The Pennsylvania State University
The Graduate School

AN ANALYSIS OF ANONYMITY IN BITCOIN USING P2P NETWORK TRAFFIC

A Thesis in
Computer Science and Engineering
by
Diana Koshy

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2013

The thesis of Diana Koshy was reviewed and approved* by the following:

Patrick McDaniel
Professor of Computer Science and Engineering
Thesis Advisor

Sencun Zhu
Associate Professor of Computer Science and Engineering

Lee Coraor
Associate Professor of Computer Science and Engineering
Director of Academic Affairs

*Signatures are on file in the Graduate School.

Abstract

Over the last 4 years, Bitcoin, a decentralized P2P crypto-currency, has gained widespread attention. The ability to create pseudo-anonymous financial transactions using bitcoins has made the currency attractive to users who value their privacy. Although previous work has analyzed the degree of anonymity Bitcoin offers using clustering and flow analysis, none have demonstrated the ability to map Bitcoin addresses directly to IP data. We propose a novel approach to creating and evaluating such mappings solely using real-time transaction traffic collected over 5 months. We developed heuristics for identifying ownership relationships between Bitcoin addresses and IP addresses. We discuss the circumstances under which these relationships become apparent and demonstrate how nearly 1,000 Bitcoin addresses can be mapped to their likely owner IPs by leveraging anomalous relaying behavior.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to thank my advisor, Dr. Patrick McDaniel, for providing endless encouragement throughout this whole process, even before I became his student. Thank you for steering me in this direction and helping me see it through. I would also like to thank Dr. Yanxi Liu and Dr. Robert Collins for giving me the opportunity to explore the fascinating world of computer vision. Thank you for making seemingly incomprehensible concepts clear and beautiful; I truly enjoyed the work we did together. Additionally, a huge thank you to Dr. Sencun Zhu for taking time out of his schedule and agreeing to be on my thesis committee on such short-notice. It is greatly appreciated.

I want to acknowledge all my fellow graduate students at Penn State for their hard work, long hours, and dedication. To those who have graduated and those that still remain, you guys are awesome! Your enthusiasm, intellect, and drive made this process worthwhile and memorable. Thank you Kyle Brocklehurst, Ingmar Rauschert, Asad Butt, Jingchen Liu, Anand Raja, and the rest of LPAC for your help in my first year of graduate school and for your continued friendship. Additional thanks to Steve McLaughlin, Devin Pohly, Josh Schiffman, and the gang at the SIIS lab for the support you've provided to my husband and me throughout this process.

Having completed my degree while working full-time, I would like to also thank my coworkers and especially my managers, Alex Casanova and Eric Lin, for being so supportive and understanding. I promise to have better timing with my days off in the coming future!

Most of all, I would like to thank my family for putting up with my absence and what must have felt like constant complaints. To my mom and dad, thank you for encouraging me every step of the way. To my sister, thank you for believing in me. To my best friend, Natalie, thank you for always listening and being understanding. Finally, to my wonderful husband, Philip, thank you for enduring my bad moods, lack of attention, and "obtuse" writing style. Without you, I would never have been able to keep going.

Chapter 1

Introduction

Bitcoin is a peer-to-peer crypto-currency first proposed and implemented by Satoshi Nakamoto in 2009 [1]. It is unique among e-currencies due to its decentralized nature; instead of a single entity, numerous peers called "miners" are responsible for vetting transactions that are relayed using a gossip protocol [2]. Bitcoin has gained wide popularity, with hundreds of services[1] emerging that either accept bitcoins directly or through third-parties such as BitPay. Users could also trade their bitcoins for nearly any fiat currency via Bitcoin exchanges. Since its inception, the exchange rate has fluctuated wildly, reaching over 550 USD per bitcoin at the time of this writing (November 2013).

Part of what attracts people to Bitcoin is that it allows for pseudo-anonymous financial transactions; instead of disclosing personal information, users create any number of Bitcoin identities/addresses, in the form of cryptographic keys, which are used to accept and send bitcoins. We have seen the perceived anonymity provided by Bitcoin leveraged when Wikileaks was able to receive over 1,000 "anonymous" Bitcoin donations totaling over 32,000 USD; other financial institutions, such as Paypal, prevented supporters from making donations using fiat currencies due to government pressure [4]. We have also seen the birth and recent death of the Silk Road, a Bitcoin marketplace once called "the Amazon.com of illegal drugs" [5, 6].

Previous studies (discussed in Chapter 3) showed that it may be possible to cluster Bitcoin identities into distinct entities, track the flow of their bitcoins, and in some instances deanonymize them using external information like forum posts where people divulged their Bitcoin identities intentionally. However, many developments in both services available to the Bitcoin community (see Chapter 5), as well as Bitcoin software [7], have weakened, if not invalidated, the assumptions necessary for such approaches. To our knowledge, there has been no work that has attempted to relate Bitcoin addresses to specific IPs. The ability to create such mappings is important since there have been cases where individuals participating in P2P networks have been identified by law enforcement after their ISPs had been subpoenaed [8]. In this work, we set out to determine

[1] Links to a wide range of tools and services can be found at [3].

if real-time transaction traffic received from directly connected peers can alone be used to create Bitcoin address-to-IP mappings. This approach was inspired by a technique proposed by Dan Kaminsky during the 2011 Black Hat conference [9].

By analyzing 5 months of data we collected using our custom-built Bitcoin client, we were able to classify distinct transaction relay patterns and design heuristics for hypothesizing transaction ownership. We then demonstrated how Bitcoin address-to-IP mappings can be derived and evaluated using aggregate statistics from our transaction data. We found that even after applying conservative thresholds, several hundred high-confidence ($> 90\%$) ownership pairings could still be discovered in our data. Over 1,000 remained if we allowed thresholds to drop to 50%. We note, however, that the majority of these were obtained from anomalously relayed transactions, and that normal transaction traffic overall proved to be very difficult to deanonymize.

The rest of this thesis is organized as follows. Chapter 2 gives a background of the Bitcoin protocol, while Chapter 3 provides an overview of related work. In Chapter 4, we describe CoinSeer, our custom-built Bitcoin client, and discuss how to create, evaluate, and prune Bitcoin address-to-IP mappings. Chapter 5 discusses our results, caveats and limitations of our method, and steps Bitcoin users can take to safeguard their anonymity. Finally, we offer concluding remarks in Chapter 6.

Chapter 2

Background

This chapter is dedicated to describing the nuances of the Bitcoin protocol that are relevant to our work, as well as providing some general background information for completeness. Since transactions are integral to our analysis, we discuss their internals, provide examples of simple usage, and discuss how they are relayed through the network.

2.1 General Overview

Bitcoin is unique due to its decentralized nature, which allows it to be independent of any controlling entity. This requires certain participants called miners to make sure that financial transactions are legitimate. In order to prevent people from (a) using money which does not belong to them, or (b) reusing money which they have already spent (this is called double-spending), the entire history of transactions must be publicly available; this is to avoid a single point of centralization. The historical transaction ledger is called the block chain and can be accessed and scrutinized by anyone. Nothing is encrypted. To protect users' identities, IP information is never stored, and cryptographic keys are used instead of personal information. Bitcoins are sent to and from users' public keys, which are often referred to as Bitcoin addresses[1]. In this way, despite all transactions being public, the parties involved remain pseudo-anonymous.

2.2 Anatomy of a Transaction

Bitcoins change hands via transactions. A transaction is a data structure that contains inputs and outputs. The sender of a transaction uses the inputs to claim coins he received in older transactions; he lists the recipient(s) of these coins within the transaction's outputs.

[1] Omitting certain details, a Bitcoin address can be obtained from a public key through a series of hashes whose digest is converted to a lightly modified version of base 58 [10]. Thus, the terms Bitcoin address and public key are often used interchangeably.

For example, if Alice wants to receive 50 bitcoins (BTC) from Bob, she creates an asymmetric key-pair and gives him her public key, A^+. Bob creates a transaction and encodes Alice's public key as the recipient of his coins within one of the transaction's outputs (Figure 2.1, Transaction 1). The next day, Alice wants to send 20 BTC to Charlie. She creates a new transaction and claims the money she received from Bob by referencing it in one of the transaction's inputs (Figure 2.1, Transaction 2). An important caveat of the Bitcoin protocol is that the amount of bitcoins claimed in an input cannot be specified. In order for Alice to only send 20 BTC to Charlie, she has to create an extra output to send 30 BTC in change back to herself (Transaction 2, Output 1). She can then reference this change in later transactions. After specifying all her outputs, Alice signs the new transaction with her private key (A^-) and includes this signature within the corresponding input. In this way, ownership of the referenced coins can later be verified and the transaction's integrity is protected.

