

RESEARCH ARTICLE

A Bayesian approach to identify Bitcoin users

Péter L. Juhász^{1,2}, József Stéger^{1*}, Dániel Kondor^{1,3}, Gábor Vattay¹

1 Dept. of Physics of Complex Systems, Eötvös Loránd University, Budapest, Hungary, 2 Business Unit IT & Cloud Products, Ericsson Telecommunications Hungary, Budapest, Hungary, 3 SENSEable City Laboratory, Massachusetts Institute of Technology, Cambridge, MA, United States of America

* steger@complex.elte.hu

OPEN ACCESS

Citation: Juhász PL, Stéger J, Kondor D, Vattay G (2018) A Bayesian approach to identify Bitcoin users. PLoS ONE 13(12): e0207000. <https://doi.org/10.1371/journal.pone.0207000>

Editor: Ivan Olier, Liverpool John Moores University, UNITED KINGDOM

Received: May 14, 2018

Accepted: October 23, 2018

Published: December 13, 2018

Copyright: © 2018 Juhász et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: Data collected in the measurement experiments are available at https://figshare.com/projects/A_Bayesian_Approach_to_Identify_Bitcoin_Users/56387.

Funding: The authors thank the Hungarian National Research, Development and Innovation Office under Grant No. 125280 and the grant of Ericsson Ltd. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: There are no conflicts of interest to disclose. The commercial affiliation and the funding received do not alter our adherence to

Abstract

Bitcoin is a digital currency and electronic payment system operating over a peer-to-peer network on the Internet. One of its most important properties is the high level of anonymity it provides for its users. The users are identified by their Bitcoin addresses, which are random strings in the public records of transactions, the *blockchain*. When a user initiates a Bitcoin transaction, his Bitcoin client program relays messages to other clients through the Bitcoin network. Monitoring the propagation of these messages and analyzing them carefully reveal hidden relations. In this paper, we develop a mathematical model using a probabilistic approach to link Bitcoin addresses and transactions to the originator IP address. To utilize our model, we carried out experiments by installing more than a hundred modified Bitcoin clients distributed in the network to observe as many messages as possible. During a two month observation period we were able to identify several thousand Bitcoin clients and bind their transactions to geographical locations.

1 Introduction

Bitcoin is the first widely used digital currency, developed by Satoshi Nakamoto after the beginning of the financial crisis in 2009 [1]. A distinctive feature of Bitcoin is that there is no central authority overseeing transactions, users are connected via a peer-to-peer network where they announce any transaction they wish to make. Transactions can then be validated by anyone using the publicly available list of transactions, the *blockchain*, which is in turn generated in a proof-of-work system. Cheating (e.g. including invalid transactions in the blockchain) thus would require one entity to control more than 50% of the computing power that users dedicate to generating the blockchain. In accordance with the decentralized nature of the system, the specifications of the network protocol is publicly available, while several open-source client programs implementing the protocol exist [2].

One of the key characteristics of Bitcoin is the high amount of anonymity it provides for its users [3]. Although one can learn the details of the transactions via the blockchain, it is still unknown who the users initiating those transactions are. This is possible since as there is no authority overseeing the operation of the system, users do not need to provide any form of identification to join; anyone with an Internet connection can download a client program, which then allows them to generate any number of Bitcoin addresses that they can use in the

all PLOS ONE policies on sharing data and materials.

transactions to send or receive Bitcoins. This results in that the identity of Bitcoin users is only revealed if they publish their Bitcoin address or this information is intercepted in some way outside the Bitcoin system. While anonymity is not among the main design goals of the Bitcoin system [3], Bitcoin is widely considered as a highly anonymous way of performing financial transactions and is often utilized for illegal uncontrolled payments [4], along legal uses where the involved parties do not wish to disclose their identities to controlling entities in the traditional financial system, e.g. banks or governments.

In the paper, we present a probabilistic model based on the information propagating over the Bitcoin network, which gives the possibility of identifying the users initiating the transactions. In this case, identification means binding the transactions to the IP addresses where they were created.

The basic idea consists of two main steps. First, the probability is determined for each transaction that a specific client (identified by its IP address) created it. Assuming that the creator of the transaction controls the Bitcoin addresses from which money is sent in it, this step then results in possible IP address—Bitcoin address pairings. Next, the most likely Bitcoin address—client pairings are identified by combining the probabilities in the list of pairings compiled in the previous step. This is further elaborated by grouping Bitcoin addresses that belong to the same user with high probability based on the transaction network. Finally, the geographical localization of the IP addresses opens the door for a large scale analysis of the distribution and flow of Bitcoin.

The rest of the paper is structured as follows. Section 3 discusses the relevant characteristics of Bitcoin and provides the necessary background for the further sections. In section 4, the mathematical model used for the deanonymization is explained. The data collection is described in section 5. Section 6 presents the results of the application of the model. Finally the method described in this study is compared to the related works of the topic in section 2.

2 Related work

In accordance with the innovative ideas behind it and the high amount of interest it generated, there has been a significant amount of work focusing on Bitcoin and cryptocurrencies. Works focusing on privacy and anonymity show that the statistical processing of a large amount of seemingly insignificant information can take the attacker closer to reveal the identity of people using Bitcoin.

Androulaki, et al. [5] evaluated the privacy of Bitcoin by analyzing the system using a simulator. After grouping Bitcoin addresses they used behavior-based clustering techniques (K-Means and Hierarchical Agglomerative Clustering algorithms) to bind the Bitcoin addresses to real users.

Reid and Harrigan [6] used mainly offline data processing of the blockchain to analyze the transaction graph. They identified its clusters and components, and analyzed the degree distribution of the user network. They also showed that the analysis of publicly available data from social websites and forums can also reveal the Bitcoin addresses of some users, similarly to previous work in different contexts [7].

