pycoin, ku, and tx

2022-03-04

# pycoin, ku, and tx

The Python library [pycoin](https://github.com/richardkiss/pycoin), originally written and maintained by Richard Kiss, is a Python-based library that supports manipulation of bitcoin keys and transactions, even supporting the scripting language enough to properly deal with nonstandard transactions.

The pycoin library supports both Python 2 (2.7.x) and Python 3 (3.3 and later) and comes with some handy command-line utilities, ku and tx.

# Key Utility (KU)

The command-line utility ku ("key utility") is a Swiss Army knife for manipulating keys. It supports BIP-32 keys, WIF, and addresses (bitcoin and alt coins). Following are some examples.

Create a BIP-32 key using the default entropy sources of GPG and */dev/random*:

$ ku create  
  
input : create  
network : Bitcoin  
wallet key : xprv9s21ZrQH143K3LU5ctPZTBnb9kTjA5Su9DcWHvXJemiJBsY7VqXUG7hipgdWaU  
 m2nhnzdvxJf5KJo9vjP2nABX65c5sFsWsV8oXcbpehtJi  
public version : xpub661MyMwAqRbcFpYYiuvZpKjKhnJDZYAkWSY76JvvD7FH4fsG3Nqiov2CfxzxY8  
 DGcpfT56AMFeo8M8KPkFMfLUtvwjwb6WPv8rY65L2q8Hz  
tree depth : 0  
fingerprint : 9d9c6092  
parent f'print : 00000000  
child index : 0  
chain code : 80574fb260edaa4905bc86c9a47d30c697c50047ed466c0d4a5167f6821e8f3c  
private key : yes  
secret exponent : 112471538590155650688604752840386134637231974546906847202389294096567806844862  
 hex : f8a8a28b28a916e1043cc0aca52033a18a13cab1638d544006469bc171fddfbe  
wif : L5Z54xi6qJusQT42JHA44mfPVZGjyb4XBRWfxAzUWwRiGx1kV4sP  
 uncompressed : 5KhoEavGNNH4GHKoy2Ptu4KfdNp4r56L5B5un8FP6RZnbsz5Nmb  
public pair x : 76460638240546478364843397478278468101877117767873462127021560368290114016034  
public pair y : 59807879657469774102040120298272207730921291736633247737077406753676825777701  
 x as hex : a90b3008792432060fa04365941e09a8e4adf928bdbdb9dad41131274e379322  
 y as hex : 843a0f6ed9c0eb1962c74533795406914fe3f1957c5238951f4fe245a4fcd625  
y parity : odd  
key pair as sec : 03a90b3008792432060fa04365941e09a8e4adf928bdbdb9dad41131274e379322  
 uncompressed : 04a90b3008792432060fa04365941e09a8e4adf928bdbdb9dad41131274e379322  
 843a0f6ed9c0eb1962c74533795406914fe3f1957c5238951f4fe245a4fcd625  
hash160 : 9d9c609247174ae323acfc96c852753fe3c8819d  
 uncompressed : 8870d869800c9b91ce1eb460f4c60540f87c15d7  
Bitcoin address : 1FNNRQ5fSv1wBi5gyfVBs2rkNheMGt86sp  
 uncompressed : 1DSS5isnH4FsVaLVjeVXewVSpfqktdiQAM

Create a BIP-32 key from a passphrase:

The passphrase in this example is way too easy to guess.

