Requesting data commitment ranges

By default, the Blobstream X deployments on Ethereum will be updating every 4 hours, and on Arbitrum One and Base, updating every 1 hour. If you wish for the Blobstream X contract to be updated at a different cadence, then you have several different options for how to update the smart contract.

To request proofs to be submitted to the Blobstream X contract at a different cadence, you can do one of the following:

Recommended setup

Run the Blobstream X operator with hosted proving on the Succinct platform, by running an operator script that pings the platform with proof requests at a specified cadence.

Follow these instructions to run the operator script.

Here are example values for the.env file:

1. TENDERMINT_RPC_URL
2. fromthe public Celestia list
3. .
4. SUCCINCT_RPC_URL
5. =https://alpha.succinct.xyz/api
6. Request forSUCCINCT_API_KEY
7. fromthe Succinct team
8. .
9. CHAIN_ID
10. is the chain ID of the deployed Blobstream X contract.
11. CONTRACT_ADDRESS
12. : Blobstream X proxy contract address.
13. NEXT_HEADER_FUNCTION_ID
14. &HEADER_RANGE_FUNCTION_ID
15. : Get thefunctionId
16. 's from the Blobstream X contract by using thenextHeaderFunctionId
17. andheaderRangeFunctionId
18. respectively, which are public storage variables.

Local proving

To run the Blobstream X operator with local proving, follow thisguide .

TIP

Note: Requires a large cloud machine to run in a reasonable amount of time. EC2 r6a.16xlarge takes ~30 minutes to generate a header range proof.

Request proof onchain

Directly request a proof via the Blobstream X contract interface. Unlike the Blobstream X operator which handles requests off-chain, requesting on-chain requires gas, but the proof will be generated and relayed by the Succinct platform.

1. CallrequestHeaderRange(uint64 _targetBlock)
2. with the end of the range you want a commitment for.
3. ADataCommitmentStored(uint256, uint64, uint64, bytes32)
4. will be emitted for the requested range when it is stored in the contract. Listen to this event to know that the proof has been generated successfully. [][Edit this page on GitHub] Last updated: Previous page Example implementation of Blobstream proofs by CryptoKass Next page New Blobstream X deployments []