BitSniffer: A tool for linkability analysis

Ryan Anderson, Luke Gessler, Sam Prestwood
{ldg3fa, rsa5bb, swp2sf}@virginia.edu
Crypto Currency Cabal, CS 4501-001, Fall 2015
University of Virginia

Abstract

We explore the idea of address linkability in two ways. Initially we looked at a specific use case for mixing coins, which involved developing a tool that clients of Bitcoin mixers could use to determine how anonymous the resulting coins were. Later, we moved towards the more general case of analyzing two addresses and developing a number of metrics to quantify how correlated they were.

Motivation

There are many ways for one to obfuscate the connection between two bitcoin addresses. However, there has been a lack of research on ways to determine how well linked two addresses are. We want to create a tool that can make this determination. We specifically attempted to address two use-cases for such a tool:

1. A privacy conscious individual uses a Bitcoin mixing service to obfuscate the connection between the Bitcoins they bought in an exchange and a different wallet address they use for pure-Bitcoin transactions. This tool would help the individual verify that the mixing service provided a satisfactory obfuscation.

2. An FBI agent has a ransom-ware address and an additional set of addresses which they think the ransomist used to transfer the “dirty” Bitcoin to. This tool could help the agent determine how like each address in the set is associated with the ransom-ware address.

To our knowledge, there is no publicly-available tool that addresses the former or the latter.

Background

Anonymity in Bitcoin

In the original Bitcoin whitepaper, Satoshi Nakamoto claimed that Bitcoin provides users with privacy by keeping the public key of a Bitcoin wallet anonymous (Nakamoto, 2008). This is because the Bitcoin blockchain records the wallet address, which is a hash of a user’s public key, instead of the actual public key. However, if one is able to associate a wallet address with a real-life identity, doing so may also reveal information about the identities of people who have received transactions from that address. The privacy and pseudo-anonymity of Bitcoin is, thus, eroded by associations between addresses.

There has been research exploring ways to parse the blockchain in an attempt to cluster together addresses and associate those clusters with real-life identities (Meiklejohn, 2013). That research is not relevant anymore due to 1. increased use of hierarchical-deterministic wallets; 2. increased inclusions of multiple input and output addresses in an individual transaction; and 3. increased use of new addresses for receiving change from transactions.

Mixing Services

To further obfuscate associations between addresses, some Bitcoin users opt to employ a mixing service. In general, mixing services act as “black boxes” which accept bitcoin from the user and eventually send it to a forwarding wallet address that is also provided by the user. Along the way, the service attempts to obfuscate the chain of transactions between the user’s input and forwarding addresses (bitcoinwiki, 2014).

Mixing services generally don’t publish their mixing algorithms and also vary in their claims of obfuscation. Bitmixer.io is an example of a mixing service that guarantees that there will be no transactional link between a user’s input and forwarding addresses (BitMixer, 2015). Furthermore, the service allows users to specify more than one forwarding address to split their output among, to increase the size of the user’s anonymity set.

Implementation

To address the two use cases under consideration--mixing service validation and address linkability assessment--we developed a suite of analytic tools wrapped in a web UI, intended for individual use on personal computers. Most code was written in Python. We implemented and tested our analytic functions first and then wrapped the web UI around them using Bottle. Our code is free enough of dependencies that all that is needed is an installation of Python 3.

We needed to allow users to run this code locally while keeping technical literacy requirements at a minimum. This was why we chose a web interface. However, the web service is not intended to be hosted on a server that clients connect to over the internet. Rather, to use

the tool, users clone the project, run the server code locally, and point their browser at localhost:8080.

We chose to use web APIs to retrieve blockchain information. Although this can become slow for some of the functions of our tool, the alternative would be to have the client set up and maintain a full node, which is a great barrier to entry. We chose to use blockchain.info's API. But our analytic code is agnostic about the API being used, so other implementations could be built on top of another web API or a local database. All API responses are cached in a folder.

The range of values an output needs to fall in to be considered part of the anonymity set was calculated by subtracting the percent and flat fee, usually around 0.5-3.5% and 50000 Satoshis, respectively. In our trial on bitmixer.io, we found that the mixing service did *not* subtract the transaction fee: we received 1464113 Satoshis, which is our original amount, 1522200, with the fees (1522200 * 0.005313 + 50000) deducted.

