Ethereum SLIP-39 Account Generation

Perry Kundert

2021-12-20 10:55:0

Creating Ethereum accounts is complex and fraught with potential for loss of funds.

Creating a BIP-39 seed recover phrase helps, but a **single** lapse in security dooms the account. If someone finds your recover phrase, the account is gone.

The SLIP-39 standard allows you to split the seed between multiple recover phrases. This is better, but creating such accounts is difficult; presently, only the Trezor supports these, and they can only be created "manually". Writing down 5 or more sets of 20 words is difficult and time consuming.

The `python-slip39` project exists to assist in the safe creation and documentation of new accounts, with various SLIP-39 sharing parameters. It generates the new wallet seed, generates standard Ethereum account(s) (at derivation path `m/66'/40'/0'/0/0` by default) with Ethereum wallet address and QR code, produces the required SLIP-39 phrases, and outputs a single PDF containing all the required printable cards to document the account.

On an air-gapped computer, new accounts can safely be generated, and the PDF saved to a USD drive for printing (or directly printed without the file being saved to disk.)

Contents

1 Ethereum SLIP-39 Account Generation

1.1 The `python-shamir-mnemonic` API

Obtain python-shamir-mnemonic, and install it.

```
$ shamir create custom --group-threshold 2 --group 1 1 --group 1 1 --group 2 5 --group
Using master secret: 87e39270d1d1976e9ade9cc15a084c62
Group 1 of 4 - 1 of 1 shares required:
merit aluminum acrobat romp capacity leader gray dining thank rhyme escape genre havoc
Group 2 of 4 - 1 of 1 shares required:
merit aluminum beard romp briefing email member flavor disaster exercise cinema subjec
Group 3 of 4 - 2 of 5 shares required:
merit aluminum ceramic roster already cinema knit cultural agency intimate result ivor
merit aluminum ceramic scared beam findings expand broken smear cleanup enlarge coding
merit aluminum ceramic shadow cover smith idle vintage mixture source dish squeeze sta
merit aluminum ceramic sister duke relate elite ruler focus leader skin machine mild e
merit aluminum ceramic smug buyer taxi amazing marathon treat clinic rainbow destroy u
Group 4 of 4 - 3 of 6 shares required:
merit aluminum decision round bishop wrote belong anatomy spew hour index fishing lect
merit aluminum decision scatter carpet spine ruin location forward priest cage securit
merit aluminum decision shaft arcade infant argue elevator imply obesity oral venture 
merit aluminum decision skin already fused tactics skunk work floral very gesture orga
merit aluminum decision snake cage premium aide wealthy viral chemical pharmacy smokin
merit aluminum decision spider boundary lunar staff inside junior tendency sharp edito
$
```

2 Converting the Seed to a Standard Account

2.1 The `eth-account` API

```
>>> seed=codecs.decode("dd0e2f02b1f6c92a1a265561bc164135", 'hex_codec')
>>> eth_account.hdaccount.key_from_seed(seed, "m/44'/60'/0'/0/0")
b'\x17\x88p\x00\x94\x16\x17L\x96\x97w{\x1d\x94"\x95\x04\xe8?%\xb1`^{\xb12\xaa[\x88\xda
>>> key=eth_account.hdaccount.key_from_seed(seed, "m/44'/60'/0'/0/0")
>>> keyhex=codecs.encode(key, 'hex_codec')
>>> keyhex
b'178870009416174c9697777b1d94229504e83f25b1605e7bb132aa5b88da64b6'
>>> keyhex.encode('ascii')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
```

```
AttributeError: 'bytes' object has no attribute 'encode'
>>> keyhex.decode('ascii')
'178870009416174c9697777b1d94229504e83f25b1605e7bb132aa5b88da64b6'
>>> keyhex = '0x'+keyhex.decode('ascii')
>>> keyhex
'0x178870009416174c9697777b1d94229504e83f25b1605e7bb132aa5b88da64b6'
>>> account = eth_account.Account(keyhex)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: Account() takes no arguments
>>> account = eth_account.Account.from_key(keyhex)
>>> acount
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'acount' is not defined
>>> account
<eth_account.signers.local.LocalAccount object at 0x7fba368ae670>
>>> account.address
'0x336cBeAB83aCCdb2541e43D514B62DC6C53675f4'
```