An Analysis of Anonymity in Bitcoin Using P2P Network Traffic

Philip Koshy, Diana Koshy, and Patrick McDaniel

Pennsylvania State University, University Park, PA 16802, USA

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.

Keywords: Bitcoin, anonymity, CoinSeer

1 Introduction

Bitcoin is a decentralized peer-to-peer crypto-currency first proposed and implemented by Satoshi Nakamoto, a likely pseudonym, in 2009 [1]. It allows end-users to create 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 [2]. We have also seen the birth and recent death of the Silk Road, a Bitcoin marketplace once called "the Amazon.com of illegal drugs" [3, 4].

Previous studies (discussed in Section 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. 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 [6]. In this work, we set out to determine 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 [5].

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 paper is organized as follows. Section 2 gives a brief background of the Bitcoin protocol, while Section 3 provides an overview of related work. In Section 4, we discuss CoinSeer, our custom-built Bitcoin client. Section 5 presents several interesting cases we discovered that inspired our later methodology. We outline our final approach in Section 6, discussing how to create, evaluate, and prune Bitcoin address-to-IP mappings. In conclusion, Section 7 discusses our results, as well as the caveats and limitations of our method.

2 Background

Bitcoin is a decentralized currency which requires certain participants called miners to validate financial transactions. 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 these 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.

2.1 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 is simply a public key to which a number of transformations and hashes have been applied. Thus, the terms Bitcoin address and public key can be 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 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 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.

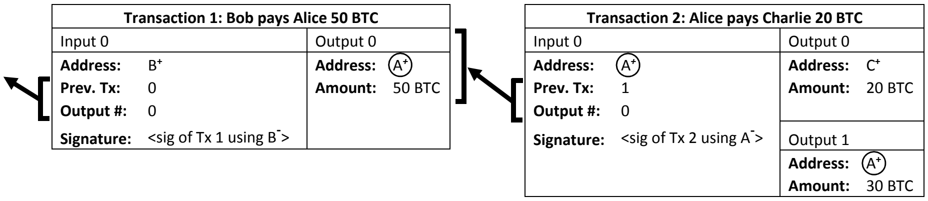


Fig. 1. This figure demonstrates how Alice, who owns Bitcoin address A, would create a new transaction (Transaction 2) which spends bitcoins received earlier (Transaction 1). Note that the Bitcoin address of the input must match the Bitcoin address of the referenced output. Note also that the sender 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 her change to a different address she owns. 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.

2.2 P2P Relaying

Bitcoin uses a gossip protocol [7] 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 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 outputs claimed by the inputs cannot be found.

Table 1. Types of Ignored Transactions

3 Related Work

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 [8] 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 [9] 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. [10] again took a similar approach, using data from a simulation of bitcoin usage in 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 their own heuristic. Meiklejohn et al. [11] 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. [12], 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. [13], arguing that Bitcoin is not truly anonymous, proposed an extension to the protocol that uses cryptography to make transactions fully anonymous. Barber et al. [14] discussed the various vulnerabilities inherent to Bitcoin, finally proposing and outlining a trust-free mixing service. Moore and Christin [15] 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.

4 CoinSeer: The Need For A Custom Bitcoin Client

Inspired by Dan Kaminsky's 2011 Black Hat presentation [5], we decided to analyze traffic patterns on the Bitcoin network to see if it was possible to create mappings from Bitcoin addresses to IPs. To increase the likelihood of receiving transactions directly from their creators in a gossip protocol, we had to connect to all listening peers. We actively collected all data, along with its IP information, being relayed on the network and stored it for offline processing.

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. 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; 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.

5 Discovering Anomalous Relay Patterns

When we began analyzing our collected data, we manually looked for interesting behavior. The following are specific cases that led us to believe that transaction relay behavior may be used to map Bitcoin addresses to IPs.

Case 1: On August 31, 2012, we received a transaction from a single IP that was never relayed again. This "single-relayer" transaction is highly unusual for a P2P system using a gossip protocol; we would expect to have received it from the majority of the approximately 2,500 peers we were connected to at the time.

On September 3, 2012, a new transaction with the same inputs and outputs was relayed network-wide and accepted into the blockchain. Given this information, can we assume the sole relayer of the first transaction was its creator and thus owns the Bitcoin addresses inside?

