Multisig Keys

Generating A Multisig Wallet

You can generate a multisig wallet using keys you own using the secretcli by:

```

Copy secretclikeysadd--multisig=acc1,acc2,acc3[...]\ --multisig-threshold=K

```

The command above will generate an output similar to the following:

```

Copy -name:test_multisig type:multi address:secret1whdl9yjy8c7p3062xjehf2m69evljp8yfcv9zt
pubkey:'{"@type":"/cosmos.crypto.multisig.LegacyAminoPubKey","threshold":2,"public_keys":
[{"@type":"/cosmos.crypto.secp256k1.PubKey","key":"AiXwUPtwTJqxKZq/BjKi+7EFhqR2Aj9QT94lFzb5Ednp"},
{"@type":"/cosmos.crypto.secp256k1.PubKey","key":"A7QMHOt+yLGddDxey51QLofwsTJWfqyzYmNOB9L1Oz1S"},
{"@type":"/cosmos.crypto.secp256k1.PubKey","key":"A0QMBqFY4J39i6NrH4qR5uOEnyytpkyeWFg/e0sPd8NJ"}]}'
mnemonic:""

```

Generating A Multisig Wallet With Keys You Do Not Own

You can generate multisig wallets with your own keys, and using other public keys you do not own using the secretcli:

```

Copy

# Add public keys of address you want to have on the multisig wallet to the

# secretcli keyring

secretclikeysaddpub_addr_1\ --
pubkey='{"@type":"/cosmos.crypto.secp256k1.PubKey","key":"A3YmkInq4LS9XJzf+4Xk7SV9Pasybpoc4mNGrko4exyH"}'

secretclikeysaddpub_addr_2\ --
pubkey='{"@type":"/cosmos.crypto.secp256k1.PubKey","key":"As3QBF2LipZdMUP7ICwS3wfkpybNImCq1uDVjSuUvQCg"}'

# Verify keys were added to secretcli keyring

# Keys added to the keyring using public keys only will be of 'type: offline'

secretclikeyslist

```

Now that all the keys are present in your local secretcli keyring, create a multisig wallet as you normally would with keys that you own:

```

Copy secretclikeysadd\ test_multisig --multisig=test,pub_addr_1,pub_addr_2 --multisig-threshold=2

```

To confirm the multisig wallet was made with the correct addresses use:

```

Copy secretclikeysshowMULTISIG_NAME-a

# You should see the multisig addresses as an output

```

K Value

K is the minimum number of private keys that must have signed the transactions carrying the public key's address as a signer. Typically the K value will be lower than the total number of wallets associated with the multisig wallet, and high enough to require the majority of associated wallets to approve transactions.

Best Practices

For example, if there are 6 controlling addresses associated with a multisig wallet it would be poor practice for set the K value to 1, 2, 5, or 6. If the K value is too low (i.e 1 or 2) a minority of multisig members will be always in control of the multisig wallet; if any one or two members agree on making a transaction they will be able to even if the remaining 4-5 members do not agree with the transactions. If the K value is too high, and one or two of the members wallets on the multisig are lost or compromised no transactions with the multisig will be possible, and all associated assets held by the multisig wallet will be lost.

Multisig Flags

The--multisig flag must contain the name of public keys to be combined into a public key that will be generated and stored asnew-key-alias in the local database.

All names supplied through--multisig must already exist in the local database. is set. The order of the supplied keys on the command line does not matter, i.e. the following commands generate two identical keys:

```

Copy secretclikeysadd--multisig=foo,bar,baz--multisig-threshold=2 secretclikeysadd--multisig=baz,foo,bar--multisig-threshold=2

```

To make the multisig wallet keys get passed into the multisig wallet in a specific order (i.e the order they are given) the--nosort flag must be used:

```

Copy secretclikeysaddmultisig_sorted\ --multisig=pub_addr_2,test,pub_addr_1\ --multisig-threshold=2\ --nosort

```

For demonstration purposes, the above command will produce a multisig wallet where each key is added in the exact order they are given to the public keys associated with the multisig wallet:

```

Copy secretclikeysshowmultisig_sorted--pubkey|jq { "@type":"/cosmos.crypto.multisig.LegacyAminoPubKey", "threshold":2, "public_keys":[ {# This is the public key associated with 'pub_addr_2' "@type":"/cosmos.crypto.secp256k1.PubKey", "key":"As3QBF2LipZdMUP7ICwS3wfkpybNImCq1uDVjSuUvQCg" }, {# This is the public key associated with 'test' "@type":"/cosmos.crypto.secp256k1.PubKey", "key":"Ai2I8uRsSxk7QjpEWShNL97ZIdqzq/YE8ymkbCpybS5P" }, {# This is the public key associated with 'pub_addr_1' "@type":"/cosmos.crypto.secp256k1.PubKey", "key":"A3YmkInq4LS9XJzf+4Xk7SV9Pasybpoc4mNGrko4exyH" } ] }

```

Last updated1 year ago On this page *Generating A Multisig Wallet * Generating A Multisig Wallet With Keys You Do Not Own * K Value * Multisig Flags

Was this helpful? Edit on GitHub Export as PDF