Analyzing the Bitcoin Transaction Graph: A Look at Mixers and Traceability

6.858 Fall 2013

Jeff Chan (jeffchan@mit.edu)
Tanya Liu (tanyaliu@mit.edu)
Eddie Xue (exue@mit.edu)

1. Introduction

Bitcoin is a digital crypto-currency which is not controlled by a central issuing authority but rather a network of nodes maintaining the transaction record, the blockchain. Although a user need not connect a real identity to their Bitcoins, which are stored on addresses, all transactions are contained in the public blockchain to prevent double-spending. We aim to use this record to find potential anonymity leaks or metadata which may de-anonymize or otherwise connect transactions a user would not want connected. Specifically, we examine the *traceability* and *detectability* of attempts to cover the traces of the origin of Bitcoins using **mixing services**, which attempt to mask the source of Bitcoins by adding intermediate transactions and combining inputs with many users before paying out. We analyze two mixing services in depth, Blockchain.info's shared wallet and Bitcoin Fog.

We initially aimed to analyze the entire blockchain (over 13GB of data) and create an association graph, linking related entities and identifying statistical patterns. However, we realized that quite an in-depth analysis had been performed [1] [2] [3]. In addition, downloading and managing this huge amount of data represented a large overhead for us. We switched to analyzing anonymity relating to mixing services instead, as it has been less thoroughly researched and presents another interesting challenge.

2. Background

Bitcoin mixing services claim that by depositing and withdrawing from their service, you will receive Bitcoins from an address than cannot be reliably tied to your input address. One reason for using this is that purchasing Bitcoins on an exchange requires personally identifying information to a third party, such as a linked bank account, debit card, or national ID, to make payments from fiat currency and verify identity to reduce fraud risk experienced by the exchanges. An attacker can then trace its source to an origin and potentially attack the exchange (there have been many high-profile breaches of various wallets and exchanges, with Bitcoin stolen and also user details leaked).

A user of a Bitcoin mixing service may also seek to hide the fact that they are using a mixing service in general or a particular mixing service. Decreasing detectability can be important to avoid an attack being mounted in the first place.

Because mixers are only effective with a large user base, it is likely that there will only be a few large, practical mixers to choose from, and creating one's own service is not a great alternative to trusting a third-party.

3. Threat Model

For the purpose of discussion, we establish two threat models. The first one is **traceability**. Given a deposit input address and an withdrawal output address, an adversary must not be able to directly establish a link between the two addresses through the Bitcoin public blockchain, whether through the graph or by other metadata or statistical means. The second one is **detectability**. Given an withdrawal output address, an adversary must not be able to say with full certainty that the withdrawal was made through a mixing service.

4. Analysis

4.1 Bitcoin Fog

We deposited twice to Bitcoin Fog and, from these deposits, withdrew a total of two times for each deposit, each time to two separate addresses, for a total of two input and 8 output transactions. We analyzed the connections between the withdrawal addresses and discovered the method of operation for Bitcoin Fog.

Bitcoin Fog operates on a server that is accessed through a Tor hidden service, although they offer an HTTPS gateway as well for increased convenience at the price of security - one could tell a certain IP address was using Bitcoin Fog, but not who they were. As of writing the report, 1:15AM EST on 13 Dec 2013, the web gateway was down, but the site is still accessible through Tor at the .onion address. This might be used if you think people would track you using Bitcoin Fog. However, anyone who receives coins from Bitcoin Fog is likely to know it has come from a mixing service, or Bitcoin Fog in particular, as shown by our analysis of withdrawals.

Graph Analysis. We found lists of transactions in Bitcoin Fog by tracing our withdrawal transaction to a high-value “slush fund,” from which many withdrawals are directly made, until its value reaches 0 and it is re-funded from deposits (usually ~3 months old) to the order of several hundred bitcoin.

One weakness may be Bitcoin Fog’s simplicity of its internal, very straight transaction graph with having one large money pot and withdrawing from that, making change with the other output address, which makes tracing relatively easier than a multi-tiered obfuscation scheme. We definitely know what it’s doing and can tell it’s just mixing.

Bitcoin Fog separates withdrawals and deposits well in the time domain, typically by at least 3 months, by analyzing older known withdrawals from Bitcoinfog and their corresponding deposits. The deposits are not spent until Bitcoinfog requires it to reconstruct a large fund for more withdrawals, consisting of hundreds of transactions

funneling into a fund address. Thus Bitcoin Fog's **traceability** is very strong.

We find that Bitcoin Fog is highly **detectable** in that its pattern of withdrawal transactions with a very clear unique graph pattern, and it does not simply look like a complex web of transactions. Similarly, after the deposits sit untouched, they are rolled into a large fund from which Bitcoin Fog then begins paying out from.

Transaction volume

Usage: Bitcoin Fog appeared to process about 880 BTC in the time period from 2013-11-26 23:26:59 to 2013-11-30 09:42:41 (UTC), a rate of about 190BTC/day. All of our withdrawals within a few days came from the same large source, however it is possible that Bitcoin Fog is also operating other simultaneous funds that were not discovered. This is of interest to users wishing to mix large amounts of bitcoin, as making large transactions stands out as atypical, and mixers rely on having lots of other users withdraw bitcoin in between any one given user.

Transaction size - Benford's law

Bitcoin Fog first helps obfuscate transaction sizes by charging a random fee between 1-3%. It also requires automatic randomization of withdrawals both over time (minimum 6 hours) and size (minimum 2 transactions).