$ ku P:foo  
  
input : P:foo  
network : Bitcoin  
wallet key : xprv9s21ZrQH143K31AgNK5pyVvW23gHnkBq2wh5aEk6g1s496M8ZMjxncCKZKgb5j  
 ZoY5eSJMJ2Vbyvi2hbmQnCuHBujZ2WXGTux1X2k9Krdtq  
public version : xpub661MyMwAqRbcFVF9ULcqLdsEa5WnCCugQAcgNd9iEMQ31tgH6u4DLQWoQayvtS  
 VYFvXz2vPPpbXE1qpjoUFidhjFj82pVShWu9curWmb2zy  
tree depth : 0  
fingerprint : 5d353a2e  
parent f'print : 00000000  
child index : 0  
chain code : 5eeb1023fd6dd1ae52a005ce0e73420821e1d90e08be980a85e9111fd7646bbc  
private key : yes  
secret exponent : 65825730547097305716057160437970790220123864299761908948746835886007793998275  
 hex : 91880b0e3017ba586b735fe7d04f1790f3c46b818a2151fb2def5f14dd2fd9c3  
wif : L26c3H6jEPVSqAr1usXUp9qtQJw6NHgApq6Ls4ncyqtsvcq2MwKH  
 uncompressed : 5JvNzA5vXDoKYJdw8SwwLHxUxaWvn9mDea6k1vRPCX7KLUVWa7W  
public pair x : 81821982719381104061777349269130419024493616650993589394553404347774393168191  
public pair y : 58994218069605424278320703250689780154785099509277691723126325051200459038290  
 x as hex : b4e599dfa44555a4ed38bcfff0071d5af676a86abf123c5b4b4e8e67a0b0b13f  
 y as hex : 826d8b4d3010aea16ff4c1c1d3ae68541d9a04df54a2c48cc241c2983544de52  
y parity : even  
key pair as sec : 02b4e599dfa44555a4ed38bcfff0071d5af676a86abf123c5b4b4e8e67a0b0b13f  
 uncompressed : 04b4e599dfa44555a4ed38bcfff0071d5af676a86abf123c5b4b4e8e67a0b0b13f  
 826d8b4d3010aea16ff4c1c1d3ae68541d9a04df54a2c48cc241c2983544de52  
hash160 : 5d353a2ecdb262477172852d57a3f11de0c19286  
 uncompressed : e5bd3a7e6cb62b4c820e51200fb1c148d79e67da  
Bitcoin address : 19Vqc8uLTfUonmxUEZac7fz1M5c5ZZbAii  
 uncompressed : 1MwkRkogzBRMehBntgcq2aJhXCXStJTXHT

Get info as JSON:

$ ku P:foo -P -j

{  
 "y\_parity": "even",  
 "public\_pair\_y\_hex": "826d8b4d3010aea16ff4c1c1d3ae68541d9a04df54a2c48cc241c2983544de52",  
 "private\_key": "no",  
 "parent\_fingerprint": "00000000",  
 "tree\_depth": "0",  
 "network": "Bitcoin",  
 "btc\_address\_uncompressed": "1MwkRkogzBRMehBntgcq2aJhXCXStJTXHT",  
 "key\_pair\_as\_sec\_uncompressed": "04b4e599dfa44555a4ed38bcfff0071d5af676a86abf123c5b4b4e8e67a0b0b13f826d8b4d3010aea16ff4c1c1d3ae68541d9a04df54a2c48cc241c2983544de52",  
 "public\_pair\_x\_hex": "b4e599dfa44555a4ed38bcfff0071d5af676a86abf123c5b4b4e8e67a0b0b13f",  
 "wallet\_key": "xpub661MyMwAqRbcFVF9ULcqLdsEa5WnCCugQAcgNd9iEMQ31tgH6u4DLQWoQayvtSVYFvXz2vPPpbXE1qpjoUFidhjFj82pVShWu9curWmb2zy",  
 "chain\_code": "5eeb1023fd6dd1ae52a005ce0e73420821e1d90e08be980a85e9111fd7646bbc",  
 "child\_index": "0",  
 "hash160\_uncompressed": "e5bd3a7e6cb62b4c820e51200fb1c148d79e67da",  
 "btc\_address": "19Vqc8uLTfUonmxUEZac7fz1M5c5ZZbAii",  
 "fingerprint": "5d353a2e",  
 "hash160": "5d353a2ecdb262477172852d57a3f11de0c19286",  
 "input": "P:foo",  
 "public\_pair\_x": "81821982719381104061777349269130419024493616650993589394553404347774393168191",  
 "public\_pair\_y": "58994218069605424278320703250689780154785099509277691723126325051200459038290",  
 "key\_pair\_as\_sec": "02b4e599dfa44555a4ed38bcfff0071d5af676a86abf123c5b4b4e8e67a0b0b13f"  
}

Public BIP32 key:

$ ku -w -P P:foo  
xpub661MyMwAqRbcFVF9ULcqLdsEa5WnCCugQAcgNd9iEMQ31tgH6u4DLQWoQayvtSVYFvXz2vPPpbXE1qpjoUFidhjFj82pVShWu9curWmb2zy

Generate a subkey:

$ ku -w -s3/2 P:foo  
xprv9wTErTSkjVyJa1v4cUTFMFkWMe5eu8ErbQcs9xajnsUzCBT7ykHAwdrxvG3g3f6BFk7ms5hHBvmbdutNmyg6iogWKxx6mefEw4M8EroLgKj

