An Introduction to Namada Addresses

All accounts in Namada have a unique address, exactly one Validity Predicate and optionally any additional data in its dynamic storage sub-space.

There are currently 3 types of account addresses:

- Implicit:
- An implicit account is derived from your keypair and can be used to authorize certain transactions from the account. Implicit accounts have no code attached to them, and can only have one key controlling it. This differs it from established accounts, which can support multiple keys.
- Established:
- An established account is a associated with one or more cryptographic keys. Each account has thevp_user
- predicate which validate any associated transaction. The main purpose of this code is to ensure that the multisignature threshold is correctly met and stores the keys that verify transactions. All established accounts will be initialized through on-chain transactions, unlike implicit accounts, which exist as soon as the keypair is generated.
- Internal:
- Special internal accounts, such as protocol parameters account, PoS and IBC.

Manage keypairs

Namada useseed25519(opens in a new tab) keypairs for signing cryptographic operations on the blockchain.

To manage your keys, various sub-commands are available under:

namada

wallet

--help

Generate a keypair

It is possible to generate keys using the CLI. By doing so, an implicit account address is also derived in the process and added to storage.

ð¤¡ Note the use of the placeholderkeysha for the key parameter. This is a completely configurable parameter, and should just refer to the alias of the key signing the transaction (that has a positive nam balance). KEY_ALIAS = "keysha" namada

wallet

gen

--alias KEY_ALIAS The derived implicit address shares the samekeysha alias. The previous command has the same effect asnamada wallet gen --alias keysha . By default, the keys are stored encrypted. The encryption password isnot a part of key generation randomness.

Namada wallet supports keypair generation using amnemonic code(opens in a new tab) andHD derivation path(opens in a new tab) . To generate a keypair for a default path use

KEY_ALIAS

"keysha" namada

wallet

gen

--alias KEY_ALIAS --hd-path

default The default HD path for Namada ism/44'/877'/0'/0'/0' . Optionally, the user may specify an additional passphrase thatis used as a part of keypair generation randomness.

ð¨ WARNING: Keep your mnemonic code and passphrase safe. Loss of any of them would inevitably lead to impossible account recovery.

Restore the keypair

To recover the keypair from your mnemonic code and passphrase use

namada

wallet

derive

--alias

keysha

--hd-path

default This will ask you to then enter a passphrase (unless you add the--unsafe-dont-encrypt flag), and then will ask you to enter your mnemonic code.

List all known keys

namada

wallet

list

--keys

Manage addresses

To manage addresses, similar to keys, various sub-commands are available:

namada

wallet

--help

Generate an implicit address

Let's call the implicit addressaccountant :

namada

wallet

gen

--alias

accountant namada

wallet

gen

--alias

keysha

--hd-path

default generates an address using amnemonic code(opens in a new tab) andHD derivation path(opens in a new tab).

Restore the address

namada

wallet

derive

--alias

keysha

--hd-path

default

List all known addresses

namada

wallet

list

--addr

Hardware wallet Multisignature