什么是VRF？VRF(Verifiable Random Function）: 可验证随机函数。

要理解VRF的工作原理，前提是理解哈希函数，例：SHA256，SHA3等，这里不做详述，读者可自行百度了解。先理解一下这里说的“随机”是什么意思：一个理想的哈希函数，其值域应该是离散的、均匀分布的，给定不同的输入值，其输出值应该没有规律，随机的分布在值域区间内。再看一个简单的哈希函数变种，即结合了密钥secret的哈希函数，比如result = SHA256（secret，info），那么要得到结果result，仅仅拥有info是不够的，必须要知道secret才能计算出来，或者说我们已经拥有了结果result和info，但是必须知道secret才能验证info和result是否是匹配，这就是带密钥的哈希函数。此时引出另一个问题，有没有可能不通过密匙secret，也能验证info和result是匹配的？于是就有了可验证的随机函数即：VRF。实现原理：结合了非对称加密技术的哈希函数，例如：result = VRF\_Hash(SK, info), SK是私钥，不公开，和SK对应的公钥PK，需要发送给验证者。

具体操作流程如下：

1. 证明者生成一对密钥，SK、PK；

2、证明者计算result = VRF\_Hash（SK，info）；

3、证明者计算proof = VRF\_Proof（SK，info）；

4、证明者把result和proof递交给验证者；

5、 验证者计算result = VRF\_P2H（proof）是否成立，若成立，继续下面的步骤，否则中止；

6、 证明者把PK，info递交给验证者；

7、 验证者计算True/False = VRF\_Verify(PK, info, proof) ，True表示验证通过，False表示验证未通过。所谓的验证通过，就是指proof是否是通过info生成的，通过proof是否可以计算出result，从而推导出info和result是否对应匹配、证明者给出的材料是否有问题。

在整个操作流程中，证明者始终没有出示自己的私钥SK，验证者却可以推导出info和result是否对应匹配，这就是VRF的妙用。

采用VRF的拜占庭容错算法VRF最大的作用生成一个可以被他人验证的随机数，并且这个随机数是他人生成不了的，也就是防碰撞和防篡改的，接下来我们看看VRF如何应用到拜占庭容错算法当中。先给出ontology项目的具体实现，稍后会分析这种方法的潜在问题。Ontology VBFT算法实现流程ontology VBFT算法是基于参与的共识节点进行的，共识节点是经过认证的具有一定ont token的节点，类似于EOS的超级节点，不是网络中的所有节点都能够参与共识。

1. 基于当前块的信息，算出新的VrfValue，所需信息如下：

a. 新的块号：当前块号 + 1

b. 产生当前块的节点id

c. 当前块的块头根hash

d. 当前块的VrfValue根据上述四个信息，采用SHA512算法，算出新的VrfValue，参考代码：ontology/consensus/vbft/node\_utils.go 方法：getParticipantSelectionSeed

2. 每个共识节点根据新的VrfValue，以及所有共识节点的PosTable，无需网络通信，就可以算出统一的产生下一块的提案节点，验证节点，确认节点，其中PosTable存放了所有共识节点的token 权重。参考代码：ontology/consensus/vbft/node\_utils.go 方法：buildParticipantConfig

3. 从VRF得到的多个提案节点，将独立提出备选区块。提案节点产生的块头会包含一个可验证的随机值：VrfValue，如果当前提案节点产生的块被选中作为最终块，则当前提案节点产生的VrfValue会用于生成下次共识节点列表。

4.从VRF得到的多个验证节点，将从网络中收集备选的区块，进行验证，然后对最高优先级的备选区块进行投票。验证节点会对提案节点产生的VrfValue进行验证，确认其合法性。

5.从VRF得到的多个确认节点，对上述验证节点的投票结果进行统计验证，并确定出最终的共识结果。

6.所有节点都将接收确认节点的共识结果，并在一轮共识确认后开启新的共识。

Ontology VBFT分叉选择由于同时由多个节点出块以及恶意节点的攻击，那么有可能出现分叉的问题。每个区块头都包含了生成当前块的节点token 权重，那么以权重最长的链作为主链，类似于比特币以算力最长链为主链。由于每次出块都会随机产生出块节点，因此恶意节点很难有机会一直进行出块，因此恶意节点导致的分叉，将会很快消亡。Ontology VBFT共识节点表的更新为维护Ontology共识网络的网络质量，Ontology共识管理合约将定期自动更新共识网络中的节点列表。在发生网络风险时，共识管理合约也支持通过基于Stake的投票，强制更新共识网络中的节点列表。一个新的节点在获得更多Stake，并且确认满足共识网络的节点性能需求后，将在下一次共识网络更新时被加入共识网络。共识网络自动更新的时间是以区块为单位。每一次更新的共识网络在完成给定数目的区块共识后，下一个区块的备选提案节点必

VRF提供了一个随机数据生成方法，而且这个随机过程和私钥有关，并可以通过公钥被验证。

Algorand的共识分为两个步骤：1）随机选举区块生成者生成区块 2）形成共识（也就是论文中的BA\*算法）。

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。

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。

这个算法每一轮的总流程，分为两个大部分——块提议和BA⋆。块提议中首先在普通用户中通过加密抽签选择出这一轮的委员会成员（委员会成员会在每一轮通讯进行替换），然后在委员会成员中再次通过加密抽签选择出这一轮的提议者，每个提议者提议一个块。然后连同哈希和证明（一种随机号码，数字签名可以很容易对其进行验证）一起传播到网络上。之后，这一轮的所有委员会成员接到消息后，通过BA⋆协议对块达成共识（暂定或最终共识）。如果这一轮是暂定共识，则只有在当后续块（后几轮的块）有达成最终共识的情况下才能确认前几轮的暂定共识块。

Gossip协议

Gossip是分布式系统中被广泛使用的协议，其主要用于实现分布式节点或者进程之间的信息交换。Gossip协议同时满足应用层多播协议所要求的低负载，高可靠和可扩展性的要求。由于其简单而易于实现，当前很多系统（例如Amazon S3，Usenet NNTP等）选择基于Gossip协议以实现应用层多播的功能。

什么是Gossip协议

Gossip Protocol利用一种随机的方式将信息散播到整个网络中。正如Gossip本身的含义一样，Gossip协议的工作流程即类似于绯闻的传播，或者流行病的传播。具体而言Gossip Protocol可以分为Push-based和Pull-based两种。Push-based Gossip Protocol的具体工作流程如下：

网络中的某个节点随机的选择其他b个节点作为传输对象。

该节点向其选中的b个节点传输相应的信息

接收到信息的节点重复完成相同的工作