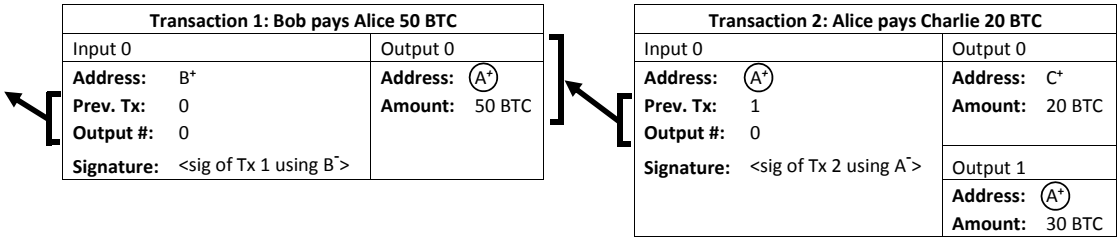

Figure 2.1: This figure demonstrates how Alice, who owns Bitcoin address A, would create a new transaction (Transaction 2) which spends bitcoins received earlier (Transaction 1). Note that the Bitcoin address of the input must match the Bitcoin address of the referenced output. Note also that the sender of the transaction must sign it with her private key (denoted in this diagram with the superscript -). We caution that this is a simplified representation of the internals of a transaction.

In general, users are encouraged to have many Bitcoin addresses. Thus, Alice could have sent all or part of her change to a different address she owns. She could also have left it unclaimed, thus awarding it to a miner when the transaction eventually got into the block chain. Additionally, if she needed to spend more than 50 BTC, she could have created additional inputs, each of which would reference older transactions. This is called a multi-input transaction.

2.3 P2P Relaying

Bitcoin uses a gossip protocol [2] to relay messages across the network. When a user creates a transaction, he sends it to his directly connected peers. These peers assess whether the transaction is valid (discussed below). If it is, they relay it to their peers and the transaction gets propagated through the rest of the network. If it is not valid, it is simply ignored.

A transaction received from a peer must pass a series of checks before being further relayed. Besides basic sanity checks to make sure the transaction format conforms to the protocol, Table 2.1 shows common reasons a peer may ignore a transaction.

Type	Description
Repeated	The transaction has already been relayed recently.
Old	The transaction is already in the main block chain.
Double-Spend	The transaction attempts to claim an output already claimed by a previous transaction.
Bad Signature	The input signature(s) cannot be verified (e.g. attempting to spend someone else's coins).
Orphan	One or more of the claimed outputs cannot be found.

Table 2.1: Types of Ignored Transactions

2.4 Blocks and Mining

Eventually, the valid transactions that are being broadcast on the network reach peers called Bitcoin miners, who group the transactions into blocks, solve a proof-of-work puzzle, and release their work onto the network for others to "vote" on. Blocks form a chain, which can sometimes fork if two blocks are created simultaneously. Each miner casts a vote for a given chain by attaching his block to the end of it. The longest chain (or, more specifically, the chain that required the most amount of total work to produce) is said to be the accepted transaction history for the network.

If a miner's block becomes part of the main block chain, he currently receives a reward of 25 BTC (though this will decrease in a pre-specified way over time). The coins are awarded in a special type of transaction called a Coinbase transaction which always appears as the first transaction of any given block. It is the only type of transaction where the inputs are not claiming old outputs and is the method by which bitcoins are generated. To control the rate of coin generation, the proof-of-work required to create a block dynamically changes in difficulty so that only one block is created every 10 minutes [11].

We include information about blocks and mining for completeness as they are not integral to our analysis. The block chain is available to the public and can be browsed on a number of websites [12, 13]. It can also be obtained by joining the Bitcon network and asking connected peers for blocks. More detailed information can be found on the Bitcoin Wiki [14].

Chapter 3

Related Work

Bitcoin anonymity has been the topic of numerous forum discussions, community efforts, and research papers. In this chapter, we present relevant discussions, Bitcoin client updates, and academic literature, as well as discuss how our work is unique.

3.1 Forums and the Community

The Bitcoin community has discussed and addressed the topic of anonymity since Bitcoin's inception. When Satoshi Nakamoto first introduced the protocol, he discussed a possible way to exploit multi-input transactions [1]. Under the assumption that only one user can create a transaction, and that the creator of a transaction must own all input Bitcoin addresses, all unique input addresses in a multi-input transaction can be clustered into a single entity. This idea gave birth to a number of papers that relied heavily on such input clustering when exploring anonymity in Bitcoin (discussed in Section 3.2). A 2010 forum post by a user called "theymos" argued for encrypting Bitcoin traffic and outlined how bitcoins of multiple users can be mixed in order to obfuscate which users coins flow between across time [15]. Meanwhile, the Bitcoin Wiki [14], an in-depth repository of information relevant to the currency, has an article dedicated to anonymity in which the authors describe scenarios for how a user's identity can become compromised [16].

Services have emerged that give users some protection from deanonymization. eWallets allow users to create accounts which they can use to receive and send money without ever downloading Bitcoin software themselves; in this way, all of a user's identities are controlled by the eWallet service and deanonymization could only occur if the service is compromised. Mixing services allow users to send their coins to one set of addresses and receive them back from a set of unrelated addresses, thus breaking any links between sent and received coins. A detailed listing of these services and more can be found at [3]. Blockchain.info [13], one of the most intricate Bitcoin tracking and service sites we have seen, provides a number of utilities and services to the Bitcoin community; these include an eWallet, Bitcoin address taint analysis, and even a way for

people to send transactions they created in raw hex format out onto the network without using their own Bitcoin software [17].

Bitcoin software has also become more user-friendly for those who take anonymity seriously. In version 0.7.0, the popular Bitcoin-Qt client exposed a "raw transaction API" with new functionality that gives users greater control over how their transactions are created and relayed [7]. For instance, by default, the client software creates multi-input transactions by selecting at random from a pool of unclaimed coins on any of a user's Bitcoin addresses. To mitigate the risk of linking potentially compromised Bitcoin addresses with ones that are still anonymous, people using the new API now have the ability to chose which addresses to include together in a given transaction. Furthermore, it has become much easier for multiple users to create a single transaction using this API; this deals a major blow to input clustering, which assumes all input addresses in a transaction are controlled by the same entity. Although these features were always available to advanced users familiar with the Bitcoin protocol at the binary level, the API has made the process easier.

3.2 Academic Literature

Several academic papers have analyzed the extent of anonymity in Bitcoin. The majority of them cluster Bitcoin addresses into distinct entities, analyze the flow of bitcoins among these entities, and in some instances tie entities to identifying information through external means. To our knowledge, no one has attempted to deanonymize Bitcoin addresses at the IP level, and no other papers discuss using actual relay traffic.

The original Bitcoin paper [1] cautioned that although users could hide their identities behind Bitcoin addresses, the public nature of the transaction ledger could allow addresses to be linked together. Multi-input transactions, which at the time could only be created by one user, were cited as a potential means to clustering multiple Bitcoin addresses into one entity. Reid and Harrigan [18] downloaded the public transaction ledger (i.e. block chain) and used this method to cluster Bitcoin addresses into "users". They created two networks, modeling the flow of bitcoins among transactions and users, and analyzed their topologies. The authors showed how these graphs, along with external information from forum posts, can be used to track a particular target (in this case, a thief). Ron and Shamir [19] mirrored Reid and Harrigan's two-graph solution when analyzing the typical behavior of entities on the Bitcoin network, including how these entities acquire and spend bitcoins and how they move their funds around to protect their privacy. Androulaki et al. [20] again took a similar approach, using data from a simulation they created which was meant to mimic the use of bitcoins within a university setting. In addition to input clustering, the authors used K-means and Hierarchical Agglomerate Clustering to tie together behavioral patterns. They also clustered inputs with outputs based on the "safe" assumption that if a 2-output transaction's outputs contain one old address and one new address, the new address must have been used for change. This assumption is faulty because it is equally likely that the old address might be an address the user previously used for other transactions and

is now using for change, while the new address may be one the payee created for receiving coins. Although they had impressive results - 35-40% of users were deanonymized - their simulation was restricted by the assumptions the authors made. Meiklejohn et al. [21] also used input and output clustering to create a set of "users." They actively interacted with parties on the Bitcoin network to create a list of known Bitcoin addresses for each party, using this information to assign identities to their clusters. Finally, they used flow analysis to study interactions among users.

