们通过各种形式的接入服务，满足不同场景的需求，从而形成相应的服务生态。而随着区块链的发展，将有更多用户和第三方业务系统使用 Randao 的去中心化的随机数协议。为了给用户带来优异的使用体验，Randao 将针对不同类型的用户提供不同的接入支持，打造服务生态。

4.1　用户分类

目前 Randao 主要从用户角色、用户专业性及使用环境三个维度对用户进行分类：

4.1.1　从用户角色进行分类

- 生产者：生产者是指参与随机数生成的用户，比如在 Commit Reveal 中参与提交随机数的用户。
- 消费者：消费者是指向 Randao 发起随机数生成申请并最终使用随机数的用户。

4.1.2 从用户专业性分类

- 开发者：开发者是指基于 Randao 服务进行二次封装然后提供自己特色业务的用户，一般是接入 Randao 的第三方服务商。
- 一般用户：这类用户不关心技术细节并希望使用 Randao 提供的开箱即用产品或服务。

4.1.3 从使用环境分类

- 链内用户：在以太坊的环境中使用 Randao 服务的用户。
- 链外用户：这类用户是指已经在传统互联网上运营了使用中心化随机数服务的实体。由于某种原因，他们不想在以太坊上重构自己的业务（即不想成为链内专业用户），但是希望能用现有互联网的方式来使用 Randao 提供的服务。

4.2 生态建设

4.2.1 针对开发者的生态建设

开发者无论是生产者还是消费者，他们的述求都是基于 Randao 的服务来开发自己的产品。为此，Randao 将创建和维护一个开发者社区，用户必须在此社区上注册成为认证的开发者才能使用 Randao 提供的各种接入 API 以及其他开发资源。此社区包含如下产品：

- 详细的 API 在线描述文档：提供 Randao API 在链内和链外的规范说明以及使用示例。
- 开发 SDK：为开发者快速开发接入 Randao 服务的开发 SDK。针对链内环境，它将是一套基于 Solidity 语言开发的调用 Randao 合约的模板；而针对链外环境，Randao 提供了桥接服务，它把 Randao 的接口封装成 Restful API，并提供多种语言版本的开发库来方便开发者调用这些 API。
- 社区：为开发者提供在线技术问答以及 Randao 服务使用讨论，推进 Randao 技术的发展和完善。

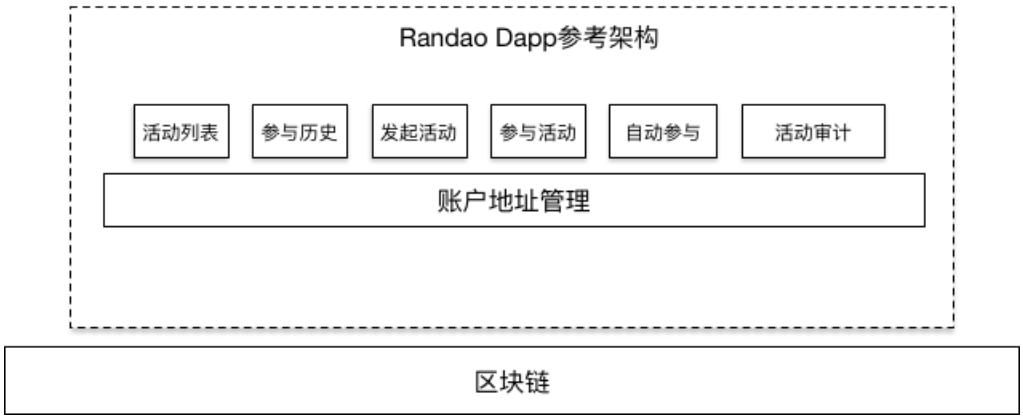
4.2.2 针对一般用户的生态建设

一般用户不会基于 Randao 进行二次开发，而是用一种直观的形式来直接或间接地使用 Randao 的服务。为此，Randao 算法将被封装到 DApp 或守护进程中，用户通过该 DApp 或守护进程可使用如下核心服务和功能：

- 作为消费者加入某个正在进行的随机数生成活动并支付费用，最终获得随机数。
- 作为消费者来发起一个生成随机数的活动并支付费用，最终获得随机数。
- 作为生产者来参与生成随机数的活动并获得收益。
- 启动生产者机器人守护进程，授权系统自动订阅新的随机数申请服务，一旦有新的申请产生，系统将自动代替用户参与随机数生成服务，无需用户干预。