Biryukov, Khovratovich and Pustogarov provided a method which connects the users to IP addresses [8]. They connected to all publicly available Bitcoin nodes (servers) and listened the messages they were relaying. They used Bitcoin's peer discovery mechanism to link transactions to their originators even if these do not accept incoming connections: the servers that broadcast the newly connected clients' IP addresses were assumed to be the same set of servers which first relayed their transactions. The difficulty of this method is that a lot of connections have to be established to reach good results as the number of the servers increases. On the

other hand, it promises results for Bitcoin clients which monitoring nodes cannot directly connect to (e.g. because they are protected by firewalls and connect to a limited). In contrast, our methodology requires direct connections to the originators and we thrive to achieve this by running a lot of Bitcoin clients accepting a large number of incoming connections. While they use a fixed number of message relays to infer the local network of the originator, we use a short initial time span for message broadcasts to infer the actual originator. A further main difference in our methodology is combining information from many transactions and linking addresses based on the blockchain to provide more transactions per Bitcoin user in that step. Our probabilistic approach could be combined with the methodology in [8] in order to identify the “hidden” Bitcoin nodes with higher probability. This also allows linking Bitcoin address groups belonging to the same user based on them being originated from the same client, even if these addresses would not be possible to link based only on the blockchain.

Koshy et al. also monitored the messages about the transactions and they classified the transactions to distinct relay patterns [9]. After applying heuristics to determine the possible owner IP addresses of the transaction, they computed simple aggregate statistics to filter out the correct Bitcoin address—IP address pairings for both input and output addresses.

Venkatakrishnan et al. proposed a new message relay mechanism called *dandelion*, which could prevent the nodes to be deanonymized with a high probability [10]. They proposed that the message propagation should have two phases: first, the message is sent to exactly one randomly chosen connected client for a random number of hops by every client, and after the first phase the message could be further broadcast with a Poisson process from the nodes that received the transaction. The authors also highlighted that requirements of the high level of anonymity and low latency are properties that can only be improved at each others expense.

Decker et al. [11] and Donet Donet et al. [12] were among the first to empirically study the peer-to-peer network that enables Bitcoin to operate, characterizing some of the important properties of information propagation. Neudecker et al. [13] present a method for inferring the topology of the peer-to-peer network based on the observation of the message propagation process. Their methodology is similar to ours as they connect to a large number of Bitcoin nodes and observe messages received announcing new transactions. The main difference is in the focus of the study: while our goal is to identify the first node announcing a transaction without the knowledge of the underlying peer-to-peer network topology, the authors in Ref. [13] aim to reconstruct the network topology but do not consider linking Bitcoin addresses to nodes in the network. In further recent work, Goldfeder et al. [14] show that online merchants accepting cryptocurrencies potentially leak substantial information that would allow third-parties to identify the transactions in the blockchain, thus linking information collected via tracking and cookies to Bitcoin addresses; Miller et al. [15] focus on Monero, a cryptocurrency with stronger anonymity features and show that under certain circumstances it might still be possible to defeat Monero’s mixing mechanism and thus its enhanced anonymity as well. Finally, we note that a good review of recent developments, including research focusing on security and privacy in Bitcoin can be found in the work of Conti et al. [16]. In their recent study, Wang and Postugarov [17] also investigate transaction information propagation in the Bitcoin network. They attempt to characterize clients that are operating behind TOR, NAT or firewalls as well. From their seven days data they report 50 unreachable clients are involved in 43% of transaction propagation in the network, and that many unreachable clients are running in public cloud services, potentially to crawl the Bitcoin network.

Although Bitcoin lacks a material basis and has no territorial currency zone Pel in his thesis [18] shows its geographical manifestation. Mining patterns, user procurement of the cryptocurrency and Bitcoin related startups are examined in [18]. These peripheral processes can naturally be linked to geographic space and strong correlations are found between the Bitcoin

volume and financial activity, density of computer center and location of mining pools. In a recent thesis Brown [19] studies the geographical distribution of Bitcoin and Ethereum mining pools and find connection to electricity pricing, TOR exit node distribution.

Basically the following common methods are used to reveal the identities of Bitcoin users:

1. analysis of transactions with multiple input and grouping the input Bitcoin addresses of the same transactions;
2. analysis of Bitcoin flow in the transaction graph using clustering techniques;
3. analysis of propagating network-layer information to bind their content to the users,
4. and finally using publicly available information (e.g. in forums) to connect Bitcoin addresses to identities.

The methodology presented in our work combines the 1 and 3 types of approaches, mainly based on statistical processing of network propagation properties.

3 The main characteristics of Bitcoin

In order to use Bitcoin one has to connect to the Bitcoin network using an open-source *client program* [2]. In this work, we concentrate on the Bitcoin Core client [2], whose source code we inspected and modified for the purpose of data collection. By default, this client establishes eight connections to other clients. If there is a link between two clients, they are *connected*. Clients exchange information of different types, e.g. the transactions they know about, their state, cryptographic signatures and others through the network. This is necessary for the validation of the transactions as it is done by the entire network.

In case of Bitcoin transactions, *Bitcoin addresses* play similar role as the bank account numbers in regular currency transactions. However, there are two major differences:

- each user may have as many Bitcoin addresses as they would like to
- and multiple source and destination Bitcoin addresses can be involved in a single transaction.

In case of the Bitcoin Core client program that was in operation at the time of the measurement, when a user initiates a transaction, the client program (the *originator*) relays a message to a randomly chosen connected client in every 100ms time interval. This method is referred to as *trickling*, and its goal is to hide the source of the transaction. The clients receiving this message (which are not the originators of the transaction) use a slightly more complex algorithm to further send the information. Besides trickling, they also relay the message to the other clients with a probability of 1/4 (in every 100ms). We expect that other types of clients apply the same mechanisms to protect the privacy of the users.