Case 2: On August 22, 2012, a single IP sent us 11,730 unique transactions within a 74-second window. The median rate we received transactions was *only* 43 per minute. Because these transactions were already in the block chain, they were not relayed by anyone else, making them "single-relayer" transactions. Using connection metadata, we saw that this large transaction dump corresponded with this user upgrading to a newer version of the Bitcoin client he was using. Could all of these belong to the single relayer?

Case 3: For 52 days, beginning on July 24, 2012, we received the *same* transaction from a single IP approximately once every hour; no one else on the network relayed it. The peer then disconnected, only for a new IP to connect and exhibit the same behavior for the next 23 hours. This occurred again with the appearance of a third IP, finally going silent a day later. Why would a transaction be continually rerelayed, and what connection does it have to its rerelayers?

6 Methodology

Manually discovering instances of exploitable anomalous behavior proved to be unscalable. We attempted to generalize the patterns we observed, some of which were demonstrated by the cases in Section 5, in order to come up with a more algorithmic approach for mapping Bitcoin addresses to the IPs that own them. This approach requires six phases:

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.

6.1 Phase 0: Pruning Transaction Data

By the end of our 5 month collection period, we had relayer information for 5,617,202 transactions. This number included some noise; there were 57,087 transactions whose hashes were advertised but which were never relayed, as well as 300 that contained a Bitcoin address we could not parse. These were removed from consideration. Additionally, we removed 114,100 transactions that exhibited relay patterns which made establishing ownership ambiguous (see Section 6.2, and Figure 5 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 [16]. 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.

6.2 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[2]). 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. As demonstrated by the three cases in Section 5, we found that certain transactions exhibited atypical behavior; the transactions from Case 1 and 2 were relayed by only a single IP, while Case 3 demonstrated rerelaying behavior. Whereas for a typical transaction, we can only hope that the creator was its first relayer[3], 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.

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 2 to demonstrate ownership assignment for transactions exhibiting this relay pattern.

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. Cases 1 and 2 from Section 5 fall into this category.

[2] We note that this does not mean the creator owns the funds associated with that Bitcoin address (see discussion on eWallets in Section 7).

[3] We discuss why this assumption is flawed in Section 7.

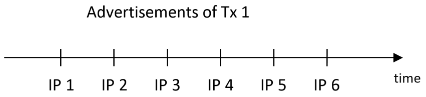


Fig. 2. In the timeline at right, Tx 1 is being relayed once by each IP. Since this is normal behavior, there is no additional information to exploit. In this case, we simply choose the **first relayer** - IP 1 - as the "owner."

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.

This behavior may arise when a peer creates an invalid transaction that its immediate peers reject. 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 3.

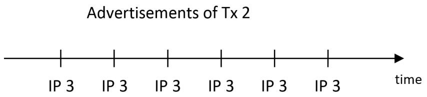


Fig. 3. The timeline at right shows the advertisements of Tx 2. Since only one IP ever relayed this transaction, there is no ambiguity; we assign the **single relayer** - IP 3 - as the "owner."

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. Case 3 from Section 5 demonstrated this behavior. 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 [17]. 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:

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 provides an example of ownership assignment for transactions in this group.

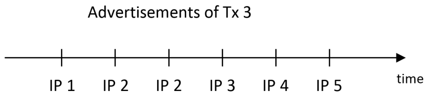


Fig. 4. For Tx 3, everyone but IP 2 is exhibiting the expected behavior of sending the transaction only once. Since only the sender or recipient of coins in a transaction is supposed to rerelay that transaction, we assign the **single rerelayer** - IP 2 - as the "owner."

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 5 provides an example of why ownership assignment for transactions in this group is ambiguous.

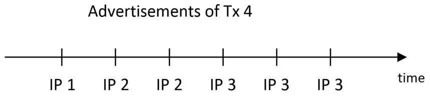


Fig. 5. This is similar to Tx 3, but there are now multiple rerelayers. This makes ownership assignment more **ambiguous**. Do we assign it to the first rerelayer, or the one with the most relays? To err on the side of caution, we **removed** transactions with more than one rerelayer from consideration.