- 浏览用户参与活动的历史。
- 审计指定某个已结束活动的数据。

下图为 Randao DApp 的参考架构：



第五部分　Randao 应用场景

依靠随机数（摇号抽签）来分配社会资源，已经应用到日常生活的方方面面。从幼儿园入学资格、到初高中分配学校、再到买车买房、政府招标，都依赖随机摇号抽签。人的一生，很多重大的选择，其实都是随机数帮助决定的。

这些抽签摇号场景需要随机数：

- 具有公信力，与所有参与者达成共识，最终没有获得资源的人，也认可其公平公正性。
- 满足人数要求，两人以上，都可以参与，且人数多少不会影响结果的公平性。
- 能比较快速地产生结果，不用待很长时间。

抽签摇号为了满足这些要求，也在逐渐演进。从最早的聚在一起掷骰子、抽签，到后来的通过互联网参与，由计算机生产。随机数在参与方式，生产方式，公示方式上不断的进行优化。

但在操作过程可证公平这点上，一直没有满意的解决方法。早期大家一起掷骰子，非常直观。虽然也存在暗箱操作的可能，但因为过程清晰，参与人数少，基本都是参与者自己确认结果，所以虽有瑕疵，但公信力仍然很强。但现在，因为参与者众多，不可能都到场亲眼目睹随机数产生的过程。同时，计算机程序一般通过给定的“种子”产生伪随机数，这也给暗箱操作留下了更多的空间。随着公众对随机数产生过程的了解不断深入，这种生成方式不断受到质疑。极端情况下，导致负责分配资源的主体的公信力的丧失。

Randao 作为可证公平的随机数服务，可以广泛应用于上述场景，为资源分配主体重拾公信力。下午详细叙述如何在车牌摇号、税务抽检等场景下，使用 Randao 服务。

1　车牌摇号

截至 2017 年初，北京市参与普通小客车指标摇号的有效申请人为 2,783,966 人，而有效指标仅 13,905 个，中签比例小于 1:200。网上流传各种摇号攻略，例如外地人摇号更容易中、晚点确定参与摇号更容易中，通过特殊渠道可以买号等等。这些传言无论真假，它们发生的基础，都是人们对摇号过程及结果不满或质疑时，无法通过有效的方式证实而导致的。每到摇号结果发布的时候，各个微信群里，就会晒摇号结果，有人摇了六年没有摇中。从概率的角度说，这完全是可能的。但是站在普通人的角度，是真的运气太差，还是有人从中作弊，无法证实，必然产生怨言。时间长了，谣言也就形成了，管理机构的公信力也必将受到打击。

所以，在这种随机数使用场景下，就更需要能自证公平，让所有参与者自由的查看结果产生的过程。随机数生成及发布过程不仅仅需要产生结果，更需要具备沟通的能力，与参与者达成共识，才是公平摇号的本质。

而 Randao 可证公平随机数服务，因其自身具有很强的沟通特性，可以在这个场景下发挥作用，更高效的达成共识，消除误解。所有参与摇号的人，可以通过平台提交自己的随机数，并全程观察随机数生产的过程。使用 Randao 进行摇号可设计并实现这些关键点：1. 期望对结果进行影响的人可选择参与随机数产生过程；2. 参与者必须尽到应尽的义务，违规者将受到惩罚；3. 参与过程透明；4. 结果可审计。

2　税务抽检

与社会资源分配事件相对应的是社会惩罚事件。在这类事件中，因为受随机数影响的参与者，是被惩罚者，一般不会质疑结果，并且不会公开讨论，这就给暗箱操作留下极大的操作空间。

在社会惩罚事件中，随机抽样的过程更加随意。比如地方税务所，每年要在十几万家企业里抽检，抽检谁不重要，但不抽检谁，很重要。因为是税务所内部系统，抽检结果不会公示，同时有很多代办公司宣称可以更改抽检结果，平安年检。因为这种信息不透明，不对称，导致很多企业在经营过程中心存侥幸，即使抽到了也可以找人摆平，这就造成的政府公信力的流失。

这种情况同样可以利用 Randao 的可证公平随机数服务，设计相应的操作流程，确保过程和结果的公平公正。类似税务抽检的情况还有很多，比如海关抽检、卫生抽检、食品安全抽检等。通过 Randao 可证公平随机数服务，杜绝部分人依赖抽检打击他人，谋取私利的可能，降低社会管理成本。

