Joseph Tobin
Cryptocurrency Cabal
Professor David Evans
11 November 2015

Problem Set 3

1a. 177avVGBEXMcoiHaoQoYj2dn8FVouvPxLd

1b. This address was in use from 2014-09-16 to 2014-10-15.

1c. It appears 30 victims paid.

1d. It appears they charged different amounts. Here are the amounts from three "randomly" selected transactions.

	Time	BTC	USD/BTC	USD
1.	2014-09-27	1.25	356.5	456.88
2.	2014-09-24	2.3	358.46	824.458
3.	2014-09-26	1.25	355.43	444.2875

1e. It appears the ransomist waited to receive all of the bitcoin at once, and then sent them out to various addresses at once. While the ransomists received one or two bitcoins at a time, he/she sent all of them out in much larger (and therefore fewer) transactions. When the ransomist is sending out these large transactions, it appear there is a cut as part of the bitcoins (although it is a varying amount relative to the large quantity being sent) is being sent to the same address.

1f. I found it interesting that the ransomist was careful to send the bitcoins to different addresses when they sent out bitcoin that was not part of their own cut. But they did not take the caution to send their own cut to different addresses, and therefore put themselves at a greater risk.

2a. In an effort to determine how anonymous bitcoin is, it was useful to conduct transactions with merchants in dark markets because these are the merchants who are attempting to be the most anonymous. While other transactors (such as the Gamblers as mentioned on page 4), advertise their public keys, these merchants are trying as hard as possible to have the customer know as little as possible. Therefore, to determine how anonymous bitcoin really is, it would be most effective to try to remove the anonymity of these merchants.

2b. Even if you are purchasing legal goods from a merchant, the profits from the transaction could be used to fund illegal activities for the merchant (such as capital for buying drugs). The merchant could also use the sales you provide for a money laundering scheme. Although you may think you are conducting a legal transaction, this can lead to illegal consequences as you're money can be tied into an illegal scheme.

3. This statement is not actually true because the bitcoin script has a "multisig" function where mutlisignature addresses need multiple private keys in order to transact funds. What is important to this function is that the individual private key holders do not need to know the private keys of the other signers in order to sign the transaction. Additionally, some multisig addresses are set up such that only 2 out of the 3 private key holders need to sign the transaction. Therefore, a collection of multisignature public keys in a single transaction could be coming from different entities who do not know the private keys of eachother.

4. Because HD wallets can create an unlimited amount of addresses to use, the change address will

always belong to the user.

5. Of the 2054 transactions in block 379818 (the block after the one containing Dave's transaction), only one transactions fit the given parameters that was in the amount of 1463971 satoshis. Therefore, with knowledge of BitMixer's model, almost anyone can connect the transactions that have been "mixed impatiently" because parameters are so specific and there are only so many addresses to guess from. (Please see p5.py).

6. By splitting the output among two forward addresses, it is much harder to connect transactions. When there is only one output address, we only have to find the addresses that receive an amount of bitcoin within the specified range. With two or more output addresses, we have to find all combinations of addresses within a block such that their sum is within the specified range. If we have multiple addresses across multiple blocks, it becomes even harder as the number of transactions increases and therefore the number of combinations increases. If you have n addresses, the computational time will take More specifically, The computational time is O(n*logn) because you have to compare every address in the list with every address in the list that comes after it (therefore two for loops, with the second one only comparing the addresses that come after the address in the first for loop).

7. In p7.py, I took all of the transactions associated with the addresses in suspects.txt such that the addresses in suspects.txt were the input. With all of these transactions, I found all of the unique addresses that were outputs and then ran them through the wallet explorer api. After printing out all of the results (which can be seen in suspect-outputs.txt), I found that the majority of the recognized transactions were going to exchanges, indicating that these suspects were trying to convert their cash. There are several still functioning exchanges listed, including "BTC-e.com". The most popular labels, though, are old/no longer exist: "MtGoxAndOThers", "BTC-e.com-old", and "BitPay.com-old".

8. Similarly to problem 7, I took all of the transactions associated with the suspect addresses such that the suspect addresses were the input. With all of these transactions, I found all of the addresses that were outputs and stored each one in a set associated with each individual suspect address. Then, I found the intersection of all of these sets with eachother and stored the unique addresses in a set (and textfile, popular-addresses.txt). This textfile contains all of the addresses that have been output in different transactions with at least two suspect addresses as inputs. 53 addresses were found, including `18dwCxqqmya2ckWjCgTYReYyRL6dZF6pzL`. Please see p8.py.

9. In problem 9 (please see p9.py), I took all of the suspect addresses, found all of the transactions where they were gave money to addresses contained within popular-addresses.txt (addresses that have received bitcoin from at least two suspect addresses). Then, I found the amount given the these popular addresses and stored that value in a map. After sorting the map, I placed it in popular-addresses-amount.txt.

10.2. I decided to run all of my programs for problem 7, 8, and 9 on a list of bitcoin addresses called "infamous-addresses.txt". This list contains the "9 infamous bitcoin addresses" in addition to the top ten list of bitcoin addresses with the most output as per Blockchain.info's list of popular addresses. First, I ran these addresses through the program created for problem 7 to find any known addresses that these accounts were sending bitcoin to. These addresses were placed in "infamous-outputs.txt". Popular addresses included: "BTC-e.com", "MtGoxAndOthers", "Bitcoin.de", "BTCC.com-old2", "SatoshiDice.com-original", "LuckyB.it", "DeepBit.net", "BitKonan.com", "Bitcoin.de-old", "Bter.com-old2". Then, I ran these addresses through the program created for problem 8 to find any

addresses that received multiple outputs from these infamous addresses, and placed them in “popular-infamous-addresses.txt”. I found it interesting that a lot of the addresses were vanity addresses begininning with “bank”. Finally, I ran both “infamous-addresses.txt” and “popular-infamous-addresses.txt” through the program created for problem 9 to find the amount of bitcoin that each of the infamous addresses were sending to these popular recipients. I then stored the results in “popular-infamous-addresses-amounts.txt”. A significantly large amount of satoshis were stored in all of the previously mentioned “bank” accounts, as the 8 accounts who received the most bitcoin all began with “1bank” in their addres. A search on WalletExplorer.com revealed that these accounts are all linked to SatoshiDice.com, revealing that SatoshiDice.com is taking in an extraordinary amount of bitcoin.