6.3 Phase 2: Creating (Bitcoin Address, IP) Pairings

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 6 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 transactions are arguably more

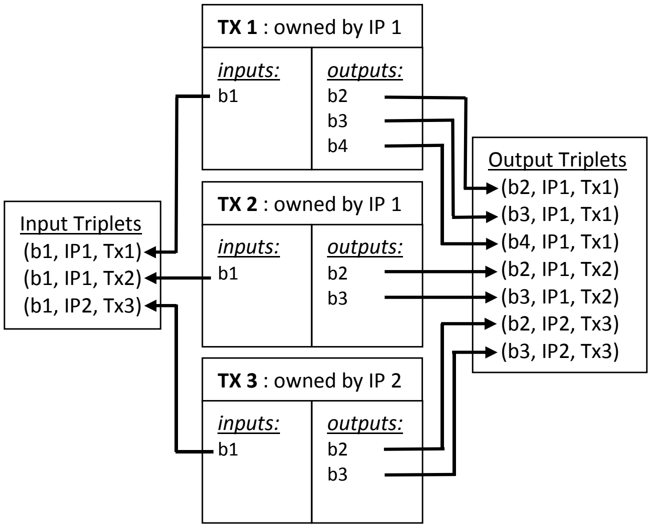


Fig. 6. Decomposing transactions into triplets involving their internal Bitcoin addresses.

likely to be correct. For instance, Figure 7 shows what our data looks like at the end of this phase.

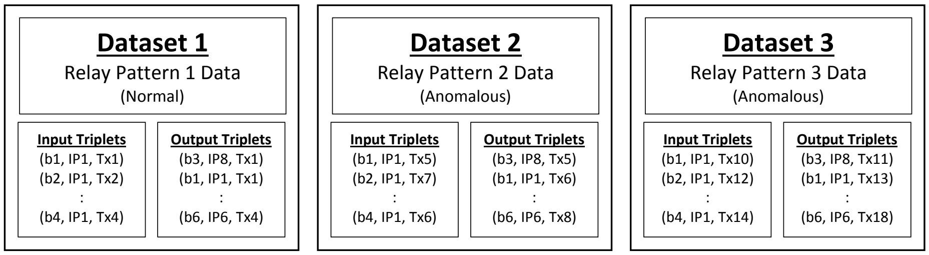


Fig. 7. This figure illustrates how our data is split according to Relay Pattern at the end of Phase 2. It maintains this split in all later phases.

6.4 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 7). This step serves to identify unique (Bitcoin address, 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 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 $b \rightarrow i$ [18], 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.

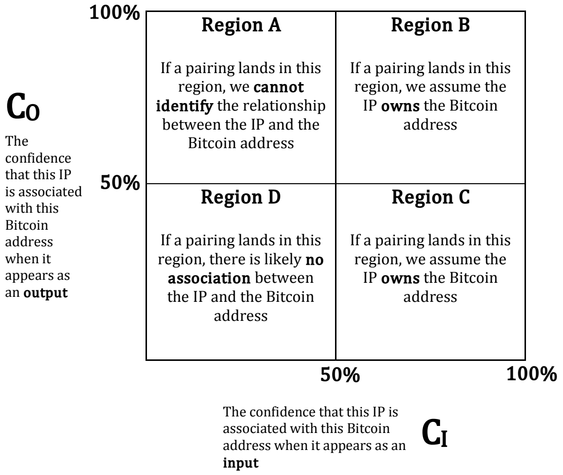
Table 2 shows how the transactions from our example in Figure 6 would be transformed into pairings with corresponding computed metrics, assuming those were the only transactions in the dataset being analyzed.

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 2. The table shows how the 3 transactions from Figure 6 would be transformed into pairings between Bitcoin addresses and IPs.

6.5 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 8 provides a summary of the



If a pairing lands in this region, we **cannot identify** the relationship between the IP and the Bitcoin address

If a pairing lands in this region, we assume the IP **owns** the Bitcoin address

If a pairing lands in this region, there is likely **no association** between the IP and the Bitcoin address

If a pairing lands in this region, we assume the IP **owns** the Bitcoin address

The confidence that this IP is associated with this Bitcoin address when it appears as an **output**

The confidence that this IP is associated with this Bitcoin address when it appears as an **input**

Fig. 8. Interpretations for the different regions a given (Bitcoin address, IP) pairing could map to on the $C_I \times C_O$ plane.

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.

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 6.2, 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 8 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.

6.6 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 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 3). 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.

Dataset	Total Ownership Region Pairings	Probability of Pairings With Support Count = 1	Probability of Pairings With Support Count ≥ 5	Probability 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 3. 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.

Our confidence thresholds were determined by the ownership regions from Phase 4 (Figure 8). 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. $C_I > \mathit{threshold}$, 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 8.