To quantify the connectedness of two addresses, we developed 5 metrics. Each metric compares some characteristic of the addresses and returns output in a quantifiable manner. Each linkability test returns a normalized value between 0 and 1.0 to aid in general comparison of results. This value is determined by the formula:

$$1 - \frac{|addr1characteristic - addr2characteristic|}{max(addr1characteristic, addr2characteristic)}$$

except in the case of lifespan overlap, which uses:

$$1 - \frac{|addr1lifespan - addr2lifespan|}{addr1lifespan, + addr2lifespan}$$

addr1characteristic and *addr2characteristic* were some measure for the specific test being run. Average Transaction Frequency calculated total number of transactions divided by the number of blocks the addresses had been alive (difference in block height between the first and last transactions on record) for both address 1 and 2.

Total Bitcoin Sent and Received

This metric looked at the total sent value of address 1 and the total received value of address 2, accessed just as fields of the address object returned by the blockchain.info API.

Average Amount Sent and Received

This metric calculated the average value sent for address 1 as total sent divided by the number of transactions and average value received for address 2 as total received divided by the number of transactions.

Average Number of Inputs and Outputs

This metric calculated the average number of outputs in address 1's transactions (sum of number of outputs for all transactions divided by the total number of transactions) and the average number of inputs in address 2's transactions (sum of number of inputs for all transactions divided by the total number of transactions).

Lifespan Overlap

This metric calculated the lifespan of each address as the difference in block height between the first and last transactions on record, and compared it to the number of overlapping blocks that they had been alive. The overlap was calculated as the difference between the smaller block height of the final transactions of each address (i.e min(block height of last transaction in address 1, block height of last transaction in address 2) and the larger block height of the first transactions of each address (i.e. max(block height of first transaction in address 1, block height of first transaction in address 2)).

Results

Our final product is an application with a web interface with 3 main functions. The first two, Direct Link and Anonymity Set, relate specifically to the mixer verification service, determining whether there is a path from the input transaction to the output transaction (i.e. whether the mixer actually gave back mixed coins) and finding all of the addresses in the output transaction's anonymity set (essentially how good of a job the mixer actually did) respectively. The final section, Linkability, addresses the more general case of correlating two addresses based on the metrics discussed previously, returning values between 0 and 1.0 for each metric. While individually each result may be subject to noise, when aggregated they should provide a general picture of how linked the two addresses are.

Direct Link Example Results

Provided these values,

> Entering Transaction Hash:
> 490898199a566dcb32a4a9cf45cc7d3cb5f1372e1703c90ad7845acf400f17a5
> Exiting Transaction Hash:
> cb9e8ec8ad02d0edd7b7d9abb85b2312304ffda263493e5ee96e83bc2e78ce17
> User's Sending Address: 1B1tDpsuUBKu25Ktqp8ohziw7qN43FjEQm
> User's Receiving Address: 1MV8oVUWVSLTbWDh8p2hof6J7hfnEm4UXM
> Mixer's Receiving Address: 1Luke788hdrUcMqdb2sUdtuzcYqozXgh4L

BitSniffer finds the link between them:

> Found link
>
> 1MV8oVUWVSLTbWDh8p2hof6J7hfnEm4UXM →
> 1Luke788hdrUcMqdb2sUdtuzcYqozXgh4L →
> 1B1tDpsuUBKu25Ktqp8ohziw7qN43FjEQm

Anonymity Set Example Results

Provided these values,

> Entering Transaction Hash:
> 664c6c87f005fa8b7314eb5d412e39f0695b17b94fe2882315a3ff0a71f980de
> Coin Value: 1522000
> Start Time: 0.0
> End Time: 0.3
> Flat Fee: 50000
> Percent Fee, Lower Bound: .005 (.5%)
> Percent Fee, Upper Bound: .034904 (3.4904%)

BitSniffer finds these addresses in the anonymity set:

> There were 3 results.
>
> - Address 1GPDTsmyVx6MSZ5ehq69yXYxWV2Pcm36Ta in transaction d6a3202f6927222c052561628d5332160b66ebb28469cb0e4823a72ffc684680, 1 blocks after the first.
> - Address 1PnNVZBdTqTLYq6v9tkhtqNpvLguR3HrW2 in transaction aba44a0cc099824379fa1088fbca0a353a033ce71087abdb16e861da3e0a2387, 2 blocks after the first.
> - Address 1B3djUrATFUkjDD29NaWg7PKLuPoj2FzBi in transaction 0df780d70b93bed4094f9b88c4d6e76eb26bf20da4b3958ccdf3a9b9043d95ca, 2 blocks after the first.

Linkability Example Results

Given these two addresses,

> address 1: 1MuFGDztkvU7xEUSamPR3zFPDhPjvQ5ZSj
> address 2: 1Luke788hdrUcMqdb2sUdtuzcYqozXgh4L

BitSniffer's metrics yield these values:

> Transaction Frequency: 0.060662
> Total Bitcoin Sent and Received: 0.6
> Average Amount Sent and Received: 0.8
> Average Number of Inputs and Outputs: 0.75
> Lifespan Overlap: 0.064516

These addresses belonged to members of the BitSniffer team and had been used to send coins to the other. Since both addresses had a direct transactional link and sent and received similar amounts, 3 of the 5 metrics yielded values close to 1.0.

> Compare these results with those of another set of addresses:
>
> address 1: 18heVLNxGLAQ1MG2wxD4UytfvFXmyxWhWs
> address 2: 1FEFqzSuK8S6gdmDea6yzmxq2BRJ1mbvz4
>
> Transaction Frequency: 1.0
> Total Bitcoin Sent and Received: 0.000248
> Average Amount Sent and Received: 0.000248
> Average Number of Inputs and Outputs: 1.0
> Lifespan Overlap: 0.000212

These randomly selected addresses did not contain a direct transactional link and also contained significantly different amounts of Bitcoin in their respective wallets. The only similarity between them was that, on average, both sent or received transactions at the same interval and had the same number of input and output addresses in the transactions.

Future Work

There are three main areas where we feel someone could expand upon our work: metrics, machine learning, and blockchain analysis.

Metrics

The strength of our linkability analysis comes from combining the results from multiple metrics. It follows that adding additional metrics to calculate for an address pair would likely result in a more conclusive determination of the address pair’s connectedness.

Machine Learning

The set of values from an address pair's linkability metrics form a feature vector for that address pair's connectedness. We predict that one could use these feature vectors to train a classifier, such as a neural network or support vector machine, for distinguishing between "dirty" and "clean" address pairs (a "dirty" address pair contains at least one address that's been used for illegal activity; in a "clean" address pair, neither address has been used for illegal activity).

In theory, with enough training examples of dirty and clean pairs, we predict that the classifier would have an acceptable level of performance. However, in this case, we would strongly suggest adding additional metrics to the feature vector.

Blockchain Analysis

We wanted to extend the Bitcoin mixer path-discovery algorithm to work with two arbitrary addresses, however, doing so was outside of the scope of this project. To search for a possible path of transactions between two addresses, one would need to run a full Bitcoin node to have access to the entire Blockchain; accessing the Blockchain through a web API would be too slow.

Further, in order to have acceptable running times for the search, we predict that one would need to store the Blockchain in an efficient data-structure, such as an SQL table, and/or store the Blockchain on a solid-state drive or RAM-disk.

References

bitcoinwiki. (2014). Mixing Services. *en.bitcoin.it*, https://en.bitcoin.it/wiki/Category:Mixing_Services

BitMixer. (2015). How does it work? *BitMixer.io.* https://bitmixer.io/how.html

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. *bitcoin.org* https://bitcoin.org/bitcoin.pdf.

Meiklejohn, S., Marjori, P., Jordan, G., et al. (2013). A Fistful of Bitcoins: Characterizing Payments among Men with No Names*. Proceedings of the 2013 conference on Internet measurement*. http://conferences2.sigcomm.org/imc/2013/papers/imc182-meiklejohnA.pdf