We note that as of today, the previously described mechanism for relaying transactions has been changed in the case of the Bitcoin Core client. Currently every client maintains a queue for the messages to be relayed for each connected clients and relays them according to a Poisson process. The parameter of the process is 5 sec for incoming connections and 2.5 sec for outgoing connections. In this work, we consider the previously described method which was in use during the time of our data collection; we believe that our model could be used for the latter case as well with minor modifications.

In accordance with the previously described methodology, the network relies on clients relaying transactions to have them spread throughout the entire network. As a consequence an arbitrarily chosen client is not necessarily directly informed about the transaction by the originator (see Fig 1).

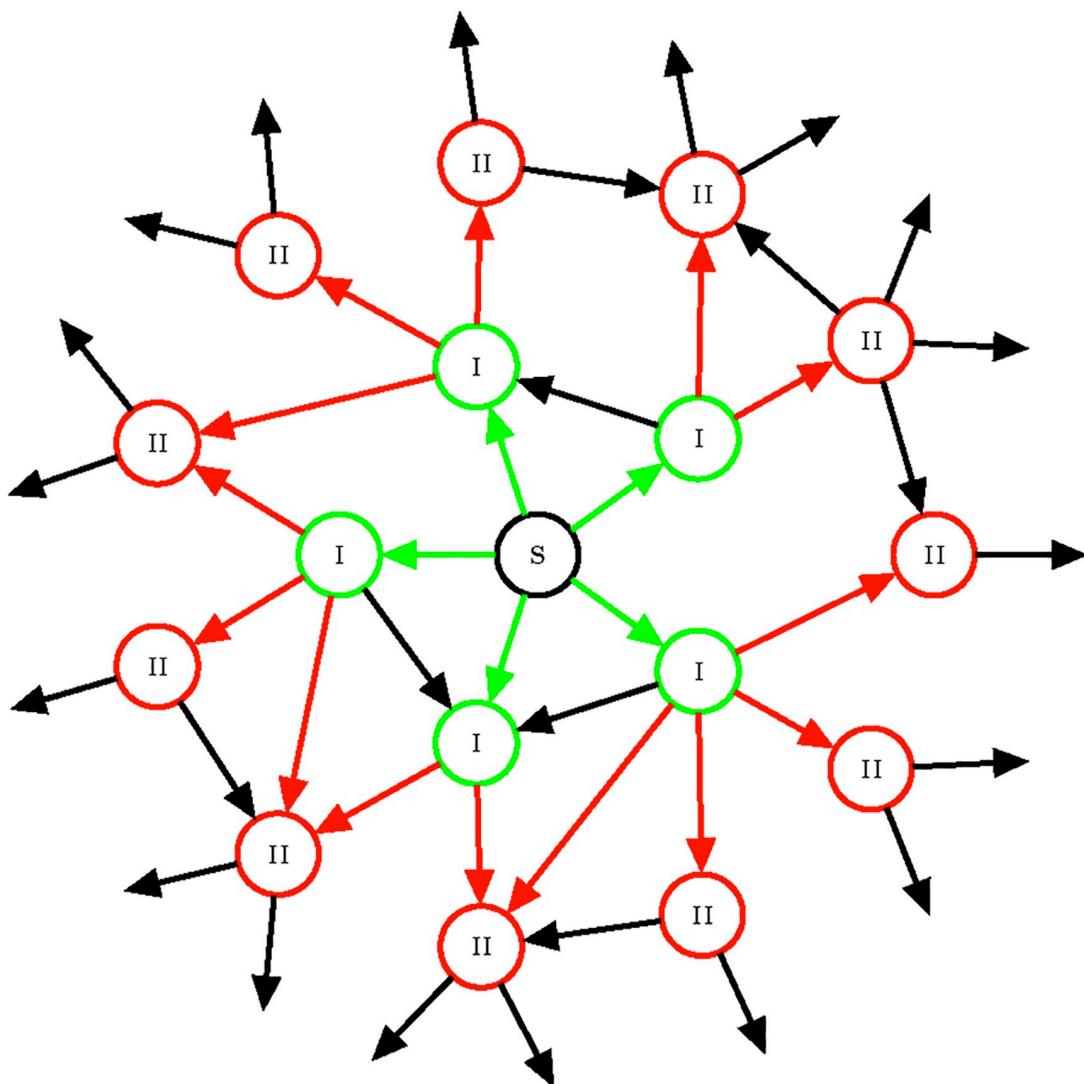


Fig 1. A new transaction is initiated by the client “S”. At first it informs the clients denoted by “I” (they are informed directly from the originator, so only the trickling method is used for the relay). Then, these clients relay the transaction further—among possibly other I type clients—to the ones denoted by “II”.

<https://doi.org/10.1371/journal.pone.0207000.g001>

As no state, bank, institute or organization controls or ensures the validity of Bitcoin transactions, cryptographic methods are used by the whole Bitcoin community for this purpose. The security of Bitcoin is based on the blockchain. In this study the source Bitcoin addresses, the destination Bitcoin addresses, the timestamps and the transferred volume of Bitcoin is extracted from the blockchain for each transaction.

If the owners of the Bitcoin addresses were known, the blockchain would reveal all of the transactions of each Bitcoin user. The open nature of the system mitigates this concern, as anyone can generate any number of Bitcoin addresses without having to reveal their identity. Nevertheless, if a Bitcoin address can be linked to someone (either because they share it in order to receive Bitcoins or by any other method), the transaction history of that Bitcoin address can be trivially retrieved from the blockchain. Thus, keeping the association between Bitcoin addresses and real identities in secret is crucial for users who wish to maintain their privacy.