Other papers did not try to deanonymize Bitcoin users, but instead gave wholistic analyses of anonymity and proposed some solutions. Ober et al. [22], using the available transaction history, analyzed what increases and decreases anonymity in Bitcoin, concluding that clustering is the most important challenge the community faces. Miers et al. [23], arguing that Bitcoin is not truly anonymous, proposed an extension to the protocol that uses cryptography to make transactions fully anonymous. Barber et al. [24] discussed the various vulnerabilities inherent to Bitcoin, finally proposing and outlining a trust-free mixing service. Moore and Christin [25] cautioned that mixing services, exchanges, and other centralized intermediaries can pose a major risk to Bitcoin investors since they can either have a security breach or close and disappear with people's bitcoins.

3.3 How We Are Different

Unlike the papers above, our work focused on analyzing real-time transaction traffic we received from directly connected peers. We did not rely on external sources for additional information, instead evaluating how network traffic alone could be used to obtain identifying information. We also did not rely on flow analysis or clustering, since many of the assumptions for these techniques have become greatly weakened. Our method was inspired by a technique proposed by Dan Kaminsky during the 2011 Black Hat conference [9]. To our knowledge, there is no literature that uses network traffic to associate Bitcoin addresses with IPs.

Chapter 4

Methodology

The goal of this thesis was to determine whether transaction relay information can be used to pair Bitcoin addresses with IP addresses. We built a custom Bitcoin client that saved all messages being relayed on the Bitcoin network and meta-information about each message. After collecting our data, we manually analyzed it for patterns and then attempted to generalize the patterns we observed in order to come up with a more algorithmic approach for mapping Bitcoin addresses to the IPs that own them. In this chapter, we present the tool we built and outline the steps we took to arrive at our final results.

4.1 CoinSeer: The Need For A Custom Bitcoin Client

Inspired by Dan Kaminsky's 2011 Black Hat presentation [9], we decided to analyze traffic patterns on the Bitcoin network to see if it was possible to create mappings from Bitcoin addresses to IPs. A preliminary step to creating such mappings is figuring out the creator of each transaction. Since Bitcoin uses a gossip protocol to disseminate information [2], we had to connect to as many peers as possible to increase the likelihood of receiving transactions directly from their creators. We also had to store not only the messages sent by each peer, but message meta-data such as IP and timestamp information.

Although numerous Bitcoin clients exist, none of them are specialized for data collection. Available clients often need to balance receiving and spending bitcoins, vetting and rejecting invalid transactions, maintaining a user's wallet, mining bitcoins, and, perhaps most detrimental to our study, disconnecting from "poorly-behaving" peers; these were precisely the peers we were interested in.

Because existing software had integrated functionality that interfered with our goals, we decided to build our own Bitcoin client called CoinSeer, which was a lean tool designed exclusively for data collection (explained in detail in [26]). For 5 months, between July 24, 2012 and January 2, 2013, CoinSeer created an outbound connection to every listening peer whose IP address was

advertised on the Bitcoin network. We maintained that connection until either the remote peer hung up or timed out. In any given hour, we were connected to a median of 2,678 peers [26]; for the duration of our collection period, we consistently maintained more connections than the only other Bitcoin superclient we know of - blockchain.info.

This data collection effort required storing 60 GB of data per week. Unlike the Bitcoin-Qt client, which drops malformed messages and transactions that cannot be vetted, we kept a record of everything. Our dataset contains information that cannot be obtained retroactively through the public ledger of transactions; we know who was connected to us at what time and have a list of every message a peer relayed during our collection period, whether or not that message was rejected by others.

4.2 Finding Ownership Mappings

After collecting our data and manually looking for patterns, we designed a six-phase method for associating Bitcoin addresses with their owner IPs. Each phase, outlined below, is explained in detail in this section.

Phase 0 Prune transaction data to remove potential sources of noise.

Phase 1 Using relay patterns we have observed for transactions, hypothesize an "owner" IP for each transaction.

Phase 2 Break transactions down into their individual Bitcoin addresses. We do this to create more granular (Bitcoin address, IP) pairings

Phase 3 Compute statistical metrics for our (Bitcoin address, IP) pairings.

Phase 4 Identify pairings that may represent ownership relationships.

Phase 5 Eliminate ownership pairings that fall below our defined thresholds.

4.2.1 Phase 0: Pruning Transaction Data

From the block chain, and during our 5 months of monitoring the Bitcoin network, we accrued hashes for 10,724,442 transactions. 410 transactions were incomplete/invalid, 57,087 were advertised but never sent to us (and thus we lack their input and output information), and 1,573 contained at least one Bitcoin address that we could not parse because the inputs/outputs were non-standard.

Our goal was to attempt to establish "ownership" between IP addresses and Bitcoin addresses, which in turn involved first predicting the "owner" of each transaction using relay information. This restricts us to only using transactions for which we know the senders' IP addresses. Such data is not stored in transactions or blocks, so it cannot be obtained retroactively by downloading the block chain. Our tool allowed us to capture relayer information for every transaction actively being relayed during our collection period - a total of 5,617,202 transactions. This number includes the 57,087 transactions that were advertised but never relayed, as well as 300 of the

transactions containing a Bitcoin address that we could not parse. These were removed from consideration in all subsequent phases.

Additionally, some valid transactions had to be pruned away to minimize noise. We removed 114,100 transactions that exhibited relay patterns which made establishing ownership ambiguous (see Section 4.2.2, and Figure 4.4 in particular). Our biggest source of potential noise were multi-input transactions. In this work, we assume that each transaction has only one owner. A multi-input transaction can be created by one or multiple, unrelated entities with no way to distinguish the difference [7]. Other academic works do not acknowledge this possibility. We argue that not excluding multi-input transactions could lead to incorrect assumptions being made about the ownership of a Bitcoin address. To be conservative, we removed all 1,544,509 multi-input transactions from our dataset, leaving us with 3,901,206 transactions to analyze.

4.2.2 Phase 1: Hypothesizing Transaction Owner IPs

Phase 1 of our approach involved hypothesizing which of each transaction's relayers is its owner. This step acts as a bridge to later mapping the Bitcoin addresses internal to each transaction to owner IPs.

We know that the creator of a single-input transaction owns the input Bitcoin address (since the transaction must be signed by the corresponding private key[1]). Given that Bitcoin uses a gossip protocol and we expect multiple people to relay a single transaction, how can we determine the IP of its creator?

When a peer either creates or receives a valid transaction, he sends advertisements to all of his peers, all of whom can request and repropagate it. Since we were connected to thousands of peers, we received a typical transaction between 1,500 and 2,500 times. During our manual analysis, we discovered certain transactions that exhibited atypical behavior; some were relayed by only a single IP, while others were being continually retrasmitted. Whereas for a typical transaction, we can only hope that the creator was its first relayer[2], anomalies provide additional information that we can leverage when hypothesizing ownership.

Below, we discuss the 3 distinct relaying patterns exhibited by transactions within our collected data and the heuristics we used to hypothesize transaction ownership.

4.2.2.1 Relay Pattern 1: Multi-Relayer, Non-Rerelayed Transactions

The first and most common relay pattern involves a transaction being relayed by multiple people, each of whom relayed the transaction a single time. This is expected behavior according to the protocol and 3,671,341 (approx. 91.4%) of our transactions exhibited this relay pattern.

We present an example in Figure 4.1 to demonstrate ownership assignment for transactions exhibiting this relay pattern.

[1] We note that this does not mean the creator owns the funds associated with that Bitcoin address (see discussion on eWallets in Chapter 5).

[2] We discuss why this assumption is flawed in Chapter 5.

