Algorand 共识算法

2018 年是公链技术爆发的一年，诞生了诸多从共识方面创新的项目。由于目前人们普遍认为存在区块链“ 不可能三角 ”，这些共识往往要在性能、安全、去中心化、激励机制中做出取舍。例如， EOS 达成每秒数千次交易速度是以牺牲去中心化为前提的。然而“ 不可能三角 ”从来没有像 FLP、CAP 这些分布式系统定理一样得到严谨的数学证明， 因此有些人认为打破 “ 不可能三角 ”是有可能的。可验证随机函数 VRF 被认为是一个有前景的方向。本次为大家带来最近热度非常高的 Algorand 项目的分析。

1、 Algorand 共识算法简介

Algorand 共识算法是图灵奖获得者 Silvio Micali 在 2017 年底提出。 Michali 是 MIT 的教授，是一位密码学家和计算机理论学家，在伪随机数以及零知识证明领域很有名。

Algorand 共识算法的论文的下载地址：

https://people.csail.mit.edu/nickolai/papers/gilad-algorand.pdf

Algorand 采用了 VRF 函数，并结合账户的余额比例， 随机确定区块生成以及投票人角色。

所谓 VRF（Verifiable Random Function ）就是可验证随机函数。

随机数对于区块链技术来说很关键。　　本质上，分布式账本的核心问题就是随机选择出块人的问题，　这个随机性要能被全网确认，　并且不能被操控，　也不能被预测，　否则恶意节点通过操控这个随机数就可以操控长链，从而实现双花攻击。

PoW 的方案是让大家进行算力竞赛，设置一个计算哈希的难题，谁先算出来谁赢，算力高的赢的概率高，　算力低的赢的概率低，　以这样的方式保证胜出者是随机的。投入的算力能够体现在哈希值上，　　这样全网能够验证，并选择包含最多算力的那条链。恶意节点只能通过提升自己的算力来增加攻击成功的概率。

PoS 的方案是选举，　大家不用浪费电力去进行算力竞赛，　　而是文明一点，　随机选举一个节点来出块，并且被选中的概率和它拥有的份额相关。　　如果“ 随机 ”这一步没有问题的话，　恶意节点只能通过增加自己的份额，　增加自己被选中的概率，从而增加双花攻击的成功概率。　　这里有一点比 PoW 的方案要好就是，要实现攻击，先得成为持币大户，　如果攻击成功币价大跌，　攻击者也会承受最大的损失。

那么接下来的核心问题就是，　这个不能被操控不能被预测的随机数从哪来。　传统地 PoS 方案尝试从链上现有的数据入手，比如使用上一个区块的哈希值，上一个区块的时间戳等等来作为随机数的来源，但这些会带来额外的安全风险。　　因为区块本身的信息就是节点写进去的，　然后又要根据里面的信息来选举后续的出块者，存在循环论证的嫌疑，安全性不会太好。　　这也是传统地认为 PoS 方案不如 PoW 可靠的部分原因。

Algorand 提出的 VRF 能够由私钥 (SK) 以及讯息 (X) 产生一组可验证的伪随机 (pseudorandom) 数 Y 以及证明 ?。任何人都可以透过 Verify 函数来检验这个

随机字串是否真的是该公钥对应私钥持有者，依照规定使用 Evaluate 函数所产生，而不是自己乱掰的。更详细一点的 VRF 三个函示描述如下：

?Keygen(r) → (VK, SK). On a random input, the key generation algorithm produces a verification key VK and a secret key SK pair. ?Evaluate(SK, X) → (Y, ?) . The evaluation algorithm takes as input the secret key SK , a message X and produces a pseudorandom output string Yand a proof ? . ?Verify(VK, X, Y, ?) → 0/1 . The verification algorithm takes as input the verification key VK , the message X , the output Y, and the proof ? . It outputs 1 if and only if it verifies that Y is the output produced by the evaluation algorithm on inputs SK and X .

为什么我们需要这么一个大家自己产生，却又要可以被验证的随机字串产生器呢？这是因为在 Algorand 的拜占庭演算法中， 虽然也存在着每一轮不同的区块生产者 (称为 Leader) 以及验证委员会 (Committee, Verifiers) ，但并不是用「公开选举」的方式来选的， 而是透过每个使用者自己对奖的方式来看看自己是不是下一轮的 Leader 。

algorand 就是通过随机算法从一群大范围的验证者中选取一部分验证者运行 BFT 算法(Micali 教授实现的 BA* 算法)，从而达到提高共识的效果。

无论是在何种 BFT 的共识机制中， 都是由 Leader 以及 Committee 来完成区块的发布以及共识决议。 例如 EOS 的 dPoS BFT 是固定 21 个 BP 轮流担任 Leader 以及投票者、 Zilliqa 透过 PoW 加入 Committee 进行 PBFT 共识算法。这些比

较直观的拜占庭共识演算法都有一个共同特征，　就是大家都可以看到下一个区块的 Leader 是谁，以及负责协议共识的　Committee　是谁。这造成了一个可能的风险，就是这些区块生产者以及　Committee　成员容易成为 DDOS 或是贿赂的目标。