4 A Bayesian method for the identification of Bitcoin users

In this section we present the methodology to assign probabilities to the distinct IP address–user pairings, which consists of three main steps.

An overview of the process is illustrated in [Fig 2](#).

First, the propagating messages are observed and recorded by several *monitoring clients* in order to cover as great part of the network as possible. For each transaction, monitoring clients record the list of clients who relayed the transaction in the *first time segment* (see the definition in the next subsection). They are the possible originators of the transaction. After some theoretical considerations, we assign probabilities to each client that show the probability of them being the originator, separately for each transaction that we recorded.

Next, the blockchain is used to group the Bitcoin addresses owned by the same user. Additionally, the blockchain also enables to calculate the balances of the users for further analysis.

Last, by having possibly several transactions of the same Bitcoin address and the grouping of Bitcoin addresses by user allows us to combine measurements from multiple transactions to identify users with higher confidence. By combining the probabilities from the first step, the users (and their balances) are paired with the clients that are most likely the originators of their transactions. The clients can be geographically localized through their IP addresses, which allows the determination of the geographical distribution and flow of Bitcoins.

Step 1: Individual probabilities

Let us consider a single transaction observed by one *monitoring client*.

A monitoring client connected to the originator does not necessarily receive the message from the originator first, because in some cases it can be relayed faster through a mediator client ([Fig 3](#)).

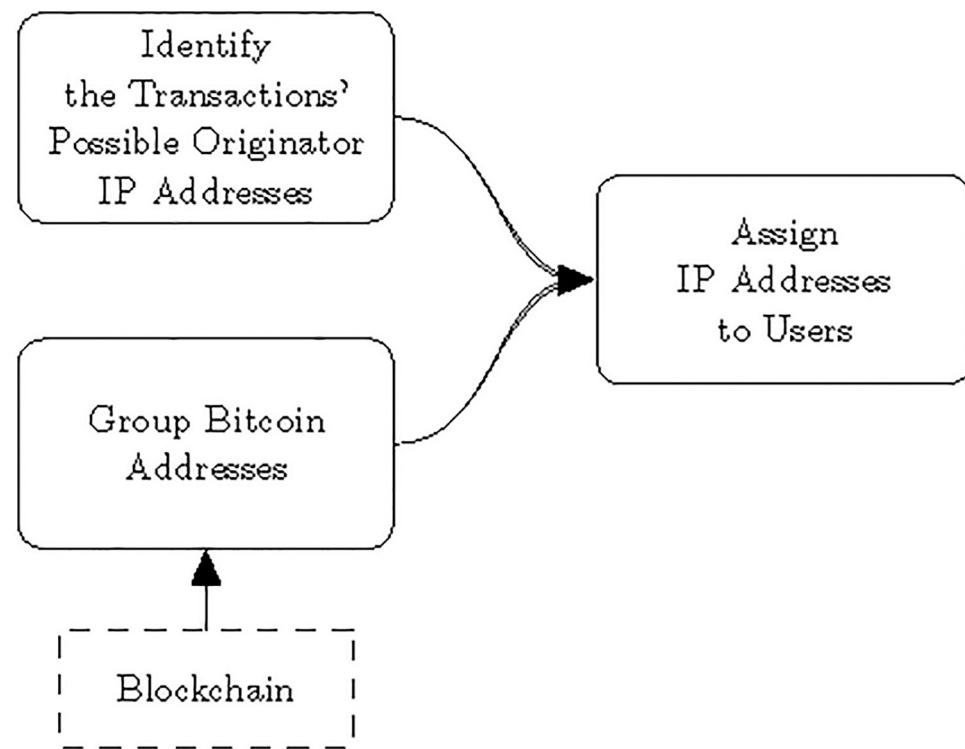


Fig 2. Main steps.