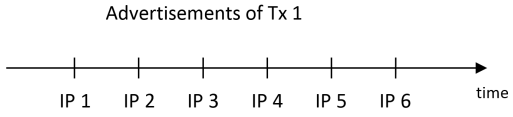

Figure 4.1: In the timeline at right, Tx 1 is being relayed once by each IP. Since this is normal behavior, there is no additional information to exploit. In this case, we simply choose the **first relayer** - IP 1 - as the "owner."

4.2.2.2 Relay Pattern 2: Single-Relayer Transactions

The second relay pattern involves a transaction being relayed by a single person. This includes transactions relayed once, as well as transactions that were relayed multiple times by the same IP. This behavior is highly unusual for a system using a gossip protocol, and only 101,462 (approx. 2.5%) of our transactions exhibited this relay pattern; 74,232 transactions were relayed only once, while 27,230 were relayed at least twice (i.e. rerelayed) by the sender.

This behavior may arise when a peer creates an invalid transaction that its immediate peers reject (see Table 2.1 in Section 2.3 for common rejection reasons). Since we attempt to be a directly connected peer of every Bitcoin node, we are able to record the transaction despite it not being relayed on the network. To demonstrate ownership assignment for transactions exhibiting this relay pattern, we present an example in Figure 4.2.

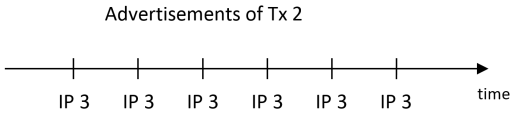

Figure 4.2: The timeline at right shows the advertisements of Tx 2. Since only one IP ever relayed this transaction, there is no ambiguity; we assign the **single relayer** - IP 3 - as the "owner."

4.2.2.3 Relay Pattern 3: Multi-Relayer, Rerelayed Transactions

The third relay pattern involves a transaction being relayed by multiple people and retransmitted by at least one of them. A total of 242,503 (approx. 6.04%) of our transactions exhibited this relay pattern.

The Bitcoin protocol states that a transaction will not be relayed twice by any node except the sender or recipient of coins in that transaction. The sender or recipient will rerelay the transaction each time a new block is created which does not contain it [27]. By rerelaying a transaction, an IP exposes its association with at least one of the keys contained inside. Although this may appear to be a clear way of establishing ownership, we found that many transactions had multiple rerelayers, thus making ownership assignment ambiguous. Besides the transaction's creator, any number of its recipients may also choose to rerelay it. Additionally, all IPs eventually "forget" which transactions they have already relayed, leading to some transactions getting relayed by the whole network in waves.

To remain conservative when hypothesizing ownership, we decided to split the transactions exhibiting this relay pattern into the following two groups:

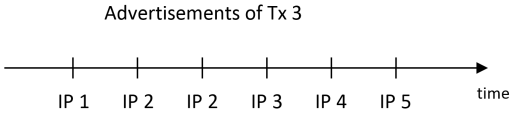

Figure 4.3: For Tx 3, everyone but IP 2 is exhibiting the expected behavior of sending the transaction only once. Since only the sender or recipient of coins in a transaction is supposed to rerelay that transaction, we assign the **single rerelayer** - IP 2 - as the "owner."

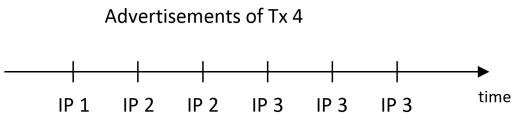

Figure 4.4: This is similar to Tx 3, but there are now multiple rerelayers. This makes ownership assignment more **ambiguous**. Do we assign it to the first rerelayer, or the one with the most relays? To err on the side of caution, we **removed** transactions with more than one rerelayer from consideration.

1. **Relay Pattern 3A**: Multi-Relayer, Single Rerelayer Transactions
   This group contains transactions relayed by multiple people, where only a single person rerelayed the transaction. Approximately 3.2% (128,403) of our transactions exhibited this relay pattern. Figure 4.3 provides an example of ownership assignment for transactions in this group.

2. **Relay Pattern 3B**: Multi-Relayer, Multi-Rerelayer Transactions
   This group contains transactions relayed by multiple people, where at least two people rerelayed the transaction. Approximately 2.8% (114,100) of our transactions exhibited this relay pattern. Figure 4.4 provides an example of why ownership assignment for transactions in this group is ambiguous.

4.2.3 Phase 2: Decomposing Transactions

In Phase 2, we pair the owner IPs assigned to each transaction in Phase 1 with the Bitcoin addresses contained within that transaction. This brings us closer to our goal of associating Bitcoin addresses with IPs and prepares our data for statistical analysis.
We begin by splitting every transaction into a set of triplets which consist of:

1. a Bitcoin address from the transaction
2. the IP which we hypothesized owns the transaction, and
3. the unique transaction number we assigned to this transaction

There is a triplet for each unique Bitcoin address found within a transaction. Because it matters whether a Bitcoin address appears as an input or an output in a transaction, we keep triplets made from input and output Bitcoin addresses separate. Figure 4.5 demonstrates how 3 transactions can be split into corresponding (Bitcoin address, IP, Tx #) triplets.

We note that at the end of Phase 1, our data consisted of 3 groups of transactions, split based on their relaying patterns. For this and subsequent Phases, the data maintains its relaying pattern split since eventual Bitcoin address-to-IP mappings obtained from anomalously relayed

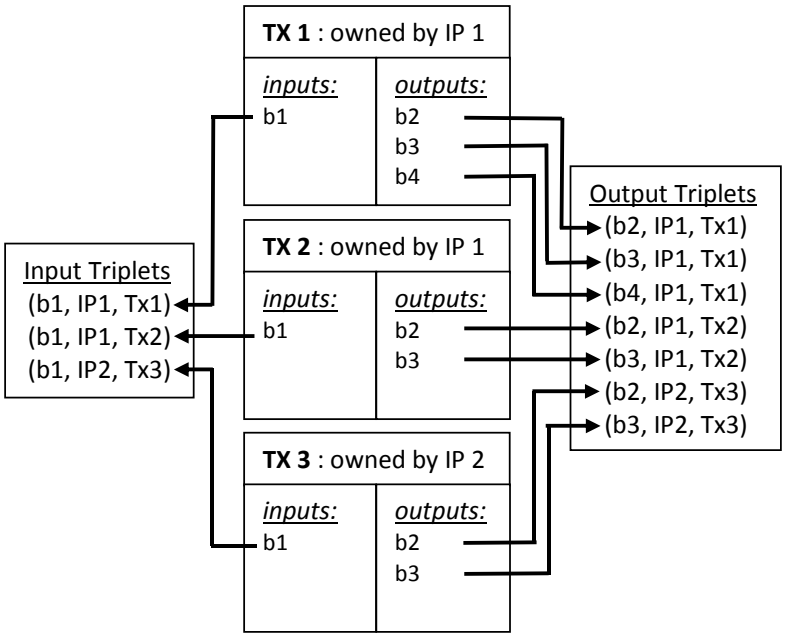

Figure 4.5: Decomposing transactions into triplets involving their internal Bitcoin addresses.

transactions are arguably more likely to be correct. For instance, Figure 4.6 shows what our data looks like at the end of this phase.

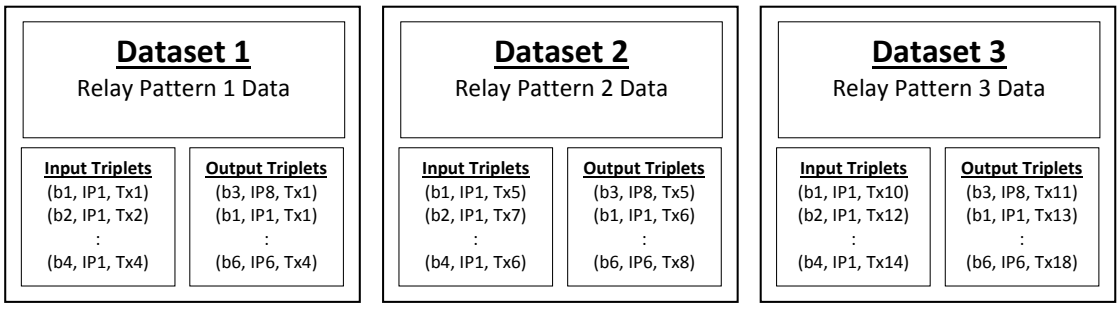

Figure 4.6: This figure illustrates how our data is split according to Relay Pattern at the end of Phase 2. It maintains this split in all later phases.