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. $C_I > \mathit{threshold}$, where *threshold* is varied from 50% to 100%.
4. $C_O > \mathit{threshold}$, 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 8.

Table 4 shows the final number of ownership pairings for each of our 3 datasets as we varied the thresholds. Table 5 shows the corresponding number of unique owner IP addresses involved within these pairings.

7 Conclusion

As we see from Table 4, even when applying highly conservative constraints, we were able to map between 252 and 1,162 Bitcoin addresses to the IPs that very

Support ≥ 5	# Ownership Pairings Found		
Confidence Threshold	Relay Pattern 1 (Normal)	Relay Pattern 2 (Anomalous)	Relay Pattern 3 (Anomalous)
> 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 (Normal)	Relay Pattern 2 (Anomalous)	Relay Pattern 3 (Anomalous)
> 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. These tables indicate the number of pairings found in each dataset which met the criteria for ownership.

Support ≥ 5	# Unique "Owners"		
Confidence Threshold	Relay Pattern 1 (Normal)	Relay Pattern 2 (Anomalous)	Relay Pattern 3 (Anomalous)
> 50%	50	168	184
> 60%	35	167	170
> 70%	28	165	157
> 80%	19	163	139
> 90%	13	162	115
> 95%	12	162	106
> 99%	11	161	92

Support ≥ 10	# Unique "Owners"		
Confidence Threshold	Relay Pattern 1 (Normal)	Relay Pattern 2 (Anomalous)	Relay Pattern 3 (Anomalous)
> 50%	17	89	120
> 60%	10	88	108
> 70%	6	88	99
> 80%	4	87	83
> 90%	4	87	72
> 95%	2	87	63
> 99%	1	86	50

Table 5. These tables indicate the number of unique owner IPs among the final ownership pairings from Table 4.

likely owned them. From Table 5, we see that these mappings were not simply the result of one or two misbehaving IPs; at least 100 different "owners" were associated with Bitcoin addresses that appear to belong to 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[4]. However, transactions sent through proxy services such as Tor, I2P, or the tool provided in [19] 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

[4] We avoided inbound connections to prevent connecting 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.

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, which both greatly affect other work in this area that relies on flow analysis.

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 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. However, we argue that mappings involving eWallet IPs are still valuable since such services can be pressured for internal client information.

Taking these limitations and our results into account, we conclude that some degree of deanonymization is possible within the Bitcoin system and we urge users to take advantage of the many existing recommendations and services offered to them in order to protect their privacy.

Acknowledgments

This material is based upon work supported by the National Science Foundation Grants No. CNS-1228700 and CNS-0905447. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. Nakamoto, S. (2008) “Bitcoin: A peer-to-peer electronic cash system,” *Consulted*, **1**, p.2012.
2. 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.
3. 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.

4. 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.
5. 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*.
6. Kao, A. (2004) "RIAA v. Verizon: Applying the Subpoena Provision of the DMCA," Berkeley Tech. LJ, 19, 405.
7. Fall, K. R. and W. R. Stevens (2011) *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley.
8. 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.
9. Ron, D. and A. Shamir (2012) "Quantitative Analysis of the Full Bitcoin Transaction Graph," *IACR Cryptology ePrint Archive*, **2012**, p. 584.
10. Androulaki, E., G. Karame, M. Roeschlin, T. Scherer, and S. Capkun (2012) "Evaluating User Privacy in Bitcoin," *IACR Cryptology ePrint Archive*, **2012**, p. 596.
11. Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., and Savage, S. (2013, October), "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.
12. Ober, M., S. Katzenbeisser, and K. Hamacher (2013) "Structure and Anonymity of the Bitcoin Transaction Graph," *Future Internet*, **5**(2), pp. 237-250.
13. 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.*
14. 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.
15. Moore, T. and N. Christin (2013) "Beware the Middleman: Empirical Analysis of Bitcoin-Exchange Risk," *Financial Cryptography and Data Security*, **7397**, pp. 455-466.
16. "Raw Transactions," https://en.bitcoin.it/wiki/Raw_Transactions.
17. "Network," https://en.bitcoin.it/wiki/Network, Standard Relaying Section (accessed of September 2, 2013).
18. Agrawal, R., T. Imielinski, and A. Swami (1993) "Mining association rules between sets of items in large databases," in *ACM SIGMOD Record*, vol. 22, ACM, pp. 207-216.
19. "Broadcast Transaction," http://blockchain.info/pushtx.