

seid keys add

Derive a new private key and encrypt to disk. Optionally specify a BIP39 mnemonic, a BIP39 passphrase to further secure the mnemonic, and a bip32 HD path to derive a specific account. The key will be stored under the given name and encrypted with the given password. The only input that is required is the encryption password.

If run with `-i`, it will prompt the user for BIP44 path, BIP39 mnemonic, and passphrase. The flag `--recover` allows one to recover a key from a seed passphrase. If run with `--dry-run`, a key would be generated (or recovered) but not stored to the local keystore. Use the `--pubkey` flag to add arbitrary public keys to the keystore for constructing multisig transactions.

You can create and store a multisig key by passing the list of key names stored in a keyring and the minimum number of signatures required through `--multisig-threshold`. The keys are sorted by address, unless the flag `--nosort` is set. Example:

```
keys add mymultisig --multisig "keyname1,keyname2,keyname3" --multisig-threshold 2
```

Usage: `seid keys add [flags]`

Flags: `--account uint32` Account number for HD derivation `--algo string` Key signing algorithm to generate keys for (default "secp256k1") `--coin-type uint32` coin type number for HD derivation (default 118) `--dry-run` Perform action, but don't add key to local keystore `--hd-path string` Manual HD Path derivation (overrides BIP44 config) `-h, --help` help for add `--index uint32` Address index number for HD derivation `-i, --interactive` Interactively prompt user for BIP39 passphrase and mnemonic `--ledger` Store a local reference to a private key on a Ledger device `--multisig strings` List of key names stored in keyring to construct a public legacy multisig key `--multisig-threshold int` K out of N required signatures. For use in conjunction with `--multisig` (default 1) `--no-backup` Don't print out seed phrase (if others are watching the terminal) `--nosort` Keys passed to `--multisig` are taken in the order they're supplied `--pubkey string` Parse a public key in JSON format and saves key info to file. `-recover` Provide seed phrase to recover existing key instead of creating

Global Flags: `--home string` The application home directory (default "~/.sei") `--keyring-backend string` Select keyring's backend (os|file|test) (default "os") `--keyring-dir string` The client Keyring directory; if omitted, the default 'home' directory will be used `--log_format string` The logging format (json|plain) `--log_level string` The logging level (trace|debug|info|warn|error|fatal|panic) `--output string` Output format (text|json) (default "text") `--trace` print out full stack trace on errors Last updated on July 1, 2024 [keys delete](#)