<https://doi.org/10.1371/journal.pone.0207000.g002>

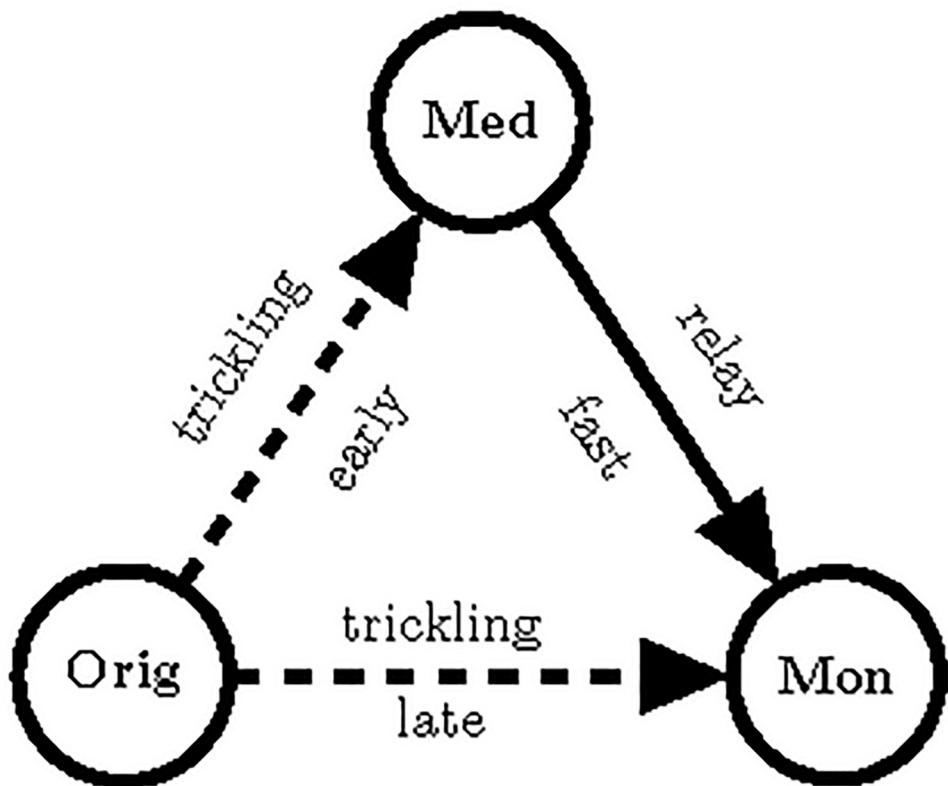


Fig 3. The message can be routed faster from the originator (Orig) to the monitoring client (Mon) through a mediator (Med) in the shown scenario.

<https://doi.org/10.1371/journal.pone.0207000.g003>

One iteration of sending messages happens every 100 ms time interval. Let us first calculate the probability that the originator relays the message to the monitoring client in one specific iteration. If the originator has c_{orig} clients connected to it (among which one is the monitoring client), then in every iteration there is $1/c_{\text{orig}}$ probability, that it relays to the monitoring client.

$$\mathbb{P}_1^{\text{orig}} = \frac{1}{c_{\text{orig}}} \quad (1)$$

In case of the mediator client, it relays the transaction to the monitor client with a probability of $1/c_{\text{med}}$ because of trickling, and it relays with a probability of $1/4$ if the other mechanism is used.

$$\mathbb{P}_1^{\text{med}} = \frac{1}{c_{\text{med}}} + \frac{1}{4} \left(1 - \frac{1}{c_{\text{med}}}\right) = \frac{1}{4} + \frac{3}{4c_{\text{med}}} \quad (2)$$

The probability that a specific client relays the transaction in the k -th iteration, follows a geometric distribution.

$$\begin{aligned} \mathbb{P}_k^{\text{orig}} &= \mathbb{P}_1^{\text{orig}} (1 - \mathbb{P}_1^{\text{orig}})^{k-1} \\ \mathbb{P}_k^{\text{med}} &= \mathbb{P}_1^{\text{med}} (1 - \mathbb{P}_1^{\text{med}})^{k-1} \end{aligned} \quad (3)$$

Let us consider the route on Fig 3 when the originator sends the message to the mediator, and then the mediator further relays it to the monitoring client. To calculate the distribution

of the iterations for the route through the mediator client, the sum of the two random variables has to be considered. This can be derived from the discrete convolution of the above two distributions.

$$\begin{aligned}
 \mathbb{P}_k^{\text{orig+med}} &= \sum_{i=1}^{k-1} (\mathbb{P}_i^{\text{orig}} \mathbb{P}_{k-i}^{\text{med}}) = \\
 &= \frac{\mathbb{P}_1^{\text{orig}} \mathbb{P}_1^{\text{med}} (1 - \mathbb{P}_1^{\text{med}})^{k-1}}{1 - \mathbb{P}_1^{\text{orig}}} \sum_{i=1}^{k-1} \left(\frac{1 - \mathbb{P}_1^{\text{orig}}}{1 - \mathbb{P}_1^{\text{med}}} \right)^i = \\
 &= \frac{\mathbb{P}_1^{\text{orig}} \mathbb{P}_1^{\text{med}}}{\mathbb{P}_1^{\text{med}} - \mathbb{P}_1^{\text{orig}}} \left[(1 - \mathbb{P}_1^{\text{orig}})^{k-1} - (1 - \mathbb{P}_1^{\text{med}})^{k-1} \right]
 \end{aligned} \tag{4}$$

As every ordinary client initiates 8 outgoing connections when connecting to the network, the number of connections is estimated to be 16 (taking into account the incoming connections as well).

If the two routes on [Fig 3](#) are considered independent from each other, the probability of the direct route being shorter (i.e. requires less iterations) is 0.5785. Here we have not taken into account the network delay, and that multiple indirect route can exist from the originator to the monitoring client possibly consisting of more steps. With this model however, we can approximate the probabilities. The goal is to determine a time frame that the monitoring client has to wait after receiving the message first until surely receiving it directly from the originator, if they are directly connected. If this waiting time is defined to be 2 sec, the above model gives a probability of 0.8841 for the direct route taking less iterations.

Since the time of our data collection, the trickling mechanism has been changed in the case of the Bitcoin Core client, so that relaying can be described by Poisson processes. In this case a similar calculation could be utilized but the waiting time will need to be adjusted to a value which maintains a high probability for the direct route. However, this does not change the further steps of our model except from the derivation of the above probabilities.

To successfully relay a transaction to another client, three messages have to be exchanged. First, the sender informs the receiver about the transactions it knows about (“INV” message). Then the receiver asks for the new, unknown transactions in the answer (“GETDATA” message). Finally, the actual information is sent to the receiver (“TX” message). As three messages need to be exchanged sequentially, the delay of the network plays an important role in the message propagation. The more mediator clients are involved in the transmission of the transaction, the longer the time it takes for the message to get to the monitoring client from the originator.

If we take into account the network delay, and that the above described “worst case scenario” (i.e. that we are connected to the originator, and an indirect route consisting of one mediator exists) is unlikely, we can *neglect* the probability that a message is received from an indirect route earlier than two seconds before receiving it from the originator. This assumption is experimentally verified in [\[8\]](#).

We call this time interval the *first time segment* of the transaction and denote it by $t_1 = 2 \text{ sec}$. If the monitoring client is not connected directly to the originator, it will only receive the transaction via possibly multiple indirect routes. Nevertheless, it will be true with high probability that connected clients that do not belong to this first time segment are not the originators of the transaction. We then proceed with this assumption to estimate the probabilities of a client being the originator of the transaction based on each received transaction.