第六部分　Randao 经济分析

随机数广泛应用于密码学、数值计算模拟、统计研究、乐透博彩、游戏抽奖等场合，具有极高的商业价值。根据这些场景对随机数的产生速率、使用频率、随机性、安全性等方面的不同需求，Randao 可根据自身算法的特性，提供差异化的随机数服务。可以从两个方面来考虑 Randao 的商业价值。首先，Randao 自身作为可证公平的随机数服务提供方，可从随机数消费方获得与提供服务相应的收入。其次，前述收入的一部分将作为 Randao 随机数生成时的生成

者的经济激励；另一部分将作为 Randao 社区维护的经济激励。这两方面的激励确保 Randao 可持续地生成可证公平的随机数。

1 Randao 服务的商业价值

目前阶段 Randao 算法自身的特性决定了其可应用于多个不同的场景，包括但不限于快速抽签随机数服务、高额博彩随机数服务等。

1.1 快速抽签

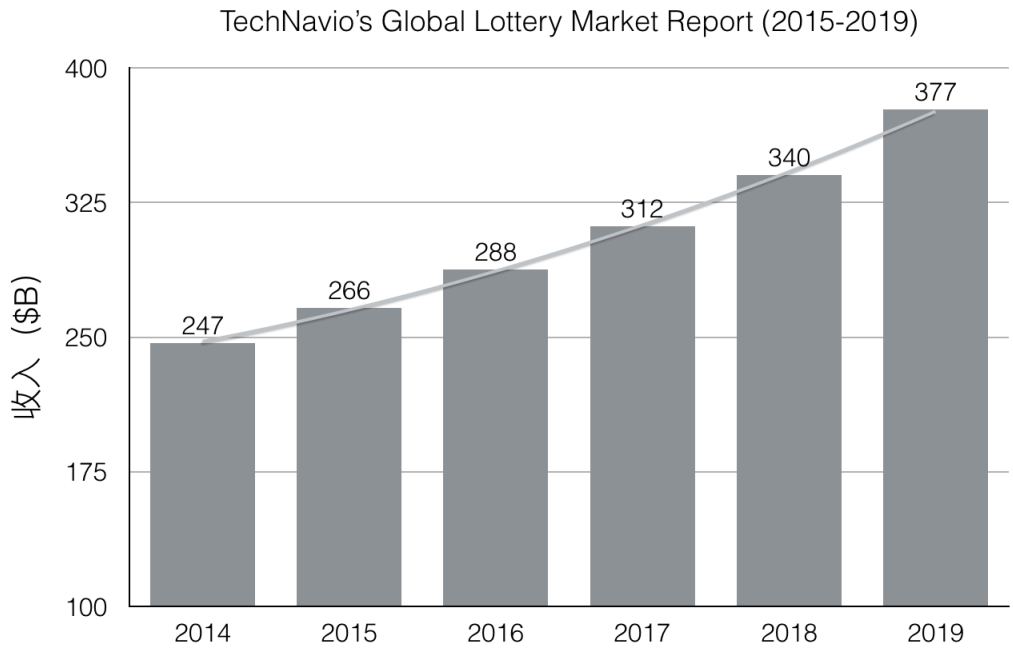
摇号抽签作为一种较为公平公正的选取方式，广泛应用于日常生活。Randao 在传统抽签基础上，进一步提供了抽签参与者最关注的可证公平性，其中的 BLS 算法非常适合为高频率、中低中奖额抽奖快速提供随机数生成服务。类比 random.org 提供的 Third-Party Draw Service 收费标准[8]，每次提供服务最低可获得 4.95 美元的收入（对应每调用的成本为 $0.0099），最高可获得超过 1149.95 美元的收入（对应每调用的成本约为 $0.00115）。根据不同随机数服务的提供及使用频率，可计算得出提供这类抽奖服务的年收入如下：

	每次调用数	每次收入	生成及使用次数	年收入
最低估计	<500	$4.95	350,400（每 90 秒一次）	~$1.73M
中值估计	500,000	$649.95	525,600（每 60 秒一次）	~$342M
较高估计	1,000,000	$1149.95	2,102,400（每 15 秒一次）	~$2.42B

1.2 高额大乐透

大乐透、六合彩等由于可玩性高、娱乐性强及头奖金额超高等特点，深受市场欢迎。根据 TechNavio 公司的报告（TechNavio's Global Lottery Market Report 2015-2019），全球今年乐透市场收入将超 3000 亿美金。