Hardened subkey:

$ ku -w -s3/2H P:foo  
xprv9wTErTSu5AWGkDeUPmqBcbZWX1xq85ZNX9iQRQW9DXwygFp7iRGJo79dsVctcsCHsnZ3XU3DhsuaGZbDh8iDkBN45k67UKsJUXM1JfRCdn1

WIF:

$ ku -W P:foo  
L26c3H6jEPVSqAr1usXUp9qtQJw6NHgApq6Ls4ncyqtsvcq2MwKH

Address:

$ ku -a P:foo  
19Vqc8uLTfUonmxUEZac7fz1M5c5ZZbAii

Generate a bunch of subkeys:

$ ku P:foo -s 0/0-5 -w  
xprv9xWkBDfyBXmZjBG9EiXBpy67KK72fphUp9utJokEBFtjsjiuKUUDF5V3TU8U8cDzytqYnSekc8bYuJS8G3bhXxKWB89Ggn2dzLcoJsuEdRK  
xprv9xWkBDfyBXmZnzKf3bAGifK593gT7WJZPnYAmvc77gUQVej5QHckc5Adtwxa28ACmANi9XhCrRvtFqQcUxt8rUgFz3souMiDdWxJDZnQxzx  
xprv9xWkBDfyBXmZqdXA8y4SWqfBdy71gSW9sjx9JpCiJEiBwSMQyRxan6srXUPBtj3PTxQFkZJAiwoUpmvtrxKZu4zfsnr3pqyy2vthpkwuoVq  
xprv9xWkBDfyBXmZsA85GyWj9uYPyoQv826YAadKWMaaEosNrFBKgj2TqWuiWY3zuqxYGpHfv9cnGj5P7e8EskpzKL1Y8Gk9aX6QbryA5raK73p  
xprv9xWkBDfyBXmZv2q3N66hhZ8DAcEnQDnXML1J62krJAcf7Xb1HJwuW2VMJQrCofY2jtFXdiEY8UsRNJfqK6DAdyZXoMvtaLHyWQx3FS4A9zw  
xprv9xWkBDfyBXmZw4jEYXUHYc9fT25k9irP87n2RqfJ5bqbjKdT84Mm7Wtc2xmzFuKg7iYf7XFHKkSsaYKWKJbR54bnyAD9GzjUYbAYTtN4ruo

Generate the corresponding addresses:

$ ku P:foo -s 0/0-5 -a  
1MrjE78H1R1rqdFrmkjdHnPUdLCJALbv3x  
1AnYyVEcuqeoVzH96zj1eYKwoWfwte2pxu  
1GXr1kZfxE1FcK6ZRD5sqqqs5YfvuzA1Lb  
116AXZc4bDVQrqmcinzu4aaPdrYqvuiBEK  
1Cz2rTLjRM6pMnxPNrRKp9ZSvRtj5dDUML  
1WstdwPnU6HEUPme1DQayN9nm6j7nDVEM

Generate the corresponding WIFs:

$ ku P:foo -s 0/0-5 -W  
L5a4iE5k9gcJKGqX3FWmxzBYQc29PvZ6pgBaePLVqT5YByEnBomx  
Kyjgne6GZwPGB6G6kJEhoPbmyjMP7D5d3zRbHVjwcq4iQXD9QqKQ  
L4B3ygQxK6zH2NQGxLDee2H9v4Lvwg14cLJW7QwWPzCtKHdWMaQz  
L2L2PZdorybUqkPjrmhem4Ax5EJvP7ijmxbNoQKnmTDMrqemY8UF  
L2oD6vA4TUyqPF8QG4vhUFSgwCyuuvFZ3v8SKHYFDwkbM765Nrfd  
KzChTbc3kZFxUSJ3Kt54cxsogeFAD9CCM4zGB22si8nfKcThQn8C

Check that it works by choosing a BIP32 string (the one corresponding to subkey 0/3):

$ ku -W xprv9xWkBDfyBXmZsA85GyWj9uYPyoQv826YAadKWMaaEosNrFBKgj2TqWuiWY3zuqxYGpHfv9cnGj5P7e8EskpzKL1Y8Gk9aX6QbryA5raK73p  
L2L2PZdorybUqkPjrmhem4Ax5EJvP7ijmxbNoQKnmTDMrqemY8UF  
$ ku -a xprv9xWkBDfyBXmZsA85GyWj9uYPyoQv826YAadKWMaaEosNrFBKgj2TqWuiWY3zuqxYGpHfv9cnGj5P7e8EskpzKL1Y8Gk9aX6QbryA5raK73p  
116AXZc4bDVQrqmcinzu4aaPdrYqvuiBEK

Yep, looks familiar.