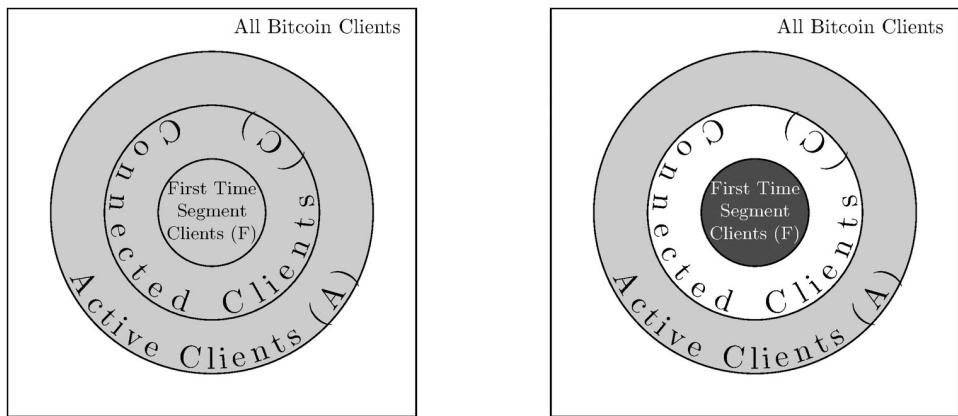


Fig 4. Relation of the different sets of clients. The darker the subset is, the higher the probability of a client in that subset is the originator of the transaction.

<https://doi.org/10.1371/journal.pone.0207000.g004>

As of today, this mechanism has changed, so that the relay can be described by Poisson processes. In this case a similar calculation can be utilized, except from the fact that the probability of not belonging to the first time segment can not be neglected.

From the perspective of a monitoring client, the other Bitcoin clients can be classified to sets based on each transaction according to Fig 4. Some of the Bitcoin clients relay the message to it in the first time segment. This constitutes a subset of the Bitcoin clients to which the monitoring client is connected to at the time of the transaction. Only active Bitcoin clients are connected to the network, but not all of the clients are working at the examined moment.

Before the transaction, no information is known, thus the best estimate we can make is that each Bitcoin client has equal probability of being the originator of the transaction, resulting in a uniform probability distribution among the active clients (left side of Fig 4). After the transaction, each Bitcoin client in the first time segment can be either the real originator of the transaction or a client relaying it (via several hops). Furthermore, the real originator can also be among the rest of the network, not connected to our monitoring client. On the other hand, based on the previous arguments, we presume that clients not relaying the transaction in the first time segment are certainly not the originators of the transaction. Thus, the probability of the first time segment clients increases while the connected clients not belonging to the first time segment will have zero probability (right side of Fig 4). Still nothing is known about the clients not connected to the monitoring client, therefore their probabilities will not change. Also, clients belonging to the same subsets can not be distinguished.

Let us calculate the probabilities of being the originator for clients in each set. The Roman font type notations of Fig 4 are used for the sets. The number of elements in the sets is denoted by $|\cdot|$. \mathcal{C} denotes that the monitoring client is connected to the originator of the transaction, \mathcal{O} denotes that the originator relays the message in the first time segment to the monitoring client and \mathcal{F} means that a randomly chosen client from the first time segment is actually the originator of the transaction. Using these notations, we have that

$$\mathbb{P}(\mathcal{C}) = \frac{|\mathcal{C}|}{|\mathcal{A}|} \quad (5)$$

as inactive clients can not be the originator of the transaction. If the monitoring client is connected to the originator, it is going to inform the monitoring client in the first time segment. At this time all of the first time segment clients have the same probability of being the

originator.

$$\mathbb{P}(\mathcal{O}|\mathcal{C}) = 1 \quad \mathbb{P}(\mathcal{F}|\mathcal{C}) = \frac{1}{|F|} \quad (6)$$

Let us apply the law of total probability for $\mathbb{P}(\mathcal{F})$.

$$\mathbb{P}(\mathcal{F}) = \mathbb{P}(\mathcal{F}|\mathcal{C}) \cdot \mathbb{P}(\mathcal{C}) + \mathbb{P}(\mathcal{F}|\overline{\mathcal{C}}) \cdot \mathbb{P}(\overline{\mathcal{C}}) = \frac{|C|}{|A| |F|} \quad (7)$$

where we exploited that a client can not send any messages in the first time segment if it is not at all connected to the monitoring client: $\mathbb{P}(\mathcal{F}|\overline{\mathcal{C}}) = 0$.

The above formula gives the probability assigned to the first time segment clients. The connected clients not belonging to the first time segment have zero probability. The rest of the active clients has the same $1/|A|$ probability. We note that these probabilities still sum up to 1: probabilities among the connected clients were “redistributed” according to whether they belong to the first time segment.

So far we only considered one monitoring client. If there are more monitoring clients the above mentioned sets are defined separately for each of them, and then the union of the corresponding sets is determined, i.e. $F(tx_i) = \bigcup_{j=1}^N F_j(tx_i)$ and $C(tx_i) = \bigcup_{j=1}^N C_j(tx_i)$ for N monitoring clients, where the subscripts denote the corresponding sets as observed for transaction tx_i by the j th monitoring client. Using this method, monitoring Bitcoin clients do not need to be synchronized in time. If time synchronization among monitoring clients was achieved, we could further limit the F set of first time segment clients to those that broadcast the transaction in t_1 time after *any* of our monitoring clients first received the transaction. In our experiments, achieving reliable time synchronization was not possible, so the union of sets was used as described. We note that the set of active clients at a given time (A) is not straightforward to estimate even with a large number of monitoring clients. To do that, we would need to perform an active network discovery over the peer-to-peer network of Bitcoin clients. Instead of implementing this functionality ourselves, we relied on the Bitnodes.io database [20], which provides the estimated number of active Bitcoin clients as a function of time (i.e. $|A|$). The actual set is not required for the calculations, only the size of the set at the time of the transactions is considered.