[8] https://www.random.org/draws/pricing/



根据 Business Insider 的统计[9]，加密货币的总市值约为 $100B，而广义下的货币为 $83.6T。如果这个比例同样适用于乐透市场，再根据前述 TechNavio 的报告（2017 年乐透市场收入约为 $312B），则可以推断乐透市场中的加密货币大约为 $373M。而根据另一份来自德勤公司的报告[10]，到 2025 年，至少 10% 的全球 GDP 将保存在区块链平台上，同样如果这个比例也适用于乐透市场，再考虑到乐透市场每年约 8% 的增长率，可推测到 2025 年乐透市场中约 $59.8B 将保持在区块链平台。而这其中绝大部分将以将以加密货币的方式交易形成。

然而，传统乐透摇奖一直以来没有找到很好的方式解决可证公平性问题，不得不采用公证人员现场公证、电视直播等方式博取彩民的信任。即便如此，乐透摇奖的公平公正性长期以来饱受质疑。Randao 的 Commit Reveal 算法恰好为这类低频率、高中奖额博彩提供可证公平的解决方案。下面通过两种方式预测这个市场的规模。

1.2.1 同比估算

根据世界彩票协会（World Lottery Association, WLA）报告 The WLA Global Lottery Data Compedium 2015[11]，2014 年其会员（全球大部分受监管的合法的乐透组织）基于抽奖的乐透收入约为 $159.9B，其中中国体育彩票基于抽奖的乐透收入约为 $16B。体彩全国联销的有：排列 3、排列 5、大乐透、22 选 5、七星彩和足球彩票等。体彩大乐透每周一、三、六开奖，每次都有公证公司相应的公证人员提供现场公证服务。根据公开的信息[12]，体彩公证费约为 $2.17M（1475 万人民币）。如果使用 Randao 可证公平的随机数服务替代这些公证服务，按比例可估算为受监管的合法的乐透组织提供随机数服务的市场规模约为 $21.7M。

[9] http://www.businessinsider.com/bitcoin-compared-to-all-of-the-worlds-money-2017-6

[10] https://www2.deloitte.com/lu/en/pages/technology/articles/impacts-blockchain-fund-distribution.html

[11] https://world-lotteries.org/media-news/publications/wla-compendia/2077-the-wla-global-lottery-data-compendium-2015

[12] http://www.ccgp.gov.cn/cggg/zygg/zbgg/201609/t20160908_7289856.htm

1.2.2 频率估算

同样根据前述中国体育彩票的公开信息，可计算得到每次公证服务的收入约为 $384。若 Randao 提供的可证公正性随机数服务能被更大范围的高额博彩使用，假定为世界彩票协会的每个会员组织每天提供 1 次随机数生成服务，可计算得出相应的年收入如下：

	每次收入	服务次数	年收入
估计	$384	~54,750 (~150*1*365)	~$21M

2 Randao 生产者的经济激励

一般地, 如果每期生成的随机数平均有 r 次调用, 每次调用价格是 p ETH, 则每期的收益为 rp ETH。假设每期随机数由 n 个生产者协作生成, 则消耗的成本为 $n*3*500*gasPrice+Ccost$ ($Ccost$ 是 Randao 智能合约 C 内部消耗的 gas 价值，包括运算以及存储等)。由此，每期每个生产者的收益为 $(rp-1500*n*gasPrice-Ccost)/n$。当前的 $gasPrice$ 是 25 gwei，估计 C 会消耗 $1500n$ gas，则生产者每期人均净收益预估为 $rp/n-7.5*10^{-5}$ ETH。

2.1 基于目标收益的服务定价分析

随机数生成的频次非常高，以本白皮书商业价值部分快速抽签中的最低生成频率为例 (每 90 秒一次)，即便每次评价收益率仅为 0.0001%，每年的线性收益率仍然可达 $0.000001*40*24*365=35.04\%$。

假设每期随机数生成有 10 个生产者参与，每期保证金为 1000ETH, 则在保证收益率大于 0.0001% 的情况下，根据收益预估公式 $rp/n-7.5*10^{-5}$，需要的最小收入 rp 是 0.01075ETH。这样如果随机数只有 100 次调用, 每次调用价格约为 0.0001075ETH, 如果是 10000 次调用, 每次调用价格只需 0.000001075ETH。

