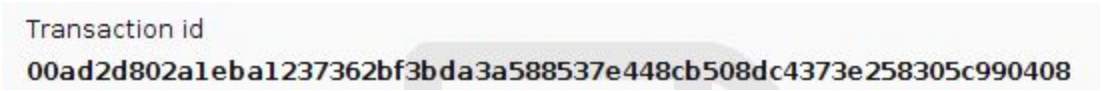
Muthu Chidambaram
CS 4501
Assignment 1

**Problem 1.**

a. The transaction ID was listed in MultiBit's payment information:




b. The transaction fee is 0.1 mB/KB (rounded up) => **0.1 mB**, or 0.1*0.23092 = **0.02 USD.**
c. My transaction was included in block #372138, whose output total was **14,487.20964744 BTC**.
d. According to blockchain.info, the transfer was included after roughly 17 minutes (6 confirmations), which indicates that it took about **8.5 minutes** to obtain 3 confirmations (assuming confirmations are uniformly distributed).

**Problem 2.**

a. Using Block Explorer, I identified the following addresses as likely to be those of my classmates: **1D6qsGhZqrRSKegqWT3e8TPR8gGczTxMLu, 17GBpDP8yHTZcBJ9WmmRMjCmgiqacy5v3n, 1AcKeSkKponv5qeZuEHvkocELf1Uo4ggCE.**
b. By repeatedly following back-links, I was able to trace the transaction back to the ID **1GymPp2dave9ByFRcFAwkPX3bMeVyoZkPq**. I am unsure as to how to determine what bitcoin exchange this ID is associated with.
c. The only geographical information I could ascertain was that of the relaying IP, which appeared to be from some location in Aurora, Colorado.

**Problem 3.**

a. A malicious developer might set up an evil wallet such that all users of the wallet software would have their wallet words automatically sent to the malicious developer, which would then allow him or her to spend the users' bitcoins. Alternatively, a malicious developer could set up a wallet to extract a small fee from each transaction, which is then sent to one of the developer's addresses (would only really work on BTC users who don't bother to check their transaction details).
b. I'm reasonably confident that my wallet is safe due to the high min-entropy of the generated wallet words associated with my wallet. However, if I were to put more money into my bitcoin wallet, I would definitely consider creating an offline storage wallet for savings. Additionally, I would also encrypt my wallet and take multiple backups to try to actively avoid a single point of failure.

**Problem 4.**

a. I verified the modulus using Python, as follows:

```
>>> p = int("FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFC2F", 16)
>>> p2 = 2**256-2**32-2**9-2**8-2**7-2**6-2**4-1
>>> p == p2
True
>>>
```

**Problem 5.**

a. If you're a multi-billionaire planning on sending money to an address generated by keypair.go, you first and foremost need to trust that the private key used to generate the public key was created with sufficiently high min entropy and that it is both undisclosed and securely stored. You need to make sure that no one else has access to the private key that was generated, and that the algorithm used to generate the public key from the private key is not reversible.

**Problem 6.**

a. My vanity function is as follows:

```
func generateVanityAddress(pattern string) (*btcec.PublicKey, *btcec.PrivateKey, *btcutil.AddressPubKeyHash) {
    //Generate vanity address that matches a pattern
    pub, priv := generateKeyPair()
    addr := generateAddr(pub)
    matched, err := regexp.MatchString(pattern, addr.String())

    for !matched && err == nil {
        pub, priv = generateKeyPair()
        addr = generateAddr(pub)
        matched, err = regexp.MatchString(pattern, addr.String())
    }

    if err != nil {
        log.Fatal(err)
    }

    return pub, priv, addr
}
```

**Problem 7.**

a. Using my function, I generated an address containing the string "fed" in order to showcase my support for Federer over Djokovic in the 2015 US Open finals:

```
This is the public key in hex:  [03f6d479452d6a0031b62047ea14249a0be40af8c969a35996a844f9d45e09a644]
This is the vanity address:     [1Dq8imCdm2vfyPbVCa8Pd3p8fed6YDeMgu]
```

**Problem 8.**

a. The vanity address, should, in theory, be just as secure as any other well generated address. This is due to the fact that the vanity address is just another randomly generated address that happens to match a specified pattern string, which should give it the same properties as a regular address.

**Problem 9.**

a. I transferred 0.1 mBTC to my vanity address, with the transaction ID being:

Transaction id

26fb15942eeff017f71a7fe9666590af2407010e5991cee37cbe35efeb1074fd

**Problem 10.**

a. Initially, I tried to transfer the 0.1 mBTC in my vanity address to a potential classmate's address, but I received this error:

```
Here is your raw bitcoin transaction:
0100000001fd7410ebef35be7ce3ce91590e010724af906566e97f1af717f0ef2e9415fb26010000
006a47304402202a104b666ec806ebeb0328b2fe669d333caaac5370d9fd6de0a4a05a530bb87502
205538c7fbfa5d4b496c62d17e68edfab311cd5374d3346bf35f385264987f3d43012103f6d47945
2d6a0031b62047ea14249a0be40af8c969a35996a844f9d45e09a644ffffffff0100000000000000
001976a914696741592c313e6d7f420614213aff226be647fa88ac00000000
Sending transaction to: https://insight.bitpay.com/api/tx/send
The sending api responded with:
Transaction rejected by network (code -26). Reason: 64: dust
```

Some StackOverflow sleuthing led me to believe that this error was due to the transaction being considered spam (0.1 mBTC is apparently too small). Thus, I transferred an additional 1 mBTC from my main address to my vanity address. Afterwards, my transaction was successful:

```
Here is your raw bitcoin transaction:
01000000010b7bc898a0b0cf747841d41f6c3dc29d0b9aec5492604f945184bacd9263005a010000
006b483045022100fbbcc91bb311b90526150bbcb2845740d341391294ef57bd341683d8e652aeaa
02207a42bc45c190b2d3f6202bdf879772d531eb67c13421dd075cf6f3fcca38fc28012103f6d479
452d6a0031b62047ea14249a0be40af8c969a35996a844f9d45e09a644ffffffff01905f01000000
00001976a914696741592c313e6d7f420614213aff226be647fa88ac00000000
Sending transaction to: https://insight.bitpay.com/api/tx/send
The sending api responded with:
{"txid":"e3a0f74aa9229b5a80f8a0114387faf2d10e22d1e8c115eb921bce002cd33a4f"}
```

**Problem 11.**

a. Using my modified spend code, I sent 50,000 Satoshis (0.4 mBTC after transaction fee) to my main account, as shown below:


b. My modified spend code (mspend.go) is attached with this assignment. Initially, I did not account for the fact that I had to manually specify a change address. This was only brought to my attention after looking at the transaction, which showed a transfer of 0.4 mBTC to my main account, but a loss of 1 mBTC from my vanity address.

**Problem 12.**

a. My attempt at double spending the same bitcoin was promptly shut down by the bitpay API, as shown below:

```
Sending transaction to: https://insight.bitpay.com/api/tx/send
The sending api responded with:
transaction already in block chain (code -27)
```