Step 2: Grouping the transactions belonging to the same user

The next task is to group the Bitcoin addresses according to the users they are owned by. After this, every transaction can be assigned to the users by looking at the source Bitcoin addresses of the transaction.

To group addresses, we exploit that Bitcoin addresses appearing on the input side of the same transaction typically belong to the same user. This assumption is employed widely in the literature as well [5, 6, 9, 21].

This can be used for grouping individual Bitcoin addresses. The process is demonstrated in Fig 5. The left side of the figure shows the transactions and the input Bitcoin addresses where the Bitcoins are sent from. These Bitcoin addresses belong to the same user. When a Bitcoin address appears in different transactions (marked red and bold), all Bitcoin addresses can be merged and assigned to the same user. Although Bitcoin users are encouraged to generate new Bitcoin addresses after every transaction they make, so that the above grouping is less efficient [22], most of the users do not follow this guideline [23, 24].

The transactions belong to the user that owns its input Bitcoin address(es).

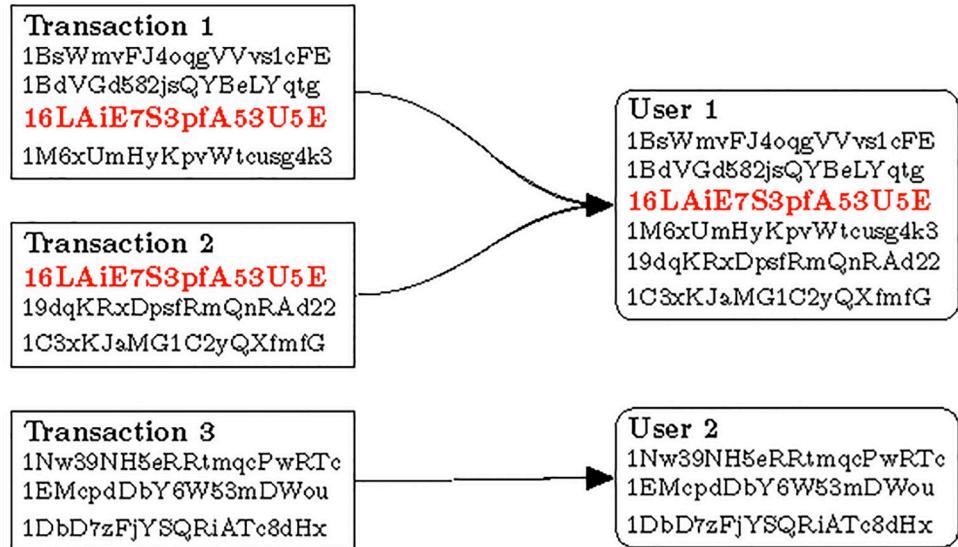


Fig 5. Grouping of Bitcoin addresses: The left side shows three transactions and the input Bitcoin addresses of these transactions, while the right side indicates how these Bitcoin addresses are grouped.

<https://doi.org/10.1371/journal.pone.0207000.g005>

Step 3: Combining probabilities—Naive Bayes classification

From the message propagation it can be determined how likely the clients are the originators of the transactions. So far we considered the transactions independently from each other.

According to our assumptions, the transactions belonging to a single user were created by a few originator clients. This means that these transactions provide probabilities for the same set of originator clients. The originator clients can be identified more efficiently by combining the probabilities belonging to these transactions, thus obtaining a more decisive result. This can be calculated by the naive Bayes classifier method [25]. Table 1 shows the transactions (denoted by tx) created by a single user. The transactions assign probabilities to the clients (IP addresses), which indicate the likelihood that the client is the originator of the transaction.

If the ratio of the connected clients is small, the individual probabilities in the table are also low. The probabilities of an IP address related to the different transactions can be combined by the naive Bayes classification, resulting in a row of combined probabilities. This shows how likely the IP addresses belong to the examined user.

The IP addresses will be divided into two classes, to the “originator” and the “non-originator” classes. For each transaction, there can be at most one IP address in the originator class. On the other hand, as a user can use multiple IP addresses to create Bitcoin transactions, after

Table 1. The transactions of a single user (tx) assign probabilities to the clients (IP addresses), which shows the likelihood that the client is the originator of the transaction. $\mathbb{P}(IP_i|tx_j)$ denotes the probability that IP_i address created the tx_k transaction.

	IP_1	...	IP_i	...	IP_n
tx_1	$\mathbb{P}(IP_1 tx_1)$		$\mathbb{P}(IP_i tx_1)$		$\mathbb{P}(IP_n tx_1)$
tx_2	$\mathbb{P}(IP_1 tx_2)$		$\mathbb{P}(IP_i tx_2)$		$\mathbb{P}(IP_n tx_2)$
...					
tx_j	$\mathbb{P}(IP_1 tx_j)$		$\mathbb{P}(IP_i tx_j)$		$\mathbb{P}(IP_n tx_j)$
...					
tx_m	$\mathbb{P}(IP_1 tx_m)$		$\mathbb{P}(IP_i tx_m)$		$\mathbb{P}(IP_n tx_m)$

<https://doi.org/10.1371/journal.pone.0207000.t001>

combining multiple transactions, more than one IP address can be in the originator class in the final result.