From secret exponent:

$ ku 1  
  
input : 1  
network : Bitcoin  
secret exponent : 1  
 hex : 1  
wif : KwDiBf89QgGbjEhKnhXJuH7LrciVrZi3qYjgd9M7rFU73sVHnoWn  
 uncompressed : 5HpHagT65TZzG1PH3CSu63k8DbpvD8s5ip4nEB3kEsreAnchuDf  
public pair x : 55066263022277343669578718895168534326250603453777594175500187360389116729240  
public pair y : 32670510020758816978083085130507043184471273380659243275938904335757337482424  
 x as hex : 79be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798  
 y as hex : 483ada7726a3c4655da4fbfc0e1108a8fd17b448a68554199c47d08ffb10d4b8  
y parity : even  
key pair as sec : 0279be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798  
 uncompressed : 0479be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798  
 483ada7726a3c4655da4fbfc0e1108a8fd17b448a68554199c47d08ffb10d4b8  
hash160 : 751e76e8199196d454941c45d1b3a323f1433bd6  
 uncompressed : 91b24bf9f5288532960ac687abb035127b1d28a5  
Bitcoin address : 1BgGZ9tcN4rm9KBzDn7KprQz87SZ26SAMH  
 uncompressed : 1EHNa6Q4Jz2uvNExL497mE43ikXhwF6kZm

Litecoin version:

$ ku -nL 1  
  
input : 1  
network : Litecoin  
secret exponent : 1  
 hex : 1  
wif : T33ydQRKp4FCW5LCLLUB7deioUMoveiwekdwUwyfRDeGZm76aUjV  
 uncompressed : 6u823ozcyt2rjPH8Z2ErsSXJB5PPQwK7VVTwwN4mxLBFrao69XQ  
public pair x : 55066263022277343669578718895168534326250603453777594175500187360389116729240  
public pair y : 32670510020758816978083085130507043184471273380659243275938904335757337482424  
 x as hex : 79be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798  
 y as hex : 483ada7726a3c4655da4fbfc0e1108a8fd17b448a68554199c47d08ffb10d4b8  
y parity : even  
key pair as sec : 0279be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798  
 uncompressed : 0479be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798  
 483ada7726a3c4655da4fbfc0e1108a8fd17b448a68554199c47d08ffb10d4b8  
hash160 : 751e76e8199196d454941c45d1b3a323f1433bd6  
 uncompressed : 91b24bf9f5288532960ac687abb035127b1d28a5  
Litecoin address : LVuDpNCSSj6pQ7t9Pv6d6sUkLKoqDEVUnJ  
 uncompressed : LYWKqJhtPeGyBAw7WC8R3F7ovxtzAiubdM

Dogecoin WIF:

$ ku -nD -W 1  
QNcdLVw8fHkixm6NNyN6nVwxKek4u7qrioRbQmjxac5TVoTtZuot

From public pair (on Testnet):

$ ku -nT 55066263022277343669578718895168534326250603453777594175500187360389116729240,even  
  
input : 550662630222773436695787188951685343262506034537775941755001873603  
 89116729240,even  
network : Bitcoin testnet  
public pair x : 55066263022277343669578718895168534326250603453777594175500187360389116729240  
public pair y : 32670510020758816978083085130507043184471273380659243275938904335757337482424  
 x as hex : 79be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798  
 y as hex : 483ada7726a3c4655da4fbfc0e1108a8fd17b448a68554199c47d08ffb10d4b8  
y parity : even  
key pair as sec : 0279be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798  
 uncompressed : 0479be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798  
 483ada7726a3c4655da4fbfc0e1108a8fd17b448a68554199c47d08ffb10d4b8  
hash160 : 751e76e8199196d454941c45d1b3a323f1433bd6  
 uncompressed : 91b24bf9f5288532960ac687abb035127b1d28a5  
Bitcoin testnet address : mrCDrCybB6J1vRfbwM5hemdJz73FwDBC8r  
 uncompressed : mtoKs9V381UAhUia3d7Vb9GNak8Qvmcsme