2.2 基于调用人次的收益率分析

按当前 (2017 年 9 月) 市场实际价格：$ethPrice$ = ~$300。再假定每次有 20 个生产者参与生成随机数、每个地址押金为 10000 ETH、每小时参数 5 次随机数生成。若采用 random.org 的 Third-Party Draw Service 价格表为依据定价，则可分析得出随机数生成的提供者收益率如下：

调用人次	服务收入	合约成本	每次收益	每次收益率	年收益率（线性）
<=500	$4.95	0.45	4.5	0.000008%	0.3285%
1,000	$8.95	0.45	8.5	0.000014%	0.6205%
2,000	$16.95	0.45	16.5	0.000028%	1.2045%
3,000	$22.95	0.45	22.5	0.000038%	1.6425%
5,000	$34.95	0.45	34.5	0.000058%	2.5185%
10,000	$54.95	0.45	54.5	0.000091%	3.9785%
25,000	$99.95	0.45	99.5	0.000166%	7.2635%
50,000	$174.95	0.45	174.5	0.000291%	12.7385%
80,000	$219.95	0.45	219.5	0.000366%	16.0235%
100,000	$249.95	0.45	249.5	0.000416%	18.2135%
250,000	$399.95	0.45	399.5	0.000666%	29.1635%
500,000	$649.95	0.45	649.5	0.001083%	47.4135%
750,000	$899.95	0.45	899.5	0.001499%	65.6635%
1,000,000	$1149.95	0.45	1149.5	0.001916%	83.9135%

3 Randao 的经济约束

Randao 在区块链系统中是作为一个基础设施存在，供其他合约调用。不同目的的合约对随机数的要求不同：有些需要高安全性，例如大乐透的开奖结果；有些需要稳定及时，每次调用都立刻有返回，例如不涉及利益的普通合约；还有些需要回调，他希望某期随机数生成后，随机数生成合约能够主动推送信息。

显然不可能通过一个合约满足各种场景下的不同需求，所以会创建很多个不同初始参数的 DAO Contract，但基本规则不变。比如对于高安全性的需求，会大幅提高第一阶段保证金的数量。这样，试图通过不提交 s 而导致随机数生成失败的成本大幅提高。而对于低要求的随机数合约，则可以对最小参与人数，保证金放低要求。

我们以一个赌奇偶的应用来说明，如何通过调整不同参数，达到期望的高安全性：即使得违约成本高于预期收益。假设该赌奇偶的应用的赌本是 1000 ETH，调用随机数合约 C_1，如果 C_1 当期随机数生成失败，则等待 C_1 的下一个随机数，一直到有随机数产生为止。现在我们来构造随机数合约 C_1，C_1 要求的保证金是 2000 ETH。如果参与奇偶赌局的赌徒 G 同时参与了 C_1，则当他发现自己处于不利局面的时候，选择不提交 s，从而使得随机数生成失败。此时他在 C_1 上损失了 2000 ETH，仅仅获得了 1000 ETH 的期望收益，是非常不划算的行为。但 G 可以通过一些手段降低在 C_1 上的损失，比如他同时用 2 个帐号参与了 C_1，发送了 2 个 $sha3(s)$，如果局面不利的话，则只让一个帐号不提交 s，如果当期只有 1 个除 G 之外的其他参与者参与了 C_1，则 G 的损失只有 1000 ETH，而在奇偶赌局的期望收益是 1000 ETH，是一次值得尝试的攻击。

一个应对方案是直接罚没，罚没的金额不作为奖励返回给参与者，这样只需要一个保证金为 1000 ETH 的随机数合约就可以满足该奇偶赌局的要求。除了罚没之外，另外一个方案通过引入额外的制度来杜绝这种攻击：Randao 会员。要成为会员必须先缴纳会费，任何人只要缴纳

了会费都可以成为会员。根据缴纳会费多少的不同，会有不同的会员等级。会员不属于某一个合约，只是有资格参与一些随机数合约的身份证明。如果该会员在任何一个合约中违约，则缴纳的会员费会被没收。现在我们给合约 C_1 增加一个额外的约定，只接受一定等级之上的会员（会员费超过 1000 ETH）提交的随机数。这样就可以确保任何人都没有动机进行攻击。

第七部分　Randao 开发计划

1　2017 年 Q4

- Commit Reveal 随机数生成的智能合约上线
- Web 版 DApp 上线

2　2018 年 Q1

- BLS 随机数生成的智能合约上线
- 机器人服务功能上线

3　2018 年 Q2

- DApp 上线

4　2018 年 Q3

- 开发者社区上线