Start EthSigner with multiple signing keys

EthSigner supports transaction signing usingmultiple keys .

This tutorial covers configuring multiple keys using V3 keystore files. To configure keys forHashiCorp Vault orAzure Key Vault , update theTOML configuration file accordingly.

note Multiple signing keys is not the same as multi-tenancy. EthSigner does not support multi-tenancy.

Multi-tenancy is a feature inHyperledger Besu andTessera allowing multiple participants in a privacy network to use the same Besu and Tessera node.

Prerequisites

- EthSigner
- Hyperledger Besu
- Node.js
- web3.js
- .

note The Ethereum client used in this documentation is Hyperledger Besu but EthSigner can be used with any Ethereum client.

Start Besu

Start Besu with the--rpc-http-port option set to8590 .

besu --network

dev --miner-enabled --miner-coinbase

0xfe3b557e8fb62b89f4916b721be55ceb828dbd73 --rpc-http-cors-origins

"all" --host-allowlist = * --rpc-http-enabled --rpc-http-port = 8590 --data-path = /Users/me/Datadir

Create password and key files

You can create one or more password and V3 Keystore key files. Create a text file containing the password for the V3 Keystore key file to be created (for example,passwordFile).

Use theweb3.js library to create a key file where:

-
- is the account private key EthSigner uses to sign transactions.
-
- is the key file password being created. The password must match the password saved in the password file created previously (passwordFile
- in this example).

Example * Create key file * Example

const

Web3

=

require ("web3") ;

// Web3 initialization (should point to the JSON-RPC endpoint) const web3 =

new

Web3 (new

Web3 . providers . HttpProvider ("http://127.0.0.1:8590")) ;

var

V3KeyStore

= web3 . eth . accounts . encrypt ('' ,

"") ; console . log (JSON . stringify (V3KeyStore)) ; process . exit ( ) ; const

Web3

=

require ('web3')

// Web3 initialization (should point to the JSON-RPC endpoint) const web3 =

new

Web3 (new

Web3 . providers . HttpProvider ('http://127.0.0.1:8590'))

var

V3KeyStore

= web3 . eth . accounts . encrypt ("0x8f2a55949038a9610f50fb23b5883af3b4ecb3c3bb792cbcefbd1542c692be63" ,

"password") ; console . log (JSON . stringify (V3KeyStore)) ; process . exit ( )`;< ! -- / tabs --

Copy and paste the example JS script to a file (for example,createKeyFile.js) and replace the placeholders.

Use the JS script to display the text for the key file:

node createKeyFile.js Copy and paste the text to a file (for example,keyFile). The file is your V3 Keystore key file. Each key file requires a TOML file.

Create the TOML file

Create the TOML file that contains the settings to access the key file. Each key that signs transactions requires a TOML file.

The file name must use the format[].toml . Remove the0x portion of the account address. For example,78e6e236592597c09d5c137c2af40aecd42d12a2.toml .

[metadata] createdAt = 2019-11-05T08:15:30-05:00 description = "File based configuration"

[signing] type = "file-based-signer" key-file = "/Users/me/project/keyFile" password-file = "/Users/me/project/passwordFile"

Start EthSigner

Start EthSigner with options:

- chain-id
- is the chain ID specified in theBesu genesis file
- .
- downstream-http-port
- is therpc-http-port
- specified for Besu (8590
- in this example).
- directory
- is the location of TOML filecreated above
- .

Start EthSigner ethsigner --chain-id=2018 --downstream-http-port=8590 multikey-signer --directory=/Users/me/project If using a cloud-based Ethereum client such asInfura , specify the endpoint using the--downstream-http-host and--downstream-http-path command line options.

ethsigner --chain-id=5 --downstream-http-host=goerli.infura.io \ --downstream-http-path=/v3/d0e63ca5bb1e4eef2284422efbc51a56 --downstream-http-port=443 \ --downstream-http-tls-enabled multikey-signer --directory=/Users/me/project

Confirm EthSigner is running

Use theupcheck endpoint to confirm EthSigner is running.

info * curl HTTP request * Result

curl -X GET http://127.0.0.1:8545/upcheck I'm up

Confirm EthSigner is passing requests to Besu

Request the current block number usingeth_blockNumber with the EthSigner JSON-RPC endpoint (8545 in this example):

curl -X POST --data '{"jsonrpc":"2.0","method":"eth_blockNumber","params":[],"id":51}' http://127.0.0.1:8545 You can nowuse EthSigner to sign transactions with the keys stored in the V3 Keystore key files.Edit this page Last updatedonMar 30, 2023 byEric Lin Previous Start with a single signer Next Quorum Developer Quickstart