AnyTrust Orbit chains: Keyset generation

AnyTrust chains rely on an external Data Availability Committee (DAC) to store data and provide it on demand, instead of using the parent chain as Data Availability (DA) layer.

The DAC hasN members; the AnyTrust protocol assumes that a minimum ofH DAC members maintain integrity.H is the minimum number of trusted committee members on AnyTrust chains, configurable by the chain's owner via theassumed-honest parameter in the keyset. In scenarios whereK = (N + 1) - H members of the DAC pledge to grant access to specific data, they must sign and attest they have the data for storage to be considered successful.

Each DAC member gets their own set of BLS public and private keys. It's important for every member to create their own new and secure BLS keys. They should do this on their own and make sure these keys are random and only for their use. If you need help generating BLS keys, please refer to our guide on generating keys in the Arbitrum documentation:Generating BLS Keys .

The main blockchain (parent chain) needs to know the names and public keys of all DAC members in order to validate the integrity of data being batched and posted. A 'keyset' is a list of all DAC members' public keys. It also shows how many signatures are needed to approve a Data Availability Certificate. This design lets the chain owner modify the DAC's membership over time, and it lets DAC members change their keys if needed. SeeInside Anytrust for more information.

In the following section, we will provide a detailed guide on the generation of a Keyset corresponding to your individual set of keys, as well as instructions for its subsequent configuration within the chain.

Batch Poster Configuration

AnyTrust works with a group of Data Availability Servers, forming a committee that ensures transaction data is accessible. When setting up the Nitro Batch Poster, you need to provide specific information for each committee member. This includes their URL, BLS public key, a unique single-bit identifier (bitmask) for each member, and a parameter known as assumed-honest. As mentioned before, assumed-honest refers to the minimum number of committee members that we trust.

To ensure data is stored properly, a certain number of committee members need to confirm they have the data. This number is calculated asK = (N + 1) - H , whereN is the total number of committee members andH is the minimum number of members assumed to be honest.

Someone setting up an AnyTrust L3 would need to first set up the committee of Data Availability Servers, including generating their BLS keys. How to do that is describedin this guide .

