**Proposal for changes to selection of random peers to become producer nodes from the worker pool**.

Original work in the consensus paper states that the peers are selected with relation to the PID and the hash of the previous data. As the PID can be manipulated by a user and thereby weight in their favor of selection to become a producer this is not usable.

Research into a RANDAO provides a viable alternative [1][2]. This is a process by which each user creates their own random value. By combining these random numbers across the network you gain a random number with high level of randomness. The larger the network the more random the number will be. The process works as follows:

- Each node $n$ in the worker pool $N$ generates a random number $r$.
- To this random number the hash of the previous ledger state, $D$, must be added.
- $n$ then creates a Blake-2b hash of the combined random number $H(r+D)$.
- Each $n$ must then send their value $r$ to the contract.
- If they do not send their $r$ value they are not eligible to become a producer node.
- Each $n$ in $N$ sends their $H(r + D)$ to a hardcoded smart contract. This creates the global random value $R$.
- The smart contract must determine:
  - That the user did in fact use the $D$ value when generating the random number. This is done by taking the $r$ value submitted by the user and hashing with $D$.
  - Ensuring the producer has paid a sufficient stake to take part in the selection process.
- This global random can the be used to determine the producers for the next cycle(s).
- This is done by determining the nodes that have a $H(r_i)$ closest to the $R$ value.

This method is secure from manipulation as hashing algorithms are one way functions meaning that there is no provably efficient method to inverse a hashing function i.e. retrieving a message $m$ from a digest $H(m)$. If a node does not input a value into the smart contract then they are not eligible for selection for becoming a producer for that cycle(s).

Addition of the $D$ value is necessary as this will prevent a producer from using known random values to create a desired digest that gives them an advantage when being selected. The value for $D$ must fulfill two rules, firstly it must always be the same for all nodes in the worker pool, secondly it must change with each draw of a random number when determining the random selection of producer nodes. This prevents a user creating a hash using known input and digest combinations to gain an advantage when being selected. Furthermore if the hash of the previous ledger state is used as $D$ is ensures that a prospective producer node knows the current ledger state. If they do not then their random number will be invalid as $H(r + D_{prod}) \neq H(r + D)$.

As described in RANDAO, this can be further extended to implement staking. Nominal fees to contribute can be added in order to prevent DDOS. This is done in such a way that a nominal fee is added as to not dissuade users from legitimately wanting to become producers while dissuading malicious entities from attempting to perform a Sybil attack against hte network in order to gain majority control over a producer pool for a cycle or multiple cycle. There is also the additional benefit of simplification of the overall consensus mechanism as it removes the need for a queuing mechanism as well as producing a verifiable method of keeping track of what nodes are registering to be producers for any given cycle. It also thereby in turn provides evidence to other nodes on the network who the producers for any given cycle are as the process will be verifiable.

# References


[1] B. Skvorc, "Two point oh: Randomness." `https://our.status.im/two-point-oh-randomness/`, 07/05/2019.

[2] randao, "randao." `https://github.com/randao/randao/`, 26/03/2019.