Algorand　为了解决这种潜在的风险，利用　VRF 来掩盖 Leader Selection　的步骤。可以想像成：一般的　BFT 在每一轮开始时公平公开选出　Leader 以及 Committee　，Algorand　则是像在每一轮开始时公布中奖号码，　每个使用者都可以自己拿自己的票根对奖，中奖的人即可成为下一轮的　Leader(或是 Committee Verifier)　，但在中奖者自己表明身分前，没有人知道谁中奖，也就是没有人能预测下一轮的　Leader 以及 Committee　。当然中奖与否并不是口说无凭，中奖者需要出示中奖证明，　而这个证明是大家都可以验证的，　这正是我们刚刚说到的 VRF。

2、 Algorand　共识算法缺陷

(1) 现实环境的随机选择的空间并不大。

VRF 是可以提供了公平且不容易收到伪造和攻击的委员会随机选择方式，但是任何随机数的生成必须依赖大的种子集合才可以有作用，在　VRF 中假设 80% 节

点是诚实的， Committee 需要 2000 个成员才够大，现实情况是不太可能有这么多成员的。

(2)完全没考虑网络延迟情况。

VRF Committee 集合选举时依赖数量庞大的主机通讯的，主机之间相互沟通造成的延迟，必然大大拖慢整个系统的处理速度。

(3)没考虑节点的动态加入和退出情况。

Algorand 的下一个区块的发布者是从 k 个区块之前的所有参与者 （在 k 区块之前的某段链上发过交易的节点） 里选。于是，恶意节点想影响下个区块的发布者，他得影响 k 个区块才行，当 k 很大的时候，这个影响也是微乎其微。于是，Algorand 得到了一个无偏向的随机数产生器。 不过，这个做法有一个问题——k 区块之前的节点，有可能已经不在线了。而对于这一点，虽然 Micali 做出了解释，但是个人觉得并不符合实际情况。

(4)签名数据庞大，造成存储浪费并影响性能。

Algorand 使用 VRF 来确定提案组与验证组，这个方式充分发挥了 VRF 的可验证性优势，且后验优势使得 Algorand 的共识体系更安全。但是， Algorand 进入验证阶段， 采用的是一种可扩展的拜占庭容错算法， 即 BA* 算法， 参与节点通过 VRF 秘密抽签选出。这一设计使 Algorand 在验证前必须等待凭证（ VRF prove ）到来，才能知晓参与节点。 而且， 由于使用了可扩展的拜占庭容错算法，使得 Algorand 的验证组规模必须比较大 (2000 ～4000 人)，这将导致签名数据

异常庞大。根据我们的估算，在平均每组 3000 个验证节点的规模下，每组的签名数据长达 126KB，加上其它信息，通知信息约 300K，每块的签名数据可达 2000*64*12=1M （共 12 组，每组 3000 人，至少 2/3 达成共识。ed25519 签名数据长度是 64。），远超一般门限签名几十个字节， 严重浪费存储和容量 （因每块存储的交易量将被占用，不存储签名又会影响安全）， 不仅造成存储浪费，而且更影响性能。

(5) 无法构建很好的激励机制

在 POW 中，提案者得到提案权需要预先付出算力成本， 若其提案区块有问题（交易双花），则该提案区块在全网其他节点验证必将失败， 从而不但没有铸块收益，还付出了算力成本。

Algorand 协议并没有设计经济激励机制， Micali 教授曾回应 ” Algorand 协议只需要进行平凡的计算，因此不需要激励 ”。在没有经济激励机制下，高性能带宽和服务器必然不愿意参与 （因为它本身要消耗费用） ，整个网络会遇到网络本身无法解决的困难。

(6) 存在潜在的安全问题

网络用户必须连续访问其私钥，以确定其在每一轮中的 VRF 状态（即验证者、提议者，或者两者都不是）。

一般认为， 对于那些将大量资产存储在区块链上的个人， 为了防止攻击， 他们应该把私钥以冷存储的方式进行保存。 而持续的验证 （需要经常签名） 会需要高频

率地动用私钥，从而增加被攻击的风险。 这显然将导致网络中很多诚实的个体 （出于安全的考虑）会避免参与验证过程，从而造成区块链缺乏活力的问题。

(7) 买断问题

在区块链的婴儿期， 系统的通证价值通常较低， 其市值也是处在相对较低的水平。Token 的发行往往要经过私募 --> 基石--> 公募 等逐步分散的过程， 因此很长一段时间里币是集中在少数人手里的， 因此任何 POS 共识都面临着 EOS 类似的中心化的问题。

(8) 没有惩罚问题

Algorand 所存在的另一个问题是，没有办法识别“ 离线验证者 ”并惩罚它们。因此，在没有惩罚措施来防止无效的情况下， 没有经济激励就是一个问题， 很多人会选择不为共识做贡献，因此离开这个网络。假设网络中只有 10% 的诚信节点在不断地进行验证， 而其余节点是离线的状态， 与此同时， 恶意的节点选择保持在线，那其就很容易超过在线委员会节点。这使得恶意节点更容易控制共识。

总的来说， Algorand 的 VRF 和加密抽签后验性给出了一个解决 “ 三角悖论 ”的很好设计思想， 但其在验证环节的设计更偏单纯的学术化理想化， 导致其对网络流量、有效通讯数据等实际工程落地思考不够， 严重影响了公链运行性能、 节点网络规模、账本存储容量和去中心化程度。