From hash160:

$ ku 751e76e8199196d454941c45d1b3a323f1433bd6  
  
input : 751e76e8199196d454941c45d1b3a323f1433bd6  
network : Bitcoin  
hash160 : 751e76e8199196d454941c45d1b3a323f1433bd6  
Bitcoin address : 1BgGZ9tcN4rm9KBzDn7KprQz87SZ26SAMH

As a Dogecoin address:

$ ku -nD 751e76e8199196d454941c45d1b3a323f1433bd6  
  
input : 751e76e8199196d454941c45d1b3a323f1433bd6  
network : Dogecoin  
hash160 : 751e76e8199196d454941c45d1b3a323f1433bd6  
Dogecoin address : DFpN6QqFfUm3gKNaxN6tNcab1FArL9cZLE

## Transaction Utility (TX)

The command-line utility tx will display transactions in human-readable form, fetch base transactions from pycoin’s transaction cache or from web services (blockchain.info, blockcypher.com, blockr.io and chain.so are currently supported), merge transactions, add or delete inputs or outputs, and sign transactions.

Following are some examples.

View the famous "pizza" transaction:

$ tx 49d2adb6e476fa46d8357babf78b1b501fd39e177ac7833124b3f67b17c40c2a  
warning: consider setting environment variable PYCOIN\_CACHE\_DIR=~/.pycoin\_cache to cache transactions fetched via web services  
warning: no service providers found for get\_tx; consider setting environment variable PYCOIN\_BTC\_PROVIDERS  
usage: tx [-h] [-t TRANSACTION\_VERSION] [-l LOCK\_TIME] [-n NETWORK] [-a]  
 [-i address] [-f path-to-private-keys] [-g GPG\_ARGUMENT]  
 [--remove-tx-in tx\_in\_index\_to\_delete]  
 [--remove-tx-out tx\_out\_index\_to\_delete] [-F transaction-fee] [-u]  
 [-b BITCOIND\_URL] [-o path-to-output-file]  
 argument [argument ...]  
tx: error: can't find Tx with id 49d2adb6e476fa46d8357babf78b1b501fd39e177ac7833124b3f67b17c40c2a

Oops! We don’t have web services set up. Let’s do that now:

$ PYCOIN\_CACHE\_DIR=~/.pycoin\_cache  
$ PYCOIN\_BTC\_PROVIDERS="block.io blockchain.info blockexplorer.com"  
$ export PYCOIN\_CACHE\_DIR PYCOIN\_BTC\_PROVIDERS

It’s not done automatically so a command-line tool won’t leak potentially private information about what transactions you’re interested in to a third-party website. If you don’t care, you could put these lines into your *.profile*.

Let’s try again:

$ tx 49d2adb6e476fa46d8357babf78b1b501fd39e177ac7833124b3f67b17c40c2a  
Version: 1 tx hash 49d2adb6e476fa46d8357babf78b1b501fd39e177ac7833124b3f67b17c40c2a 159 bytes  
TxIn count: 1; TxOut count: 1  
Lock time: 0 (valid anytime)  
Input:  
 0: (unknown) from 1e133f7de73ac7d074e2746a3d6717dfc99ecaa8e9f9fade2cb8b0b20a5e0441:0  
Output:  
 0: 1CZDM6oTttND6WPdt3D6bydo7DYKzd9Qik receives 10000000.00000 mBTC  
Total output 10000000.00000 mBTC  
including unspents in hex dump since transaction not fully signed  
010000000141045e0ab2b0b82cdefaf9e9a8ca9ec9df17673d6a74e274d0c73ae77d3f131e000000004a493046022100a7f26eda874931999c90f87f01ff1ffc76bcd058fe16137e0e63fdb6a35c2d78022100a61e9199238eb73f07c8f209504c84b80f03e30ed8169edd44f80ed17ddf451901ffffffff010010a5d4e80000001976a9147ec1003336542cae8bded8909cdd6b5e48ba0ab688ac00000000  
  
\*\* can't validate transaction as source transactions missing

The final line appears because to validate the transactions' signatures, you technically need the source transactions. So let’s add -a to augment the transactions with source information:

