Decentralized Randomness

Introduction

Randomness is a cornerstone in various aspects of modern civilization, from voting systems to financial services. The most significant application of randomness is in the realm of cryptography. Over the years, several systems have attempted to provide robust randomness, but many have fallen short in various ways.Drand , a distributed randomness beacon, aims to address these shortcomings by creating a decentralized random number generator that is unpredictable, publicly-verifiable, bias-resistant, always available, and decentralized.

How it works

Overview

- The Drand network consists of nodes that run the Drand protocol. Before generating random numbers, these nodes agree on a threshold parameter.
- Each node creates a signature. Random numbers are generated by nodes broadcasting a part of their signature to the rest of the network.
- Nodes wait and collect these signatures until they have enough to match the threshold parameter. Once matched, a node can create the final signature, which is then hashed to produce the randomness.
-

Public Randomness

- Generating public randomness is Drand's primary function. This randomness is collectively generated by drand nodes and made publicly available.
- A Drand randomness beacon consists of two phases:
-
    - Setup
-
    - : Each network uses a long-term public/private key pair. All public keys are written to a group file with other necessary metadata. After distributing this group file, nodes perform a distributed key generation (DKG) protocol to create a collective public key and a private key share for each server.
-
    - Generation
-
    - : After setup, the network switches to randomness generation mode. Any node can initiate a randomness generation round by broadcasting a message. All other participants sign this message. Once a node (or third-party observer) has gathered enough partial signatures, it can reconstruct the full signature, which is then hashed to produce the collective random value.
- *
-

How Automata uses Drand

Leveraging Drand's decentralized and publicly-verifiable, Automata builds a VRF project to generates verifiable random numbers that can be easily integrated into dApps. This project runs in TEE(trusted executed environment), and uses Drand as one factor for the hardware-based unpredictable and verifiable random number. Learn more inAutomata VRF .

References

- Drand
- , a distributed randomness beacon daemon
- BLS12-381
- elliptic curve
- Automata VRF
-

Previous Account Abstraction Next Maximal Extractable Value Last updated6 months ago On this page *Introduction * How it works * How Automata uses Drand * References

Was this helpful?