CS480001 / SEC532

Blockchain: Security and Applications

Homework 1

You happen to know most of the WIF of a secret key in the Bitcoin Test Network:

cUJjSe7q8Zm0kxiQE2fC8c400T1cKJow511jc8yZJhLf9E9PLi1z

As you see, some of the digits are 0 (Zero) in the WIF. This cannot happen since this must be in base58. The digits marked with 0 are missing.

a. Find the WIF and corresponding Bitcoin address and print these. The code below is the starting point – you are not allowed to use any other imports for this part (**if you do your submission will not be evaluated**). Obviously, you need to use a search engine to find information on which functions you need to call from these python modules to perform the task. This is allowed. The rest of the information you need is in the slides (download the updated version).
b. By using python-bitcoinlib, create a Bitcoin Test Network account for yourself (see the accompanying CoLab file to learn how to do this and rest of the tasks).
c. Send all the BTC (except the fee) to the address you found in part (a) via a P2PKH transaction.
d. Spend the UTXO you created (since you know the private key you can unlock it) and send it back to the faucet.

Name your CoLab file as name_surname.ipynb and submit until the deadline.

Do not show your code to anyone or discuss your solution. You need to do everything by yourself.

```
!pip install base58
!pip install ecdsa

import base58
import hashlib
from ecdsa import VerifyingKey, SECP256k1, SigningKey

wif = "cUJjSe7q8Zm0kxiQE2fC8c400T1cKJow511jc8yZJhLf9E9PLi1z
";

#this is hash160 – do not modify
def hash160(hex_str):
    sha = hashlib.sha256()
    rip = hashlib.new('ripemd160')
    sha.update(hex_str)
    rip.update( sha.digest() )
    return rip.hexdigest()
```

#Complete the rest and find the Bitcoin Address