4.2.4 Phase 3: Computing Pairing Statistics

In Phase 3, we turn our triplet data from Phase 2 into (Bitcoin address, IP) pairings by aggregating over all transactions within the corresponding dataset (from Figure 4.6). This step serves to identify unique (Bitcoin addres, IP) pairings and compute statistics for the occurrence of each pairing within the dataset.

We can think of a transaction owned by IP i which contains Bitcoin address b as a "vote" for the pairing between b and i. We can aggregate our triplet data over these "votes" to form a set

Bitcoin address	IP address	$N_I(b,i)$	C_I	$N_O(b,i)$	C_O
b1	ip1	2	$2/3 = 66.67\%$	0	0
b1	ip2	1	$1/3 = 33.33\%$	0	0
b2	ip1	0	0	2	$2/3 = 66.67\%$
b2	ip2	0	0	1	$1/3 = 33.33\%$
b3	ip1	0	0	2	$2/3 = 66.67\%$
b3	ip2	0	0	1	$1/3 = 33.33\%$
b4	ip1	0	0	1	$1/1 = 100\%$

Table 4.1: The table shows how the 3 transactions from Figure 4.5 would be transformed into pairings between Bitcoin addresses and IPs.

of unique (Bitcoin address, IP) pairings, each with the following metrics:

1. The number of unique transactions owned by IP i that contain Bitcoin address b within their **inputs**.

$$N_I(b,i)$$

2. The number of unique transactions owned by IP i that contain Bitcoin address b within their **outputs**.

$$N_O(b,i)$$

3. The confidence (probability) that a transaction containing Bitcoin address b within its **inputs** is owned by IP i.

$$C_I = \frac{N_I(b,i)}{N_I(b)}$$

4. The confidence (probability) that a transaction containing Bitcoin address b within its **outputs** is owned by IP i.

$$C_O = \frac{N_O(b,i)}{N_O(b)}$$

where $N_I(b)$ and $N_O(b)$ represent the number of unique transactions that contain Bitcoin address b as an input and output, respectively. After formulating our data in this way, this problem becomes much like an evaluation of association rules of the form $b \Rightarrow i$ (see Appendix A), where C_I and C_O represent the confidence scores and $N_I(b,i)$ and $N_O(b,i)$ gauge the support counts for the rule when the Bitcoin address is either an input or an output, respectively.

Table 4.1 shows how the transactions from our example in Figure 4.5 would be transformed into pairings with corresponding computed metrics, assuming those were the only transactions in the dataset being analyzed.

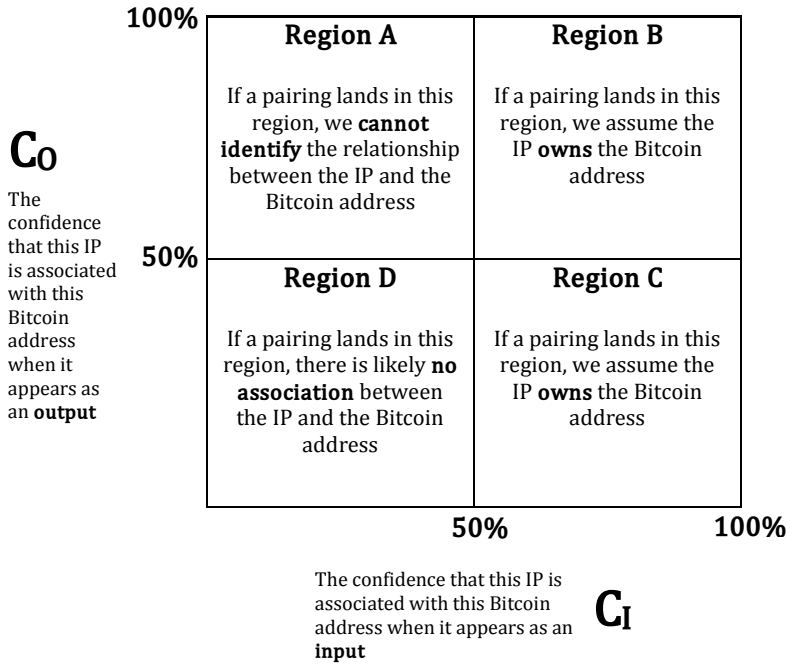

Figure 4.7: Interpretations for the different regions a given (Bitcoin address, IP) pairing could map to on the $C_I \times C_O$ plane.

4.2.5 Phase 4: Identifying Ownership Pairings

Phase 4 involves interpreting the statistics obtained in Phase 3 to figure out which pairings may indicate ownership relationships. The relationship between the Bitcoin address and the IP in a given pairing depends on the region the pairing maps to on the $C_I \times C_O$ plane. Figure 4.7 provides a summary of the interpretations of the different regions on this plane and we explain how we came to these conclusions below.

Region A If a pairing (b, i) maps to Region A ($C_I \leq 50\% \wedge C_O > 50\%$), we can interpret the high C_O as indicating that the majority of transactions sending money to Bitcoin address b (i.e. where b was an output) were created by IP i. The low C_I indicates that this is not the case for transactions drawing on funds from b (i.e. where b was an input). There are two situations that can give rise to this combination of confidence scores:

1. IP i owns Bitcoin address b, using it frequently for receiving change, its own funds (ex: if it is an offline wallet), or payments from others but rarely drawing on those funds for future payments.

2. IP i does not own Bitcoin address b but frequently sends money to the person who does own it. This could indicate a business relationship.

Without additional information to discern between the two cases, we *cannot* form conclusions about Region A pairings.

Region B If a pairing (b, i) maps to Region B ($C_I > 50\% \wedge C_O > 50\%$), we can say that the high C_O and C_I indicate that IP i created both the majority of transactions sending money to Bitcoin address b (i.e. where b was an output) as well as the majority of transactions spending funds tied to b. This would usually occur when a user reuses the same Bitcoin address for making payments and receiving change and thus very likely implies an ownership relationship between the IP and the Bitcoin address.

Region C If a pairing (b, i) maps to Region C ($C_I > 50\% \wedge C_O \leq 50\%$), we know due to the high C_I that IP i created the majority of transactions drawing on funds from b; however, the low C_O signifies that the IP did not create many transactions that involved receiving money using b. Such a combination would occur if a user often sends money from b but does not reuse it for receiving change. Thus, b would be paired as an output with anyone paying the user, but not with the user himself. We classify pairings in Region C as ownership relationships.

Region D Pairings in Region D ($C_I \leq 50\% \wedge C_O \leq 50\%$) do not have high C_I nor C_O, which implies that there may be no association between the Bitcoin addresses and IPs involved. Such pairings are likely the result of noise coming from incorrect ownership hypotheses in Phase 1.

4.2.5.1 Final Ownership Regions

In Phase 1, we assigned owner IPs to every transaction. These owners were then propagated to our (Bitcoin address, IP) pairings in Phase 2. The above interpretation only applies if our definition of "owner" was synonymous with "creator." For Relay Pattern 1 and 2, this is the case; the first or only relayer of a transaction likely created it. To find ownership mappings within Relay Patterns 1 and 2 data, we thus only keep pairings that map to Regions B and C. This makes intuitive sense since transaction creators are associated with inputs and may or may not be associated with outputs, making C_I the only important variable.

For Relay Pattern 3 data, however, the assumption that the "owner" is the creator is not guaranteed to hold. As we described in Section 4.2.2, transactions exhibiting rerelaying behavior could have been rerelayed by either their creator or one of their recipients. Recipients are generally associated with a transaction's outputs and may or may not be associated with its inputs, thus making C_O the only important variable. In the event that an IP is the *recipient* of its assigned transactions, the interpretations for Regions A and C in Figure 4.7 are thus swapped. Unfortunately, there is no way to know if the IPs assigned as owners to Relay Pattern 3 transactions were creators or recipients. Since Region B is the only one where the interpretations overlap for either scenario, we only consider Region B pairings from Relay Pattern 3 data.

