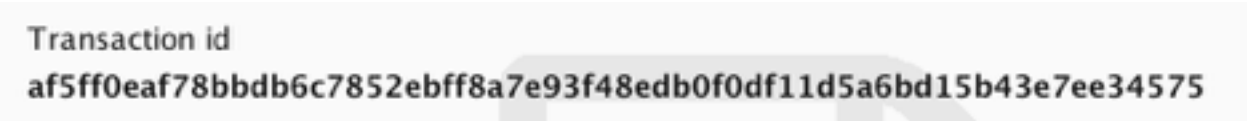
1.

a)



b) 0.0001 BTC
c) 1,755.15212027 BTC
d) Roughly 30 minutes for 3 confirmations because a transaction gets roughly 1 confirmation every 10 minutes.

2.

a)
4d894aa7bc7e8c012ab00348adb356a573ca8c20657156b8755d691c5475b6e7

539518e2dbf578f70a846eb63f920ccb10dc3f7e41321a68fa1c6e9324f10a9c

aba8afc58f31a6f4f8e4e4dc72bd7a20c1bc922935a91792ec1b1695020f9896

b)
Address:
1AxtTbSeLopt9LkacZ9w8kmzqxLaFobhyj

At this address there are MANY inputs to creating the transaction my bitcoin was traced to and the bitcoin is so fragmented among its inputs that there was not clear path to continue tracking backwards further.

c) if you go to the transaction ID, you can see where on a map the given transaction originated from based on the IP address. The bitcoin in part b of this question is tied to somewhere near Frankfurt, Germany.
https://blockchain.info/tx/
1e0781b6ecbf226fd068ccd9a5c61324a9e566aef42d1c21173533dd18291fa5

Revision: I think that this map is tied to what block your transaction is a part of and who won the block and where they are, not the specific location the transaction originated.

3.

a) A malicious developer might generate wallet words in a way that they could reverse engineer them and gain access to your wallet so they could steal bitcoin out of it in small quantities. They could also find a clever way of tacking on to the mining fee so they steal a tiny portion of your bitcoin with every transaction you make.
b) I am somewhat confident the money in my bitcoin wallet is safe because I have a secret password, secret wallet words, and other means of protecting my wallet, as well as the fact that if someone tried to steal my bitcoin they would have to fool the block chain into thinking I made a verifiable transaction to their wallet, which is extremely difficult to do without a real, valid transaction taking place (ie. hacking into my wallet and sending it to themselves). I am only

somewhat confident in this because I personally don't trust the bitcoin system very much regardless to store any of my income. To increase my confidence in it, it would have to be regulated by someone I trust, which defeats the purpose of bitcoin's creation but that's the only way I would really trust it.

4.

I verified that it was correct manually by calculating the equation 2^256 - 2^32 - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1 using wolfram alpha and converting the hex value on line 909 (FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFC2F) into decimal using an online hex to decimal converter and then compared the results, which were the same.

5.

To send money using a key pair generated by keypair.go you need to trust :

- the random generator used to generate the private key was a good one (so private key isn't guessable
- the btcec is a secure package that is not generating any back doors to find out what your keys are
- that your computer is not being monitored or compromised so someone can see the private key that was just logged to your console
- that btcec has implemented elliptic curve cryptography correctly
- that reversing a discrete log is hard (since not proven it's technically just trust)

6.

```
func generateVanityAddress(pattern string) (*btcec.PublicKey, *btcec.PrivateKey){
	// In order to recieve coins we must generate a public/private key pair.
	pub, priv := generateKeyPair()
	// check if they match
	matches, err := regexp.MatchString(pattern,
hex.EncodeToString(pub.SerializeCompressed()))
	//stay in loop generating new public keys until one matches pattern
	for matches != true {
		matches, err = regexp.MatchString(pattern,
hex.EncodeToString(pub.SerializeCompressed()))
		pub, priv = generateKeyPair()
	}
	if err != nil {
		// There was an error. Log it and bail out
		log.Fatal(err)
	}

	return priv.PubKey(), priv
}
```

8.

My picking my own sequence I removed some of the randomness by restricting what my private key was allowed to generate as an address so I think my vanity address is less secure and less

anonymous than a randomly generated address. So even though it's cool, it seems stupid to generate a vanity address if you are really concerned about security.

9.

b552d0155efbcc48d84016bf6e7d42d17e488588c63b737b7d537d9a4dbde3ae

10.

sent to Kienan Adams
db3aeafd9d8ba47638c85042ca532b80a65e7864daf6082e464e0680c26a163f