Eliminating the Security-vs-Scalability Dilemma: Randomized-Committee Consensus Protocols

by Oded Noam, Chief Architect, Orbs

Executive Summary

Current-generation blockchain protocols can determine their level of decentralization on an security-performance spectrum. Any choice determines both the level of security the network enjoys, where a larger number of independent validators yields higher security, and the throughput of the network, where a lower number of validators yields higher throughput. The result of this conflict is that networks need to compromise on both aspects, making them sub par in either or both. A new family of consensus protocols enable the maximization of both security and throughput by employing novel cryptographic techniques for trustlessly generating random seeds in a group, without allowing any party to manipulate or predict the randomization results.

The Security-vs-Scalability Dilemma

Consensus protocols are at the core of the blockchain structure, as these are the rules that regulate the security of the system, and maintain the integrity of its purpose and design. All blockchain platforms use consensus protocols in one form or another, and while there are many different algorithms for different protocols, they are all designed to provide the same result; to provide a wide agreement on data which will be secure from malicious manipulation.

The decentralized environment is secured by the number of participants working together to authorize data and provide consent. The common assumption is that the larger the group that makes up the decentralized network, the more complex the network is and the more impervious the system becomes to a focused attack. Essentially, the assumption states that the larger the quorum of consensus holders, the more difficult it becomes for an adversary

to perform coercion, bribery, hacking, denial of service, infiltration or breaking into the system. Based on this assumption of scale, emphasis has been placed on a large and diverse quorum as the ideal way to sufficiently secure the network.

However, increasing the size of the required quorum comes at a cost: reaching consensus in large groups takes more time and requires more communications bandwidth than in small groups. The bandwidth requirements of consensus protocols varies but is at least proportional to the number of participants, and many grow in proportion to its square.

With both security and bandwidth performance acting as barriers to the growth of a blockchain system, and the two being in conflict with each other, users have become accustomed to making compromises on both. But this compromise may be avoidable.

A new family of consensus protocols has been developed that provides an efficient, optimized solution to the issue of scale integrity. These new protocols provide networks with truly optimized performance, without the need for compromising on security and throughput.

Randomized Committee Protocols

Randomizing the consensus quorum members from within a set of verifiers, provides the network with a fully operable consensus protocol set that disconnects the scale of the system from the speed with which it verifies transactions. Essentially, such protocols eliminate the necessity for compromise, and offer a secure system with the benefits of a large network at the speed of a small network. These protocols include Algorand, Dfinity, Helix and more. Security is maintained by ensuring the selection of verifiers is unpredictable, and thus an attacker is unable to take advantage of the small size of the verifier quorum. By maintaining the requirement to attack a large number of verifiers, the cost of the attack remains high.

For example, we can analyze the costs of attacking a network with 1000 verifiers that uses a consensus protocol that is tolerant to ⅓ of the validators being rogue. Without randomization, attackers need to hack into 667 nodes to take over the consensus. With randomization, if the attacker hacked into 100 nodes they need to see 57 billion blocks before they have 50% chance of maliciously manipulating a single block. With more nodes compromised, the probability of success rises: hacking into 200 nodes, the attacker will need to see 1.9 million blocks before they have 50% chance of success, and with 300 they only need 3,600 blocks. This gives very good level of security compared to other fast protocols: hacking into 200 nodes is significantly harder than hacking into 15, which is what the attacker needs in case of non-randomized fast protocols that have a consensus committee of 21.

Yet, high-enough stakes may make it worthwhile for an attacker to hack into hundreds of nodes. Their strategy should then be to remain dormant until a quorum they control is randomized. As we saw in the previous example, by compromising 300 nodes the attacker only needs to wait 3,600 blocks for a 50% chance of an opportunity to embezzle from the system. To prevent that, protocols can require multiple steps in the confirmation of every transaction to be carried out by subsequent quorums. In such settings, the dormant attacker knows when she can manipulate a single block, but needs to bet on whether she will be able to manipulate the subsequent block. Such attacks may be easier to notice when they're unsuccessful. Collaterals can then be used to make sure an attack cannot be profitable: if, in the 2nd step of validation, the committee discovers its predecessor created an invalid block, it can forfeit the collateral put in place by the previous committee members.

When we look back at the case of 1000 verifiers and a protocol that is tolerant to ⅓ of them being rogue, and requires 2 consecutive committees for validation. An attacker that controls 300 nodes can only attack a block with a probability of around 1:10,000. If she cannot predict the election of the next committee, the probability of successfully attacking the subsequent block is, again, 1: 10,000. So even an attacker who waited for an opportunity to attack a block will lose collateral 9,999 times out of 10,000. Assuming the attack could pocket assets worth $1m, a collateral greater than $100 would make the attack unprofitable. Any additional committee added to the validation process adds to this multiplier's exponent, making a collateral of $100 too high to attack even for a bounty of $10b.

This method enables the consensus protocol to work with a small group of verifiers similar to protocols that enjoy fast and efficient consensus, but maintains the property of fully decentralized protocols in that its level of security increases as the network grows.

The Technology Gap

Randomized committee consensus is a cryptographic challenge: a decentralized group of validators needs to collectively randomize a committee. The protocol must ensure it is impossible to predict the result before it is committed, and then that nobody is able to manipulate it. Specifically, this should include simple manipulations, such as (intentional) failure to agree on the selection of the next committee in case the attacker isn't content with the random result.

The key to solving these challenges is *unique threshold signatures*. A digital signature is, essentially, a very large number. A protocol can determine a simple algorithm to determine random processes (such as election of random committees) using that large number as a "seed". Unique Threshold Signatures are a family of digital signature schemes, in which we can assure everyone gets exactly the same signature if they signed the same document, yet nobody can produce that signature on her own. Only when enough participants calculate the signature together will they get that number. By calculating the signature on a piece of data

that is not known in advance (such as the hash of the previous block), we can assure the value of the signature cannot be known by anyone before its due time.

Besides the implementation challenges of employing threshold signatures securely in consensus algorithms, another challenge arises - generation of private keys by validators. Traditional consensus algorithms allow validator keys to be generated independently. When a new validator wants to join, they would privately generate their own key and can immediately participate in consensus. Threshold signatures bring forth a new technical complexity - validator keys are dependent on one another and must be generated together. Within a centralized system, a governing entity could issue new keys for all participants but this is naturally impossible in a decentralized one. This challenge is overcome by a new family of decentralized key generation protocols.

Orbs and the Helix Consensus Protocols Family

Orbs is developing consensus protocols based on randomized committee consensus, under the name Helix. All of the Orbs validators participate in creating a random beacon, by propagating their shares of a BLS signature of the previous random beacon result. Propagation of the signatures uses signature aggregation to put an upper bound on the amount of traffic each node needs to send out. In total, the amount of network traffic generated by the random beacon is proportional to the number of participants. Next, the random seeds generated are used to determine membership in consensus committees for each of the virtual chains operated inside the Orbs network. Each of the committees reaches consensus on the contents of its blocks separately, with bounded throughput thanks to the bounded size of the committees. This enables Orbs to operate an unlimited number of high-throughput blockchains within the network.

info@orbs.com