4.2.6 Phase 5: Eliminating Insignificant Pairings

In our final Phase, we apply thresholds to the statistical metrics of our ownership pairings from Phase 4 in order to obtain final Bitcoin address-to-IP mappings. There are two types of thresholds to consider - one on support count and one on confidence. Support count tells us

Dataset	Total Ownership Region Pairings	Probabilty of Pairings With Support Count = 1	Probability of Pairings With Support Count ≥ 5	Probabilty of Pairings With Support Count ≥ 10
Relay Pattern 1	1,678,390	99.411%	0.012%	0.004%
Relay Pattern 2	71,714	91.027%	2.047%	1.051%
Relay Pattern 3	27,708	76.732%	3.190%	1.660%

Table 4.2: We see that the vast majority of pairings found in the ownership regions (Regions B and C for Relay Patterns 1 and 2, and Region B for Relay Pattern 3) of each dataset had a support count of 1. Choosing 5 and 10 as thresholds allows us to conservatively eliminate more than 97% of potentially erroneous pairings.

how statistically significant a pairing is, while confidence measures the strength of the ownership relationship between the Bitcoin address and IP.

We found that the vast majority of our (Bitcoin address, IP) pairings had a support count of 1 (see Table 4.2). These results are not surprising; to protect their anonymity, Bitcoin users are encouraged to create a new Bitcoin address for every transaction, thus decreasing the number of times they may become paired with any one address. We also note that within data obtained from anomalous transactions (Relay Pattern 2 and 3), pairings with higher support counts were slightly more common. We decided to use support count thresholds of 5 and 10. These cutoffs allow us to be very conservative since they eliminate over 97% of our pairings. They also make sense from a practical standpoint since in the Bitcoin system, 5 or 10 transactions sent by the same IP containing the same Bitcoin address are highly infrequent.

Our confidence thresholds were determined by the ownership regions from Phase 4 (Figure 4.7). However, the region boundaries only provided the minimal thresholds necessary for interpretations. We were interested in seeing how many ownership pairings would remain as we increased these thresholds to progressively more conservative values. We computed statistics for 7 confidence threshold values for each support count threshold value. The following indicate the criteria a pairing had to meet in order to avoid elimination.

Relay Pattern 1 and 2: Keep pairing (b, i) iff all the following are met:

1. $N_I(b, i) \geq 5$ or 10, depending on the computation being run.
2. $C_I > max(50\%, threshold)$, where $threshold$ is varied from 50% to 100%.

This corresponds to pairings with a support count of at least 5 or 10 that are found in Regions A and B of Figure 4.7.

Relay Pattern 3: Keep pairing (b, i) iff all the following are met:

1. $N_I(b, i) \geq 5$ or 10, depending on the computation being run.
2. $N_O(b, i) \geq 5$ or 10, depending on the computation being run.
3. $C_I > max(50\%, threshold)$, where $threshold$ is varied from 50% to 100%.
4. $C_O > max(50\%, threshold)$, where $threshold$ is varied from 50% to 100%.

The thresholds are kept equal for inputs and outputs. This corresponds to pairings with a support count of at least 5 or 10 for both inputs and outputs that are found in Region B of Figure 4.7.

Table 4.3 shows the final number of ownership pairings for each of our 3 datasets as we varied the thresholds.

Support ≥ 5	# Ownership Pairings Found		
Confidence Threshold	Relay Pattern 1	Relay Pattern 2	Relay Pattern 3
$> 50\%$	178	591	393
$> 60\%$	104	585	362
$> 70\%$	68	577	332
$> 80\%$	39	565	288
$> 90\%$	19	544	243
$> 95\%$	17	542	218
$> 99\%$	16	538	188

Support ≥ 10	# Ownership Pairings Found		
Confidence Threshold	Relay Pattern 1	Relay Pattern 2	Relay Pattern 3
$> 50\%$	53	194	196
$> 60\%$	22	191	183
$> 70\%$	9	190	165
$> 80\%$	5	187	139
$> 90\%$	4	180	121
$> 95\%$	2	178	101
$> 99\%$	1	174	77

Table 4.3: These tables indicate the number of pairings found in each dataset which met the criteria for ownership.

Chapter 5

Discussion

5.1 Results and Assumptions

As we see from Table 4.3, even when applying highly conservative constraints, we were able to map between 252 and 1162 Bitcoin addresses to the IPs that very likely owned them. This shows that it is indeed possible to deanonymize some subset of Bitcoin addresses simply by observing transaction relay traffic.

We note that the vast majority of our final mappings were derived from Relay Patterns 2 and 3 - anomalous transaction traffic. This implies that either (1) most users on the Bitcoin network follow the recommendation of creating a new Bitcoin address for every transaction (thus reducing the support count for any given mapping to 1), or (2) the heuristic of assigning a transaction's ownership to its first relayer is ineffective at best and invalid at worst.

There are indeed several assumptions and caveats to our method. To increase the likelihood that the creator of each transaction was among our directly connected peers, we tried to connect to all listening nodes[1]. However, transactions sent through proxy services such as Tor, I2P, or [17] would still be assigned to incorrect owners since we cannot establish direct connections to their true creators. Incorrect ownership would also be assigned for transactions created by directly connected peers with slow connections, since we may receive their transactions from other peers first. Our statistical approach allows us to be tolerant of incorrect ownership assignments provided that the transactions of such peers do not always arrive through the same intermediary.

There are also several caveats when using our method in the presence of centralized Bitcoin entities such as mixing services and eWallets.

Mixing Services allow users to send their coins to one set of service-controlled addresses and receive them back from a set of unrelated addresses. This breaks any analysis that tries to relate entities by tracking the flow of bitcoins across transactions. Since we do not attempt to connect

[1] We avoided inbound connections to prevent connectioning to Tor/I2P nodes. A listening Bitcoin peer cannot be hidden by Tor or I2P since these technologies only protect the anonymity of people making outbound connections.

different users or find links between an individual user's transactions, *our method is not affected by mixing services.*

eWallets, much like banks, allow users to create accounts which they can use to receive and send money. Users never need to download the Bitcoin software themselves and all of a user's transactions are made on behalf of the user by the eWallet service using keys controlled by the service. We caution that using our method, Bitcoin addresses controlled by an eWallet would be paired with the eWallet despite the funds actually belonging to a different user. This is an unavoidable limitation of our approach.

We note that since other work in this area makes heavy use of flow analysis, they are greatly affected by both eWallets and mixing services.

5.2 Suggestions For Safeguarding One's Anonymity

There are several ways of protecting one's anonymity. Some involve following best-practices, while others make use of publicly available tools (assuming these tools are trusted). In this section, we offer some suggestions for the conscientious Bitcoin user.

5.2.1 Best Practices

People running their own Bitcoin clients can take several steps to protect their anonymity. All of the following suggestions are effective against both flow analysis and IP activity tracking when used correctly.

5.2.1.1 Avoid Address Reuse

For any type of statistical analysis, support/sample size is key. The more a given Bitcoin address is reused, the more transactions it links together, increasing a sample to analyze. To avoid this, two measures should be taken. Every time a new payment is to be received, it is recommended that a new public key is created [28]. This will result in the key appearing in only two transactions - the receiving transaction and the subsequent spending transaction. Furthermore, do not reuse keys for change, since this will force those keys to be used as inputs in later transactions.

5.2.1.2 Thwart Clustering

There are two types of clustering that can be done on keys, and both rely on very restrictive assumptions. Input clustering hinges on the assumption that only one user can create a transaction, and thus all input keys in a multi-input transaction must belong to the same individual. Although it is difficult to avoid using multiple inputs in a transaction, it has become easier to collaboratively create transactions with other individuals. This thwarts input clustering techniques because the keys of different users will be seen as belonging to the same individual, and thus will incorrectly tie many completely unrelated transactions to each other. Output clustering depends on being able to distinguish a real recipient from a change address. One way to make

this difficult is to return change to multiple freshly created addresses. The more outputs there are in a transaction, the more difficult it is to discern which is being used for change.

5.2.1.3 Confuse IP Activity Tracking

Finally, it is important to hinder IP activity tracking, which relies on finding patterns in traffic sent from a given IP address. One alternative is to use a service like Tor or I2P, which will obscure the actual sender of a message by relaying that message through a network of other nodes before sending it out onto the Bitcoin network. Alternatively, certain services can relay users' transactions for them; thus, the transaction would be coming from the service's IP address instead of the user's [17]. If a user sends his own transactions, he should monitor his own relay traffic to make sure it is as random as possible. As we have shown, nodes stuck in abnormal rerelaying states pose a danger to themselves.