$ tx -a 49d2adb6e476fa46d8357babf78b1b501fd39e177ac7833124b3f67b17c40c2a  
warning: transaction fees recommendations casually calculated and estimates may be incorrect  
warning: transaction fee lower than (casually calculated) expected value of 0.1 mBTC, transaction might not propagate  
Version: 1 tx hash 49d2adb6e476fa46d8357babf78b1b501fd39e177ac7833124b3f67b17c40c2a 159 bytes  
TxIn count: 1; TxOut count: 1  
Lock time: 0 (valid anytime)  
Input:  
 0: 17WFx2GQZUmh6Up2NDNCEDk3deYomdNCfk from 1e133f7de73ac7d074e2746a3d6717dfc99ecaa8e9f9fade2cb8b0b20a5e0441:0 10000000.00000 mBTC sig ok  
Output:  
 0: 1CZDM6oTttND6WPdt3D6bydo7DYKzd9Qik receives 10000000.00000 mBTC  
Total input 10000000.00000 mBTC  
Total output 10000000.00000 mBTC  
Total fees 0.00000 mBTC  
  
010000000141045e0ab2b0b82cdefaf9e9a8ca9ec9df17673d6a74e274d0c73ae77d3f131e000000004a493046022100a7f26eda874931999c90f87f01ff1ffc76bcd058fe16137e0e63fdb6a35c2d78022100a61e9199238eb73f07c8f209504c84b80f03e30ed8169edd44f80ed17ddf451901ffffffff010010a5d4e80000001976a9147ec1003336542cae8bded8909cdd6b5e48ba0ab688ac00000000  
  
all incoming transaction values validated

Now, let’s look at unspent outputs for a specific address (UTXO). In block #1, we see a coinbase transaction to 12c6DSiU4Rq3P4ZxziKxzrL5LmMBrzjrJX. Let’s use fetch\_unspent to find all coins in this address:

$ fetch\_unspent 12c6DSiU4Rq3P4ZxziKxzrL5LmMBrzjrJX  
a3a6f902a51a2cbebede144e48a88c05e608c2cce28024041a5b9874013a1e2a/0/76a914119b098e2e980a229e139a9ed01a469e518e6f2688ac/333000  
cea36d008badf5c7866894b191d3239de9582d89b6b452b596f1f1b76347f8cb/31/76a914119b098e2e980a229e139a9ed01a469e518e6f2688ac/10000  
065ef6b1463f552f675622a5d1fd2c08d6324b4402049f68e767a719e2049e8d/86/76a914119b098e2e980a229e139a9ed01a469e518e6f2688ac/10000  
a66dddd42f9f2491d3c336ce5527d45cc5c2163aaed3158f81dc054447f447a2/0/76a914119b098e2e980a229e139a9ed01a469e518e6f2688ac/10000  
ffd901679de65d4398de90cefe68d2c3ef073c41f7e8dbec2fb5cd75fe71dfe7/0/76a914119b098e2e980a229e139a9ed01a469e518e6f2688ac/100  
d658ab87cc053b8dbcfd4aa2717fd23cc3edfe90ec75351fadd6a0f7993b461d/5/76a914119b098e2e980a229e139a9ed01a469e518e6f2688ac/911  
36ebe0ca3237002acb12e1474a3859bde0ac84b419ec4ae373e63363ebef731c/1/76a914119b098e2e980a229e139a9ed01a469e518e6f2688ac/100000  
fd87f9adebb17f4ebb1673da76ff48ad29e64b7afa02fda0f2c14e43d220fe24/0/76a914119b098e2e980a229e139a9ed01a469e518e6f2688ac/1  
dfdf0b375a987f17056e5e919ee6eadd87dad36c09c4016d4a03cea15e5c05e3/1/76a914119b098e2e980a229e139a9ed01a469e518e6f2688ac/1337  
cb2679bfd0a557b2dc0d8a6116822f3fcbe281ca3f3e18d3855aa7ea378fa373/0/76a914119b098e2e980a229e139a9ed01a469e518e6f2688ac/1337  
d6be34ccf6edddc3cf69842dce99fe503bf632ba2c2adb0f95c63f6706ae0c52/1/76a914119b098e2e980a229e139a9ed01a469e518e6f2688ac/2000000  
 0e3e2357e806b6cdb1f70b54c3a3a17b6714ee1f0e68bebb44a74b1efd512098/0/410496b538e853519c726a2c91e61ec11600ae1390813a627c66fb8be7947be63c52da7589379515d4e0a604f8141781e62294721166bf621e73a82cbf2342c858eeac/5000000000