Security Considerations After reading this page you will:

- Understand the importance of security measures when implementing Gelato VRF in your dApp.
- Recognize the need for state locking to prevent front-running and maintain the integrity of the randomization process.
- Learn the benefits of using RNGLib to ensure the randomness you receive is unique and secure, particularly when handling multiple requests simultaneously. * Important Note

Contrary to some other VRF providers, Gelato VRF is verifiable off-chain but not on-chain. This is due to the nature of the BLS signatures used by Drand network, which are not yet supported at EVM level. With the upcoming EIP-2537 release, adding BLS precompile for BLS12-381 curve, we aim to add support for on-chain randomness verification in a near future on all networks that will include this precompile.

Security Precautions

When integrating with GelatoVRF, it's essential to take several precautions to ensure the safety and reliability of your application. Here are key considerations:

1. State Locking and Front-Running Prevention

After you initiate a request for randomness and before the random number gets delivered, it's essential to lock the relevant application state in your consumer contract. This step minimizes the risk of front-running attacks.

In essence, front-running involves gaining an unfair advantage by making transactions based on foreknowledge of pending transactions. By locking the state, you add an additional layer of security against such tactics.

1. Usage of RNGLib

Instead of using the received randomness directly, consider integrating it with our RNGLib. This approach:

- Enables dynamic fetching of random values as required.
- Offers protection against certain bet arbitrage attacks, especially when multiple applications operate simultaneously.
-

By inheriting fromGelatoVRFConsumerBase.sol , your contract will automatically benefit from enhanced security. All fulfilled randomness requests will be dynamically derived from the drand randomness using a pseudo-random number generator (RNG). This is crucial to ensure the uniqueness of values, particularly for concurrent requests, and adds another layer of protection against potential vulnerabilities.

Previous How does Gelato VRF Work?Next Template Last updated3 months ago On this page *Security Precautions * 1. State Locking and Front-Running Prevention * 2. Usage of RNGLib