CS 251: Bitcoin and Crypto Currencies Fall 2015

Project 1

due: October 12, 11:59 PM via email to cs251.stanford.fall.2015@gmail.com

Introduction

In this assignment you'll create several transactions and post them to the Bitcoin blockchain. We will provide starter code for this using `bitcoinj`, a free and popular Java library for interacting with Bitcoin. You are free to create and post the transactions using an alternate library and language if you want, but you must submit your code in any case.

Getting started

1. Download the code from the course website and and import it into your favorite IDE. You can use maven to download the required dependencies.
2. Familiarize yourself with Bitcoin's scripting system.
3. Make yourself familiar with the `bitcoinj` API and the starter code. You should check out the `ScriptTransaction` class and the example in `PayToPubKey`.
4. Implement code for the exercises below, using the Bitcoin test network ("testnet") to test your code (as well as offline testing). You can get free testnet coins from http://tpfaucet.appspot.com/. It is courteous to send the testnet coins back to the faucet after you are done experimenting with them.
5. You must implement the transactions by specifying the Scripts in the specific subclasses. You will not receive credit if you create the transactions using a different tool. We will test your implementation.
6. Email the TA list (cs251ta@cs) to receive some bitcoin that you can play around with. In the Email, please provide a Bitcoin address that you own.
7. You can use the transaction hashes to track your transactions on a block explorer tool such as https://test-insight.bitpay.com/ (testnet) or https://insight.bitpay.com/ (mainnet).
8. **Important**: The transaction for exercise 1 should be done on the **mainnet.** The transactions for exercises 2 and 3 may be done on the **testnet** (you can also do them on mainnet if you want, but you will need to submit them directly a mining pool such as Eligius which allows non-standard transactions).

Submission: For all exercises, submit the source code as well as the transaction hashes. Please create a single tar or zip file that includes all your deliverables for all three exercises and email it to the address at the top of this page.

Exercises

1. Generate an address whose standard Base58Check representation starts with 1 and then at least the first four letters of your surname in lowercase (if your surname is shorter than four letters, please append as many 'x' characters as necessary). You may generate this address

either using `bitcoinj` or using an external generator. Send some bitcoins to this address using a standard Pay2PubKeyHash transaction and then redeem them.

2. Generate a transaction that can be redeemed by the solution (x,y) to the following system of two linear equations:

   x+y = (first half of your suid) and x-y = (second half or your suid)

   [to ensure that an integer solution exists, please change the last digit of the two numbers on the right hand side so the numbers are both even or both odd]
   Create and redeem the transaction. The redemption script should be as small as possible. That is, a valid script sig should consist of simply pushing two integers *x* and *y* to the stack.

3. Generate a multi-sig transaction involving four parties such that the transaction can be redeemed by the first party (bank) combined with any one of the 3 others (customers) but not by only the customers or only the bank. Create and redeem the transaction and make sure that the script is as small as possible. You can use any legal combination of signatures to redeem the transaction but make sure that all combinations would have worked