Verifiable Random Function (VRF)

An additional cryptographic primitive utilized in the AIVM Execution layer is the Verifiable Random Function (VRF). VRFs are pivotal for the generation of unbiased and unpredictable random values that are also verifiable by any party with public information. A VRF is a construct that allows its holder to provide proof that a number was generated in a truly random and secure manner, akin to a cryptographic lottery that cannot be rigged.

On DNA Layer, VRFs are utilized to enhance the security and fairness of node selection processes. When executing AI tasks, the committee of nodes responsible for these tasks must be chosen without any bias or manipulation. VRFs serve this purpose by allowing for the secure and fair selection of the inference committee. Each node generates a random number with its private VRF key and publishes both the number and its proof. The verifiability aspect of VRFs ensures that all other nodes or participants can independently verify the correctness of the random number, ruling out any foul play in the selection process.

This mechanism preserves the decentralization ethos of DNA Layer by preventing central authorities from controlling the selection process but also instills confidence in the network participants. This ensures that each AI task is processed by a randomly chosen, yet reliably determined, committee of nodes, which upholds the integrity and unpredictability of the selection protocol for the guarantee of increased transparency and security over assumptions of good intent.

We briefly give some more details about VRF below. It consists of three functions: keygen, evaluate, and verify.

- $\mathsf{keygen}(r) \rightarrow (pk, sk)$: generates a public key pk and a secret key sk for a given random input r.
- $\mathsf{evaluate}_{sk}(x) \rightarrow (y, \pi)$: generates pseudorandom output y and a proof π from the secret key sk and x as inputs.
- $\mathsf{verify}_{pk}(x, y, \pi) \in \{\mathsf{true}, \mathsf{false}\}$: takes the public key pk, the pseudorandom output y, the proof π, and the message x as inputs, and returns true if y is actually the output produced by sk and x. If not, it returns false.

Previous
Secure Multi-Party Computation (MPC)

Next
Zero-Knowledge Proof (ZKP)