5.2.2 Tools

Additionally, a number of services can help people protect their identities and make Bitcoin easier to use. In this section, we discuss two main services available to Bitcoin users - eWallets and mixing/laundry services.

5.2.2.1 eWallets

eWallets offer an alternative to downloading a Bitcoin client and maintaining one's own public and private keys. They operate like banks, and keeping bitcoins in an eWallet is analogous to maintaining a bank account instead of keeping money under the mattress. The eWallet has its own public and private keys, which a user can use for receiving and sending bitcoins. The transactions themselves are created and sent out onto the Bitcoin network by the eWallet. Some eWallets do allow users to have personal public keys and will maintain the corresponding private keys for them. eWallets offer great protection from deanonymization attempts that use flow analysis or IP activity tracking. As with a bank, the money a user puts into the eWallet is not necessarily the same physical money he withdraws. Payments made from an eWallet could have been made on the behalf of anyone using the same eWallet. Because the transactions originate from and are sent out by the eWallet, a user's IP activity is completely safe. Furthermore, key management and anonymity protection are performed by the eWallet service. These services do pose some major risks, however. First, they are a centralized repository of information identifying their users. Second, they are a centralized repository of bitcoins and are more likely targets for theft. Finally, they act as trusted third parties in a system designed for operating without trust; there is no guarantee that the owner(s) of an eWallet service will not abscond with their users' money [25].

5.2.2.2 Mixing/Laundry Services

Mixing and laundry services offer more light-weight ways of obscuring one's bitcoin trail. Unlike eWallets, they are not meant to be used for storing one's bitcoins. Their purpose is to make it difficult to follow a chain of transactions back to a source. Users send their coins to the service's address(es). The coins then get shuffled around and combined in transactions with the coins of others using the service. Eventually, the coins are returned to the original user from a different address. Thus, the transaction in which a user receives back his coins should be completely unrelated to the transaction in which the coins were deposited to the mixing service. Some services will return the coins in one lump sum, while others can return them in chunks at randomly scheduled intervals. Although this does defeat flow analysis, because a user is still running his own Bitcoin client and sending and spending received coins using his own addresses, IP activity tracking is still a threat to the user's anonymity. Furthermore, as with eWallets, there is no guarantee that the mixing service will return the deposited coins.

Chapter 6

Conclusion

Due to the cryptographic techniques used within Bitcoin transactions, users of the currency sometimes mistakenly assume they are using their money completely anonymously. We investigated the feasibility of deanonymizing Bitcoin using network traffic alone. This eliminated our reliance on the assumptions necessary for clustering and flow analysis, which we showed are becoming weaker as the protocol and available Bitcoin services evolve. By creating CoinSeer, a custom Bitcoin client specially designed for mass data collection, we stored all the information relayed on the Bitcoin network for 5 months. We used our knowledge of the Bitcoin protocol to find anomalous transaction relay patterns that we exploited to hypothesize owner IPs for each transaction. We then propagated these owners to the Bitcoin addresses contained within the transactions and investigated the frequencies with which each Bitcoin address appeared with each IP.

We found that despite applying very conservative thresholds, it was still possible to map a subset of Bitcoin addresses to their likely owner IPs using our technique. However, the vast majority of the mappings we established were derived from anomalous relay traffic. We discussed the limitations of our method and proposed steps users could take to help protect their anonymity in light of our (and others') findings.

Appendix A

Applying Association Rules

A.1 Why This Method?

We frame our problem as a search for and evaluation of association rules between Bitcoin addresses and IP addresses. Association rule mining was originally introduced by Agrawal et al. [29] to aid in finding strong, "interesting" relationships among variables in large datasets. The method looks at sets of co-occurring values and computes (a) the likelihood of seeing each unique set of values within the whole dataset ("support"), and (b) the likelihood of seeing the values in a set paired together as opposed to with other values ("confidence"). We provide a more formal description in Section A.2.

We chose this method because it is robust in the presence of incorrect assumptions. For instance, if the owner of a transaction was incorrectly hypothesized in Phase 1 (Section 4.2.2), his co-occurrences with the transaction's Bitcoin addresses will have insignificant confidence scores and support sizes since it is unlikely that he will be paired with these same addresses again unless he is actually associated with them.

We note that, with the exception of "misbehaving" nodes, users who do not use a constant IP address and/or who are careful about reusing their Bitcoin addresses are generally well-protected from this type of analysis. We offer a more detailed discussion on anonymity protection in Section 5.2.

A.2 Association Rule Mining Background

Classic association rule mining involves searching for statistically strong relationships between a subset of items, X, and another item I_j not in X but which is from the finite itemset $I = I_1, I_2, ..., I_m$ of which X is a proper subset. For instance, if the itemset consisted of $\{rice, milk, eggs, flour, bread, cheese, baking\ powder\}$, X could be the set $\{milk, eggs, flour\}$ and I_j could be the *baking powder*.

A co-occurrence of items is called a *transaction*, which we will italicize to distinguish it from a Bitcoin transaction. Continuing our example, a purchase of *milk*, *eggs*, *flour* and *baking powder* would constitute one *transaction*. Agrawal et al. [29] defines a *transaction* t as a binary vector of length m where $t[k] = 1$ if an item I_k from the itemset I can be found in that *transaction* and $t[k] = 0$ otherwise. In our example, the *transaction* would be $t = [\ 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1\ ]$. This is equivalent to defining a *transaction* as a subset of items from I (i.e. $t = \{milk, eggs, flour, baking\ powder\}$).

An association rule is defined as an "implication of the form $X \Rightarrow I_j$, where X is a set of some items from I, and I_j is a single item in I that is not present in X. The rule $X \Rightarrow I_j$ is satisfied in the set of transactions T with confidence factor $0 \leq c \leq 1$ iff at least $c\%$ of transactions in T that satisfy X also satisfy I_j." The notation $X \Rightarrow I_j|c$ is used to indicate that a rule is met with a confidence factor of c. "The support for a rule is defined to be the fraction of transactions in T that satisfy the union of items in the consequent and antecedent of the rule," where X is the antecedent and I_j is the consequent in a given rule [29].

In our example, say we observe that when a customer buys *milk*, *eggs*, and *flour*, he often also buys *baking powder*. We are curious to analyze the relationship between these two sets of items. This equates to looking at the support and confidence for the rule $\{milk, eggs, flour\} \Rightarrow$ *baking powder*. Assume that the supermarket has 100 total customer *transactions* on record. If 20 of them contained *milk*, *eggs*, *flour*, and *baking powder*, then the support for our rule is 20%. If an additional 30 contain *milk*, *eggs*, and *flour* but no *baking powder*, then the confidence for our rule is 40%; 20 of the total 50 *transactions* that contain the antecedent also contain the consequent.

A.3 Association Rules In Our Bitcoin Data

When our data is transformed into triplets of the form (Bitcoin address, IP, Tx #), as discussed in Section 4.2.3, our problem becomes easy to frame as a search for strong association rules of the form $b \Rightarrow i$, where b is a Bitcoin address and i is an IP. Our itemset can be defined as $I = B \cup N$, where B is the set of all Bitcoin addresses and N is the set of all owner IP addresses (a.k.a. nodes). Our *transactions* take on the form (b, i), where $b \in B$ and $i \in N$.

The reader may have noticed that the transaction number (the third part of each triplet) was excluded from the above definitions. It serves a vital role when computing support and confidence. Both of these measures require using *transaction* counts, since in association rule mining, a single *transaction* serves as a vote for the co-occurrence of the items contained inside. Because we split each Bitcoin transaction into (potentially) many association rule *transactions*, we run the risk of double-counting it if we rely directly on *transaction* counts when computing our statistics.

