Homework 3: Transactions in Bitcoin

Deadline: January 11, 2023

1. Administration

Each homework in this course will contribute 20 % to the final grade. There will be 4 homework assignments and one final project. If needed, we might have a bonus assignment, which can replace your worst homework, but not the final project.

Your solution should be sent to the following email as a text file:

- domagojvrgoc@gmail.com

There is no restriction or penalty for using materials you found online. It is a good practice to mention those if you use them. This will not result in any penalty to your homework. If you understand how to use a commercial Bitcoin wallet to resolve the homework there will be no penalty either. You only need to make sure to send me the data specified below.

2. The homework

In this homework, we will create various P2PKH addresses and generate transactions to and from these addresses on the Bitcoin testnet network. In essence, the homework will simulate what a Bitcoin wallet does when working with P2PKH (and P2SH) addresses, and when it sends funds to an address that is of type P2PKH and P2SH.

Together with the homework, you will receive a minimal implementation allowing to create and validate transactions with P2PKH scripts (also P2PK, but we will not use these to be nice to the miners). The implementation is a simplification of what we saw in the class (it supports only P2PKH and not full P2SH support), and will help you resolve the homework. The implementation also includes an example of how to send funds from one PO2PKH address to another, or how to send funds from a P2PKH address to a P2SH address.

The specific tasks you need to resolve are as follows

1. **Receive testcoins [3 points]**. First thing you need to do is generate *two* Bitcoin **Testnet** addresses of the type P2PKH. The secret keys should be as follows:

   - `hash256(b'YourName1')`

- `hash256(b'YourName2')`,

where the substring ”YourName”should be replaced by your name (without special symbols; e.g. č would be replaced by c, etc.). For the rest of the homework, we will refer to these addresses as **address 1** and **address 2**.

The addresses can be generated by your solution for Homework 2. If you do not have it, the code also includes the file ”hw2_solution.py”which allows you to do this.

Once the addresses have been generated, you should send testcoins from a testnet faucet in **two** different transactions to **address 1**. A faucet you can use is, for example, `https://testnet-faucet.com/btc-testnet/`[1].

For this part of the homework, you should hand in: (i) your secret keys; and (ii) the generated addresses. Using the block explorer `https://live.blockcypher.com/`, we will validate that you received funds in two different transactions.

2. **Transaction P2PKH to P2PKH** [**6 points**]. Now that you have two addresses, and two transactions received by **address 1**, you should generate **a single transaction** that spends these two outputs received by **address 1**, and sends them to **address 2**. The two outputs need to be spent in a single transaction! You can send the transaction to the network by using `https://live.blockcypher.com/btc/pushtx/` (using the Bitcoin testnet network). Do not forget the transaction fee!

   For this part of the homework, you should send me the hexadecimal version of the (entire) transaction that was **accepted** by the testnet network. In our code, you can generate this hex using `newTx.serialize().hex()`, after you added all the required signatures. To receive the credit for this part of the homework, you need to make sure that the funds reached **address 2**. Again, you can check this in a block explorer. We will review the status of your transaction using its hex, so you need to make sure that we can do this correctly. The final two lines in the file ”txP2PKH.py” shows you how we will look for the transaction in the block explorer.

3. **Transaction P2PKH to P2PKH and P2SH** [**6 points**]. Now that the funds are controlled by **address 2**, we should send it to two different addresses in a single transaction.

   For this, you first need to generate your own P2SH address, whose redeem script is simply a P2PKH script for **address 1**, as shown in class. We will call this P2SH address **address 3**.

   You should now generate a single transaction that spends the output generated in part 2 of the homework, and sends 50 % of the funds to **address 3**, and the other 50 % to the address `mohjSavDdQYHRYXcS3uS6ttaHP8amyvX78` (the faucet). The 50 % are modulo the transaction fee (you determine it). Again, it is important to stress that **address**

[1] The faucet `https://bitcoinfaucet.uo1.net/send.php` that we used for Homework 2 no longer supports base58 addresses; you can code bench32 if you so desire.

3 is a P2SH address, and not a P2PKH address! You can send the transaction to the network by using `https://live.blockcypher.com/btc/pushtx/` (using the Bitcoin testnet network).

Here you should send me the same thing as in part 2; namely: the hex of the transaction. Additionally, you should send me **address 3**.

4. **Transaction P2PKH to P2SH** [**5 point**]. Finally, you should return the funds received by **address 3** in part 3 of the homework to `mohjSavDdQYHRYXcS3uS6ttaHP8amyvX78`. This is a transaction with only one input and one output, but you need to generate a P2SH spend (as shown in the code in class).

   Here you should send me the same thing as in parts 2 and 3; namely, the hex of the entire transaction.

Bonus 1. [**an extra homework**] This bonus replaces an entire homework. Namely, if you solve this one, it either replaces the worst homework assignment you handed in, or the next homework. For this, you are asked to support bench32 addresses, and be able to send funds to them. In particular, you should read addresses that start with `tb1`, determine whether it's a P2PKH or a P2SH address, and be able to send the funds to it. For this, you might also need to figure out the basics of SegWit (last link below). To test your implementation, you can use the `https://bitcoinfaucet.uo1.net/send.php` faucet. Places to start reading about bench32 are:

- `https://en.bitcoin.it/wiki/List_of_address_prefixes`
- `https://en.bitcoin.it/wiki/Bech32`
- `https://en.bitcoin.it/wiki/BIP_0173`
- `https://github.com/jimmysong/programmingbitcoin/blob/master/ch13.asciidoc`

Bonus 2. [**+10 Karma**] Return the testcoins you received in Homework 2 to the faucet. The code in Homework 2 contains the secret key you need to sign the transactions.