Randomness API - Secret VRF

An introduction to Secret VRF, a secure and verifiable random number generator

Introduction

Secret Network's randomness API allows developers to access random numbers in their CosmWasm contracts, enhancing the capabilities of the platform. The randomness feature is accessible within Secret Contracts through the Env struct. It includes an optional random field, which contains a random number as a Binary type. The random field is only available when the "random" feature is enabled.

Use Cases

Randomness is essential in many applications, including:

- Gaming and gambling platforms, where fair and unpredictable outcomes are crucial
- Cryptographic systems that require secure random keys or nonces
- Randomized algorithms for various use cases, such as distributed systems or optimization problems
-

How It Works

1. The proposer for each block generates a strong, random seed inside SGX
2. .
3. This seed is then included in the block header and signed by all validators who can verify its authenticity inside their SGX.
4. Secret Network's in-SGX light client prevents the proposer from simulating a block before all other validators sign it. Consequently, the proposer cannot gain maximal extractable value (MEV) by generating random seeds until they find a favorable simulation of the block.
5. Before calling the contract, the chain injectsenv.block.random = hkdf_sha256(block_random_seed + wasm_call_count)
6. .
7. Thus each contract call gets a unique random seed.
8.

For a more in-depth explanation of why and how this method of randomness works feel free to read the feature explainer

Last updated3 months ago On this page *Introduction * Use Cases * How It Works

Was this helpful? Edit on GitHub Export as PDF