Getting Started with Band VRF

Compatibility Notice Band VRF is compatible withBandChain v2 Mainnet and isnot yet implemented in BandChain v3 .

Please ensure you're using BandChain v2 for this module to function as expected. This guide serves as a quick reference on how to request random data from the Band VRF. For a detailed reference with examples, please refer to theVRF Integration section.

Step 1: Prepare a VRF consumer contract

1. Create a VRF consumer contract that can call therequestRandomData
2. function on theVRFProvider
3. contract.
4. Implement a callback function on the VRF consumer contract, which allows theVRFProvider
5. contract to call back and execute some logic against the returned result.

Step 2: Choose a resolving method

There are currently 3 methods for relaying and resolving the VRF request:

- Band's VRF worker solution
-
    - We provide both standard and customized solutions for all clients. PleaseEmail Us
- for more details.
- Manually resolve on CosmoScan
-
    - This is an ideal and low cost solution for one-off Band VRF requests. Please refer to thisguide
- for how to resolve manually.
- Implement your own resolver bot
-
    - Anyone can implement their own version of resolver bot. An open-source version of Band's VRF worker bot is available atVRFWorkerV1 repository
- .

Step 3: Request a random value

You are now ready to request a random value from the Band VRF.

A summary of the Band VRF process is outlined below:

1. Simply call the request function on you VRF consumer contract that implements therequestRandomData
2. function in Step 1, providing aseed
3. and an optionalmsg.value
4. .
5. Depending on the resolving method chosen in Step 2, the request is sent to the BandChain.
6. The VRF oracle script on the BandChain forwards the request to a randomly chosen data source, and then retrieves the returned result and the corresponding proof of authenticity.
7. Depending on the resolving method chosen in Step 2, the proof is relayed to theBridge
8. contract for verification on the client chain via theVRFProvider
9. contract.
10. If the verification succeeds, the result (random value) is returned to the VRF consumer contract via the callback function mentioned in Step 1. Previous Introduction Next VRF integration