As an example, consider Figure A.1 (which is identical to Figure 4.5). To compute support for a rule $b \Rightarrow i$, we would need to divide by the total number of *transactions* in our dataset. This is correct if we are dealing with the dataset derived from input Bitcoin addresses since the

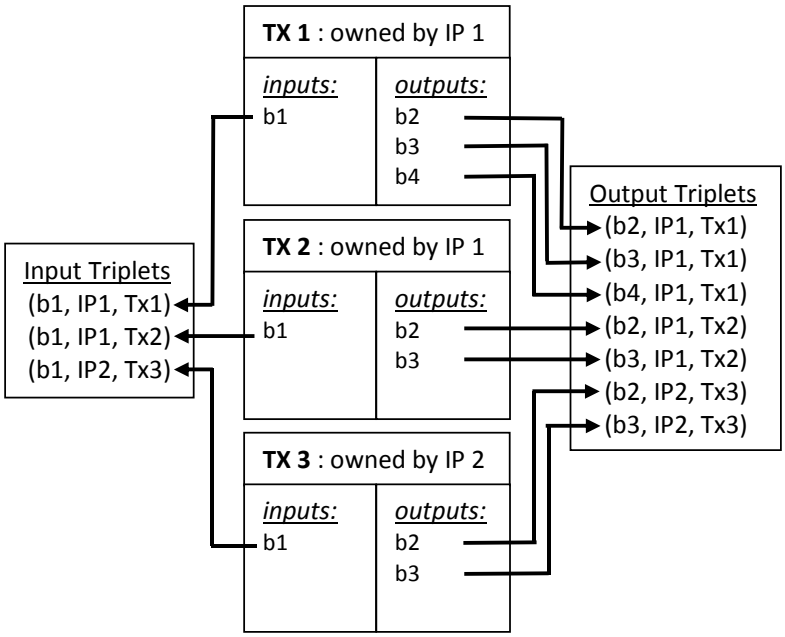

Figure A.1: Decomposing transactions into triplets involving their internal Bitcoin addresses.

count of Bitcoin transactions is equal to the count of association rule *transactions*. If we are considering the outputs dataset, this is not the case; despite having only 3 Bitcoin transactions, we have 7 association rule *transactions* containing output Bitcoin addresses. Thus, if we were computing the support for b2 $\Rightarrow$ IP1, we would mistakenly obtain $\frac{2}{7}$ instead of $\frac{2}{3}$ if we used the *transaction* count. To ensure that a single Bitcoin transaction counted as only one vote for the co-occurrence of the owner IP and each Bitcoin address inside, we made sure to use the unique Bitcoin transaction count over the association rule *transaction* count whenever the two were not equal. We also removed duplicate input and duplicate output Bitcoin addresses from transactions before computing statistics, since this could also lead to double-counting.

Appendix B

Relayer Distributions For Multi-Relayer Transactions

We wanted to see how many of our connected peers knew about each multi-relayer transaction (Relay Patterns 1 and 3 from Section 4.2.2). This count can be used as a measure of the transaction's validity, since by relaying a received transaction, a peer casts a vote of confidence for that transaction.

Figures B.1 and B.2 show the distribution of relayers for Relay Pattern 1 and Relay Pattern 3 transactions, respectively. We see that 95.5% and 92.1% of Relay Pattern 1 and Relay Pattern 3 transactions, respectively, were relayed by at least 1500 unique peers, indicating that the vast majority of the transactions in our data were valid.

# Relayers	# Transactions	Percentage
0-499	144036	3.9233%
500-1499	20567	0.5602%
1500-2499	3043660	**82.9032%**
2500-3499	463078	**12.6133%**

Figure B.1: Distribution of all 3,671,341 Relay Pattern 1 transactions by number of relayers. According to this data, 95.5% of all such transactions were relayed by at least 1500 unique nodes.

# Relayers	# Transactions	Percentage
0-499	16467	6.7904%
500-1499	1700	0.7010%
1500-2499	160531	**66.1975%**
2500-3499	62898	**25.9370%**
3500-4499	902	0.3720%
4500+	5	0.0021%

Figure B.2: Distribution of all 242,503 Relay Pattern 3 transactions by number of relayers. According to this data, 92.1% of all such transactions were relayed by at least 1500 unique nodes.

Bibliography

[1] Nakamoto, S. (2008) "Bitcoin: A peer-to-peer electronic cash system," *Consulted*, **1**, p. 2012.

[2] Fall, K. R. and W. R. Stevens (2011) *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley.

[3] "Trade," `https://en.bitcoin.it/wiki/Trade`.

[4] Matonis, J., "WikiLeaks Bypasses Financial Blockade With Bitcoin," `http://www.forbes.com/sites/jonmatonis/2012/08/20/wikileaks-bypasses-financial-blockade-with-bitcoin/`, *Forbes*, 20 Aug 2012.

[5] NPR, "Silk Road: Not Your Father's Amazon.com," `http://www.npr.org/2011/06/12/137138008/silk-road-not-your-fathers-amazon-com`, *NPR*, 12 Jun 2011.

[6] Roy, J., "Feds Raid Online Drug Market Silk Road," `http://nation.time.com/2013/10/02/alleged-silk-road-proprietor-ross-william-ulbricht-arrested-3-6m-in-bitcoin-seized/`, *Time*, 2 Oct 2013.

[7] "Raw Transactions," `https://en.bitcoin.it/wiki/Raw_Transactions`.

[8] Kao, A. (2004) "RIAA v. Verizon: Applying the Subpoena Provision of the DMCA," *Berkeley Tech. LJ*, **19**, p. 405.

[9] Kaminsky, D., "Black Ops of TCP/IP 2011," `http://www.slideshare.net/dakami/black-ops-of-tcpip-2011-black-hat-usa-2011`, *Black Hat USA 2011*.

[10] "Technical background of version 1 Bitcoin addresses," `https://en.bitcoin.it/wiki/Technical_background_of_version_1_Bitcoin_addresses`.

[11] "Target," `https://en.bitcoin.it/wiki/Target`.

[12] "Bitcoin Block Explorer," `http://blockexplorer.com/`.

[13] "Blockchain.info," `http://blockchain.info/`.

[14] "Bitcoin Wiki," `https://en.bitcoin.it/wiki/`.

[15] "theymos", "Topic: Anonymity," `https://bitcointalk.org/index.php?topic=241.0`, 07 Jul 2010.

[16] "Anonymity," `https://en.bitcoin.it/wiki/Anonymity`.

[17] "Broadcast Transaction," `http://blockchain.info/pushtx`.

[18] Reid, F. and M. Harrigan (2013) "An analysis of anonymity in the bitcoin system," in *Security and Privacy in Social Networks*, Springer, pp. 197–223.

[19] Ron, D. and A. Shamir (2012) "Quantitative Analysis of the Full Bitcoin Transaction Graph," *IACR Cryptology ePrint Archive*, **2012**, p. 584.

[20] Androulaki, E., G. Karame, M. Roeschlin, T. Scherer, and S. Capkun (2012) "Evaluating User Privacy in Bitcoin," *IACR Cryptology ePrint Archive*, **2012**, p. 596.

[21] Meiklejohn, S., M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage (2013) "A fistful of bitcoins: characterizing payments among men with no names," in *Proceedings of the 2013 conference on Internet measurement conference*, ACM, pp. 127–140.

[22] Ober, M., S. Katzenbeisser, and K. Hamacher (2013) "Structure and Anonymity of the Bitcoin Transaction Graph," *Future Internet*, **5**(2), pp. 237–250.

[23] Miers, I., C. Garman, M. Green, and A. D. Rubin (2013) "Zerocoin: Anonymous Distributed E-Cash from Bitcoin," in *IEEE Symposium on Security and Privacy*.

[24] Barber, S., X. Boyen, E. Shi, and E. Uzun (2012) "Bitter to Better - How to Make Bitcoin a Better Currency," in *Financial Cryptography and Data Security*, Springer, pp. 399–414.

[25] Moore, T. and N. Christin (2013) "Beware the Middleman: Empirical Analysis of Bitcoin-Exchange Risk," *Financial Cryptography and Data Security*, **7397**, pp. 455–466.

[26] Koshy, P. (2013) *CoinSeer: A Telescope Into Bitcoin*, Master's thesis, Pennsylvania State University.

[27] "Network," `https://en.bitcoin.it/wiki/Network`, standard Relaying Section (accessed of September 2, 2013).

[28] "Some things you need to know," `http://bitcoin.org/en/you-need-to-know`.

[29] Agrawal, R., T. Imieliński, and A. Swami (1993) "Mining association rules between sets of items in large databases," in *ACM SIGMOD Record*, vol. 22, ACM, pp. 207–216.