Here is a sample of the JSON configuration, taken from a past configuration of Arbitrum Nova, which was used with to the batch poster configuration. Note that due to a quirk in a configuration library we use in Nitro, thebackends field is an escaped JSON string withurl ,pubkey , andsigner-mask fields.pubkey is the base64 encoded BLS public key of the committee member, andsigner-mask should be a power of 2, starting from 2^0, 2^1, etc.

```
{ .. . "data-availability" :

{ "enable" : true, "rpc-aggregator" :

{ "enable" : true, "assumed-honest" :

2 , "backends" :

"[{ \" url \" : \" http://example \" , \" pubkey \" : \"
YAbcteVnZLty5qRebeswHKhdjEMVwdou+imSfyrI+yVXHOMdLWA3Nf4DGW9tVry/mhmZqJp01TaYIsREXWdsFe1S5QCNqnddyag5yZ/5Y6GZRqx0BXmHTaxPY5kHrhvGnwxmlJVbUk1xjKRFgxxTdTk3c
\" , \" signermask \" :1},{ \" url \" : \" http://example \" , \" pubkey \" : \"
YAg1+ZXyR48kiS0FDaoon4trnBsYW80oUy+I1hDCZCotxvNQl0AjbTPD4tkTaqsX+BnIxnEpO7ondxd2Lo0cH3usnhfdKNKTmpWbs45QD5wRw4zrvEJuLeqXxAF1pIXRdACubHX/SeiEx5RpJJ5wlTJYhUtk+
\" , \" signermask \" :2},{ \" url \" : \" http://example \" , \" pubkey \" : \"
YAXbmOUQgLs5Kntevb/PM+D08BkxAsxA95qe8KlVfFpi3R74AAVpRugyn5eboMyCUQ0Nx4w8zv+mbuXeXimJh6mFi/UmIXFhTlVvQGh85pEsvqaltERyyz/xB+zmnL0P2g2zkqZKgr5xQHc1HWOE1s6iVK
\" , \" signermask \" :4},{ \" url \" : \" https://example \" , \" pubkey \" : \"
YAbOg53k1qOuAvJbQIlTHmo9LeVWvQBr0wzy00CLl30Y8XVt1KG8PADbkALw2O8a9Q+6ppWd7L7By+I0zG72JwoDM5CQ4COPisn4oY9EuHNMjzthI90SiuSKCGO5p/bYgwIENoF3LCt581DBS8nXsY5X
\" , \" signermask \" :8},{ \" url \" : \" https://example \" , \" pubkey \" : \"
YA+HK4mKT9G4rnNRX30zzXvh6XHOGJaqvvL4km5YbEJI3A23/XhRQCwUFJ3D3ITzgww0YWfDnlMjlxrDQEFfCi6wVKmo4KXVA6Ks/s690d9xrurDs4JgSAxpm8CZNPCRPg7lquq9VzEyhSB+uJNmtBEo
\" , \" signermask \" :16},{ \" url \" : \" https://example \" , \" pubkey \" : \"
YALC7DeOtroXqegbj9RCY9aZw0cZSSpOzx7napQrwiR4+3qflOLxWCJjDy1hbDKjNAOHEY5LluJtbkHbqrn+J61gi9gjoUL5iPfamZzeygirSv7baz2i1NsgjMC6kb/UThU71zc2t98BNBeAqqfxhfyg06R437U7Y
\" , \" signermask \" :32},{ \" url \" : \" https://example \" , \" pubkey \" : \"
YBN+CWUmeRP56vhb/yLjzl9Euxv67XZ5sWgKzRVDaoQyXrp/KWLKRpN8y/Rtme3JRANM3Ze8T7HY3DrducNIQxqZl1lZ5qyCODdq8x8D51T6PDFZJ81oYCZeyObpfaQKlQkyd3PnqlvPrvdpDXaQYzNvb
\" , \" signermask \" :64},{ \" url \" : \" https://example \" , \" pubkey \" : \"
YAR40SbOOU71LW/8aEVnLfztsU1Mq+dqzZ7/8liSsx3DLYvSFCZXXwijCxuEu4wfZQeBDiXUeFLx8qBrZrU0HQLXSBoczgElfnaKoaWbaDoo9veUZnRUHw9OI2Q9Md/X6QlYo2HH24a2KP4HXZTIXixD+F
\" , \" signermask \" :128}]" } } .. . }
```