We also analyzed the randomness of the digits in Bitcoin Fog algorithms to see how well they would match with a 'natural' distribution taken from Benford's law, and if it would reveal any peculiarities in the mechanism used to generate these withdrawals.

Benford's law describes an empirical observation about the frequency of digits in real-world sources, for instance that 1 as a first digit tends to be much more common than 9, especially as numbers come from natural sources such as house numbers, receipt totals, stock volumes, etc. It is often used to detect fraud when data is generated by hand or through a mechanical process. We analyzed the sizes of both transaction withdrawals (which Bitcoinfog creates random sizes for, since all withdrawals must be split across at least 2 transactions), and deposit amounts. A regular random number generator would generally yield a uniform distribution of digits.

First digit of the transaction size (0.024 = 2, 74.22 = 7, 0.1 = 1, etc.)

Digit	Expected Benford's	Bitcoin Fog withdrawal	Bitcoin Fog deposit transaction

	law for first digit	sizes (460 transactions)	sizes (605 transactions)
1	.301	.335	.312
2	.176	.170	.053
3	.125	.143	.129
4	.097	.093	.129
5	.079	.070	.093
6	.067	.046	.056
7	.058	.054	.060
8	.051	.05	.101
9	.046	.039	.068

Second digit of the transaction size (0.024 = 4, 74.22 = 4, 0.1 = 0, etc.)

Digit	Expected Benford's law for second digit	Bitcoin Fog withdrawal sizes (460 transactions)	Bitcoin Fog deposit transaction sizes (605 transaction sizes, inputs and outputs)	Bitcoin Fog transaction deposit sizes (inputs only)
0	.1197	.1196	.2512	.2661
1	.1139	.1087	.0826	.0939
2	.1088	.1109	.0860	.0900
3	.1043	.1000	.0678	.0705
4	.1003	.0913	.1091	.0861
5	.0967	.1109	.0926	.0861
6	.0934	.1021	.0694	.0685
7	.0904	.0870	.0727	.0724
8	.0876	.0848	.0579	.0587
9	.0850	.0848	.1114	.1076

Humans tend to deposit round numbers of coins.

Blockchain.info

We used the Blockchain.info shared wallet service, which charges 0.05% for all transactions and a 0.0001 BTC fee for small transactions. We made 3 separate deposits, of 0.20, 0.20, and 0.05 BTC, and 3 withdrawals, of 0.19991, 0.05009, and 0.05 BTC. The second transaction drew on some Bitcoin that was leftover in the wallet from the first time, and allowed us to trace the behavior in that situation.

One critical finding is that temporal separation is poor - although the Blockchain shared wallet purports to always ensure at least 250 degrees of separation, and provides a "taint analysis" tool for users to find associated addresses (the source of Bitcoin or

the sinks/recipients of Bitcoin for a given addresses), unlike the 3-4 months of time separation offered in Bitcoin Fog, there is actually 0 time separation between withdrawal and deposit, at least when analyzed on Blockchain.info. When the user deposits, a new address is not generated by default - it must be selected manually. This is not a huge deal, however, when the user makes a withdrawal, an equal amount of Bitcoins is immediately transferred out of the wallet deposit address (marked as spent), at the exact same timestamp. By downloading the blockchain and looking at all transactions within the appropriate time range (which can be very small, since the timestamp seen on Blockchain.info is equal to the second), an attack can be made on this channel to find the originating address.

If the user makes several withdrawals to the same address, or even easier one withdrawal containing all the input coins, all that needs to be done is to find the original deposit amount is to total the withdrawals and find another spend transaction from the deposit address to elsewhere, at the same time (divide by 0.995 due to the 0.5% fee and add any transaction fees such as 0.0001 BTC for small transactions). This attack works best when the volume of Bitcoin transaction is relatively low, today it is currently ~40,000-80,000 bitcoin per day, or 28 per minute if uniformly distributed, however due to the precision of the timestamps this attack should still work well. This is an implementation weakness that could likely be fixed by adding a randomized delay job queue, or tracking deposits in a manner similar to blockchain.info and only spending when needed.

Future Work

Create a graph analysis and statistical analysis for blockchain.info by running the extend_graph analysis on one node (the withdrawal). Its structure is less predictable than that of Bitcoinfog so we did not yet analyze that. Similarly, once transactions associated with Blockchain.info are known, run statistical analyses on the size of transactions, digits, and amount of bitcoins moving through shared wallets, for detectability and traceability.

Attempt the attack on blockchain.info by (1) finding known withdrawals from blockchain.info through looking at forums, and (2) without knowing anything about blockchain.info withdrawals, looking for all pairs of transactions that occurred at the same time which match the criteria described above for a shared wallet withdrawal. If such an attack is successful it would invalidate the security model of the blockchain.info shared wallet. This would require downloading the entire blockchain, and analyzing all transactions in the most recent days or given timespan.

Conclusion

Detecting whether someone is using a Bitcoin mixing service is not difficult, especially if they are using one of the most popular services, whose methods of operation can be analyzed since they have relatively clear patterns. In order to mix effectively, services need Alternatives include looking at graphs and deducing that the convoluted-looking structures and deliberate circular transactions

References

[1] http://eprint.iacr.org/2012/584.pdf
[2] http://www.mdpi.com/1999-5903/5/2/237
[3] http://cseweb.ucsd.edu/~smeiklejohn/files/imc13.pdf