Abstract

Currently, randomness, be it on-chain or off-chain, is only uniform. Gaussian randomness is made available by simply counting 1’s in the binary representation of a hashed value calculated by the keccak256

hashing algorithm. It is simple, costs little gas, and can open up many possibilities in gaming and DeFi.

Motivation

DApps may desire to generate some numbers more frequently than the others, but currently, the randomness produced by keccak256

hashing algorithm is uniform in the domain [0, 2**256-1]

. That is limiting what is possible with Solidity and blockchains. This on-chain Gaussian RNG can satisfy such needs.

Specification

The algorithm relies on the count of 1’s in the binary representation of a hashed value produced by the keccak256

hashing algorithm. By Lyapunov Central Limit Theorem, this count after proper transformations, has a Gaussian distribution. The theoretical basis, condition and proofs as well as Solidity implementation and practical issues can be found here.

Backwards Compatibility

This is a brand new algorithm and there is no backwards compatibility issue. Actually, it is already with Solidity and it got a chance to come to light.