SPOS共识 2023-9-7

共识模块是实现区块链共识机制的模块，共识机制是区块链技术的重要组件。区块链共识机制的目标是使所有的诚实节点保存一致的区块链视图，通俗来说，共识机制在区块链网络内起到决定谁负责生成新区块以及维护区块链统一的作用。采用了创新的SPOS共识，即 Safe POS共识机制。

SPOS的创新及应用主要体现在两个方面。
通过Ticket选票机制实现POS共识。
在区块产生的过程中生成公平的随机数。
#通过选票机制实现共识
通过**Ticket**实现**POS**的安全挖矿逻辑。用户使用钱包账户中代币余额购票，获得挖矿权（目前3000个代币对应1票），一票对应一个唯一的Ticket ID；每个区块只会由一张票挖到，实际的挖矿几率各票均分（如全网有N张票，则每张票挖到矿的几率为1/N）。
选票挖矿流程如下所示：
在钱包中定期检查账户中的代币余额来购票，当满足购票条件后构造一条买票交易发往区块链。
SPOS共识会一直尝试使用本地持有的票去打包区块，一旦打包成功，即表示对应的Ticket持有人挖矿成功，并获得对应的区块奖励。
智能合约会把地址对应的Ticket信息写入到区块链数据库，每一张Ticket都对应有一个唯一的Ticket ID，也会有一条数据记录在数据库。
注意：试图分叉的恶意节点，或者任何系统能检测到的恶意行为，都可能会被惩罚，每次惩罚会损失20%的资产。挖矿必须以基金会发布的标准钱包进行，篡改挖矿行为，如果被系统自动判定为恶意，都会给矿工造成巨大的损失。
SPOS共识特点：

相较与**POW、POC**等共识机制免去了共识过程中不必要的资源浪费，如存储浪费、算力浪费等。
智能合约执行1/N的选票逻辑确保了出块的公平公正。
惩罚机制的设计，确保矿工正向运作，保证主网运行安全。

#在区块产生过程生成公平随机数
在游戏等应用场景，为了体现公平性，区块链上需要一个不能被预测的随机数，来保证公平公正。同样的，比特元在区块产生的过程能够产生公平的随机数，保证链上应用随机数调用的安全。
目前的区块链大体有以下实现方案：
合约中调用外部中心化的随机数发生器获取随机数；
使用区块hash中的某些值作为随机数。
但是这两种方案都有非常明显的弊端，原因在于：
区块链多节点之间智能合约执行结果是要求强一致的，如果合约从外部读取数据，是很有可能获取到不同结果的（比如网络原因导致有的节点读取正常，有的返回错误），进而导致分叉。
区块的哈希可以被控制，导致随机数被控制。由于一些区块链系统没有提供很好的随机数算法，所以很多DApp开发者会自己封装自认为完美的随机数算法，导致随机数被预知。
针对以上两个弊端，实现了随机数生成的优化：在原来的随机数基础上引入了VRF（Verifiable Random Functions）可验证随机函数逻辑，使得随机性进一步增强。
###随机数生成的实现逻辑

用户通过使用钱包账户中的代币购票（Ticket），3000代币对应一票。
☒钱包同时生成一个**randNum**，哈希过后再结合钱包

挖矿地址的私钥，票对应的**index**（一次可以买多张票）等元素再做两次哈希，得到一个公开哈希参数pubHash：**pubHash**=hash(hash(privateKey:index:hash(randNum)))。

新购买的票中包含pubHash和randNum，并将其存入区块链，Ticket需要12小时的成熟期，过了12小时才可以参与挖矿。

共识算法从区块链中找到已经成熟的Ticket开始打包。☒由于共识打包区块操作只在节点本地执行，所以它可以读取本地存储的私钥，算出一个私密哈希（privHash）并将其放入挖矿交易：privHash = hash(privateKey:index:hash(randNum))

智能合约收到挖矿交易，对比hash(privHash)和pubHash的值，两者一致挖矿交易成功，对应的节点获得挖矿奖励，否则挖矿交易执行失败。

关键术语
VRF：Verifiable Random Functions 可验证随机函数
SK, PK: VRF 中使用的公私钥对，SK 为私钥，PK 为公钥
M：输入数据
R：VRF 哈希输出
P：VRF 证明
Prover：证明者，拥有 VRF 的公私钥（PK 和 SK）
Verifier：验证者，拥有 VRF 中的公钥（PK）
函数组
生成随机数中包括四个函数，主要分为两类：生成函数和验证函数。

生成函数：

R=VRF_Hash(SK,M)
P=VRF_Proof(SK,M)
验证函数：
R=VRF_P2H §
VRF_Verify(PK,M,P)
VRF 使用流程

证明者生成一对密钥（PK 和 SK）。

证明者计算 R 和 P（R = VRF_Hash（SK, M），P = VRF_Proof（SK, M））。

证明者把 R, P, PK, M 递交给验证者。

验证者计算，满足 **VRF_P2H**（P) = R 并且 VRF_Verify(PK, M, P) = True☒满足条件即验证通过，否则验证不通过。

BTY 使用 VRF 实现随机性的流程

SK 对应挖矿节点账户的私钥，PK 对应挖矿节点账户的公钥。

在2019年升级分叉的那个高度，挖矿节点先读取前一个区块中存储的privHash作为输入的M值，并通过**VRF**生成函数分别计算出R和P的值。

挖矿节点将 R 和 P 写入挖矿交易中，打包进区块并广播。

所有节点收到区块，在智能合约中通过 VRF 验证函数来验证正确性。满足VRF_P2H（P) = R 并且 VRF_Verify(PK, M, P) = True 即验证通过，否则验证不通。

所有节点收到区块，在智能合约中通过 VRF 验证函数来验证正确性。满足VRF_P2H (P) = R 并且 VRF_Verify(PK, M, P) = True 即验证通过，否则验证不通。

再往后的区块，读取前一个区块中存储的 R 作为输入 M 来计算 R 和 P。

可以看出，当前区块引入的随机数，都需要依赖于上一个区块的随机值 R，随机值更不容易被控制，随机性大大增强。因此，SPOS 共识的实现结合了随机数，由于一般情况下是无法预测其它节点的共识信息，所以也无法获取到它的共识随机数。并且系统设定私密哈希（privHash）不能提前泄露，就算有恶意矿工自己提前暴露，它对应的票也会被作废，同时本金会被冻结较长时间（2天以上）。再加上系统设定票需要经过 12 小时的成熟期后才可以参与挖矿，以及在共识逻辑中引入 VRF 可验证随机函数的实现，每一个区块的随机值都依赖于上一个区块的随机数 R，随机性进一步增强。
以上这些条件组合起来，系统的随机数几乎是无法被操控的。这样当开发者实现的 DApp 中需要保证公平随机时，就可以直接使用系统提供的这个安全的随机数了。