Keyset Generation

For the Batch Poster to be able to post batches, the keyset corresponding to the configuration it is using must be enabled on the Inbox contract. You'll need to generate the keyset and keyset hash binary blobs to pass to theSetValidKeyset call on the Inbox contract. Here's an example using the same Nova @Keyset 8 configuration as before, and the datool utility which is distributed with Nitro:

```
{ "keyset" :

{ "assumed-honest" :

2 , "backends" :

"[{ \" url \" : \" http://example \" , \" pubkey \" : \"
YAbcteVnZLty5qRebeswHKhdjEMVwdou+imSfyrI+yVXHOMdLWA3Nf4DGW9tVry/mhmZqJp01TaYIsREXWdsFe1S5QCNqnddyag5yZ/5Y6GZRqx0BXmHTaxPY5kHrhvGnwxmlJVbUk1xjKRFgxxTdTk3c
\" , \" signermask \" :1},{ \" url \" : \" http://example \" , \" pubkey \" : \"
YAg1+ZXyR48kiS0FDaoon4trnBsYW80oUy+I1hDCZCotxvNQl0AjbTPD4tkTaqsX+BnIxnEpO7ondxd2Lo0cH3usnhfdKNKTmpWbs45QD5wRw4zrvEJuLeqXxAF1pIXRdACubHX/SeiEx5RpJJ5wlTJYhUtk+
\" , \" signermask \" :2},{ \" url \" : \" http://example \" , \" pubkey \" : \"
YAXbmOUQgLs5Kntevb/PM+D08BkxAsxA95qe8KlVfFpi3R74AAVpRugyn5eboMyCUQ0Nx4w8zv+mbuXeXimJh6mFi/UmIXFhTlVvQGh85pEsvqaltERyyz/xB+zmnL0P2g2zkqZKgr5xQHc1HWOE1s6iVK
\" , \" signermask \" :4},{ \" url \" : \" https://example \" , \" pubkey \" : \"
YAbOg53k1qOuAvJbQIlTHmo9LeVWvQBr0wzy00CLl30Y8XVt1KG8PADbkALw2O8a9Q+6ppWd7L7By+I0zG72JwoDM5CQ4COPisn4oY9EuHNMjzthI90SiuSKCGO5p/bYgwIENoF3LCt581DBS8nXsY5X
\" , \" signermask \" :8},{ \" url \" : \" https://example \" , \" pubkey \" : \"
YA+HK4mKT9G4rnNRX30zzXvh6XHOGJaqvvL4km5YbEJI3A23/XhRQCwUFJ3D3ITzgww0YWfDnlMjlxrDQEFfCi6wVKmo4KXVA6Ks/s690d9xrurDs4JgSAxpm8CZNPCRPg7lquq9VzEyhSB+uJNmtBEo
\" , \" signermask \" :16},{ \" url \" : \" https://example \" , \" pubkey \" : \"
YALC7DeOtroXqegbj9RCY9aZw0cZSSpOzx7napQrwiR4+3qflOLxWCJjDy1hbDKjNAOHEY5LluJtbkHbqrn+J61gi9gjoUL5iPfamZzeygirSv7baz2i1NsgjMC6kb/UThU71zc2t98BNBeAqqfxhfyg06R437U7Y
\" , \" signermask \" :32},{ \" url \" : \" https://example \" , \" pubkey \" : \"
YBN+CWUmeRP56vhb/yLjzl9Euxv67XZ5sWgKzRVDaoQyXrp/KWLKRpN8y/Rtme3JRANM3Ze8T7HY3DrducNIQxqZl1lZ5qyCODdq8x8D51T6PDFZJ81oYCZeyObpfaQKlQkyd3PnqlvPrvdpDXaQYzNvb
\" , \" signermask \" :64},{ \" url \" : \" https://example \" , \" pubkey \" : \"
YAR40SbOOU71LW/8aEVnLfztsU1Mq+dqzZ7/8liSsx3DLYvSFCZXXwijCxuEu4wfZQeBDiXUeFLx8qBrZrU0HQLXSBoczgElfnaKoaWbaDoo9veUZnRUHw9OI2Q9Md/X6QlYo2HH24a2KP4HXZTIXixD+F
\" , \" signermask \" :128}]" } } .. ./nitro ./target/bin/datool dumpkeyset --conf.file datestconf/datool-keyset.conf Keyset:
0x0000000000000002000000000000000801216006dcb5e56764bb72e6a45e6deb301ca85d8c4315c1da2efa29927f2ac8fb25571ce31d2d603735fe03196f6d56bcbf9a1999a89a74d5369822c4445d676c
KeysetHash: 0xf8bb9a67839d1767e79afe52d21e97a04ee0bf5f816d5b52c10df60cccb7f822
```

Example with single private key = zero

For example, in the case of a solitary, zero-valued private key, the setup of the keys, Keyset, and Sequencer configuration can be set as detailed below.

Key

```
cat /tmp/orbit-bls/das_bls
```

this is an empty file, the private key is zero

```
cat /tmp/orbit-bls/das_bls.pub
YAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
==
```

Keyset

Once you've generated the keys, it is imperative to generate the Keyset as previously outlined. For example, in the scenario involving a zero-valued private key, the Keyset configuration would be as follows:

Keyset:
0x00000000000000010000000000000001012160000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
KeysetHash: 0x4d795e20d33eea0b070600e4e100c512a750562bf03c300c99444bd5af92d9b0 Upon successfully generating the Keyset, it is essential to establish it within the parent chain. This step ensures that the parent chain is accurately informed of the Committee members' keyset.

The Keyset can be configured by invoking thesetValidKeyset method within the SequencerInbox contract.

Note: Only rollup owner(s) can call this method to set the new valid keyset.

Sequencer Configuration

It is necessary to modify the node configuration file associated with the node intended to initiate your chain. To incorporate the Committee keys into this node configuration, the following segment must be appended to the JSON file:

{ .. . "data-availability" :

{ "enable" : true, "rpc-aggregator" :

{ "enable" : true, "assumed-honest" :

1 , "backends" :

"[{ \" url \" : \" http://localhost:9876 \" , \" pubkey \" : \"
YAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
\" , \" signermask \" :1}]" } } .. . } Edit this page Last updatedonMar 7, 2024 Previous How to add your testnet Orbit chain to Arbitrum's bridgeNext Orbit chain ownership