It is assumed that the Bitcoin users can be identified by a limited number of IP addresses they use when connected to the Bitcoin network. This involves that the users do not use TOR (“The Onion Router”), proxy servers or other similar systems hiding their IP addresses. If this does not hold, i.e. the users use TOR, the probabilities would be distributed among several IP addresses thus resulting in small final probabilities. We note, that the invalidity of this assumption for some users does not result in false IP address—user pairings: only those users will be identified whom the assumption holds for. Furthermore, previous work showed that the usage of the TOR network can be prevented by an active malicious attacker by connecting to the TOR network as well and sending malformed Bitcoin messages via the TOR exit nodes [8, 26]. This kind of attack would result in users being unable to connect to the Bitcoin network via TOR. In the current work however, we limit our analysis to regular users, i.e. who connect to the Bitcoin network using only a few IP addresses.

By the application of the naive Bayes classifier (see Appendix 7 for the detailed derivation), the combined probability of an IP address (IP_i) belonging to the C_o originator class is given by

$$\mathbb{P}(IP_i \in C_o | \mathbf{tx}) = \frac{1}{1 + \exp \left[(1 - m) \ln \left(\overline{|A|} - 1 \right) + \sum_{k=1}^m \ln \left(\frac{1}{\mathbb{P}(IP_i \in C_o | \mathbf{tx}_k)} - 1 \right) \right]} \quad (8)$$

where \mathbf{tx} denotes the vector of all considered transactions, $\overline{|A|}$ is the average of the total number of active clients through the transactions and m is the number of transactions. The $|A|$ number of active clients varies through the transactions as they occur in different times. Thus, the $\overline{|A|}$ average of the different $|A|$ values is used as it is suggested in [27].

We note that the naive Bayes classification can only be applied if the transactions provide conditionally independent probabilities. Otherwise the dependencies between the transactions should be determined [28].

5 Data collection

During the data collection campaign, we used our modified Bitcoin clients to connect to the network and monitor information about transactions relayed by connected clients. As the program code is open-source, it was straightforward to implement a monitoring client.

Our monitoring clients logged the incoming “INV” messages along with the IP address of the sender client and the time of reception. These messages contain the 128-bit hash code of the transactions which are relayed.

Using this hash code, the Bitcoin addresses, the amount of Bitcoin sent and other information of interest can then be looked up in the blockchain.

In order to monitor as large part of the Bitcoin network as possible, the modified Bitcoin clients were installed simultaneously to 140 computers located at different parts of the world, and all of these were recording the observed traffic during the campaign. Bitcoin clients behind firewalls usually do not allow incoming connections, i.e. our monitoring clients can not establish connections to them. By using a large number of monitoring clients, it is more likely that Bitcoin clients behind firewalls initiate connections to some of our monitoring clients when they enter the network. We installed the monitoring clients on computers that are part of PlanetLab, a system maintained for network communication research. [29]

The data collection campaign took slightly more than two months between 10/14/2013 and 12/20/2013. During this period 300 million records were obtained, in which 4155387

transactions and 124498 IP-addresses were identified. The collected data was imported into an SQL database server.

To calculate the probabilities described above, the total number of active clients need to be determined. From the Bitnodes.io database [20] one can look up the number of active IP addresses of the Bitcoin clients as a function of time.

Ethics statement

All data used in the analysis is made publicly available by the Bitcoin users as it is required by the Bitcoin protocol. Collecting data on the level of network traffic possibly allows linking Bitcoin addresses to the IP addresses of Bitcoin users. No other personally identifiable information beside IP address was collected about users, and no attempt was made to link IP addresses to actual people beside establishing coarse-grained geographic location. In the shared data, IP addresses were replaced with random identifiers to prevent connecting the transactions with individuals based on other IP address related information.

6 Results

When calculating the combined probability of each IP address belonging to the specific user, the question arises when should a pairing be *accepted*? As more than one IP address can be used by each user and one IP address can be used by several users, no restriction is made of this kind. A pairing is accepted, if its probability is higher than 0.5. This means that the IP address of interest has at least 0.5 probability of being used by the user. [Fig 6](#) shows the distribution of the probabilities of the accepted pairings. It can be seen, that the vast majority of the probabilities are above 0.9, the average value of the pairings is 95.52%, so we expect the false positive rate of the results to be low. Two peaks can be observed on the figure, one with a maximum at 0.952, and another close to 1. The first peak is due to usual clients that initiate a relatively small number of transactions. We speculate that the other peak consists of servers offering wallet services, i.e. servers that can be used by several people thus initiating a lot of transactions (see below in more detail). The more initiated transactions can be taken into account, the higher the probability will be that can be assigned to the pairings.

As a result, 22363 users could be identified, and altogether 1797 IP addresses were assigned to them.

The imbalance is caused by three outstanding IP addresses to which 20680 users are assigned. These IP addresses probably belong to Bitcoin wallet services, which can be used for creating transactions on a website without using a private computer. Note that the incomplete grouping of Bitcoin addresses can also result in an IP address being associated with several groups of Bitcoin addresses. These groups actually belong to the same user, but they could not be connected in the grouping algorithm.

For the remainder data, 1.14 IP addresses belong to one user on average. This is due to the fact that a user can use multiple IP addresses when connecting to the Bitcoin network. The maximum number of IP addresses identified as belonging to a single user is 8.

Calculating the balances

Examining the blockchain data alone allows to investigate the time evolution of user balances before, during and after the data collection campaign.

[Fig 7](#) shows the total balance of all identified users versus time. The time interval in which the data collection was taking place is marked by a shaded area. Before data collection, the amount of Bitcoin owned by the identified users is increasing. This is due to the fact that some of the identified Bitcoin addresses were created before the beginning of the

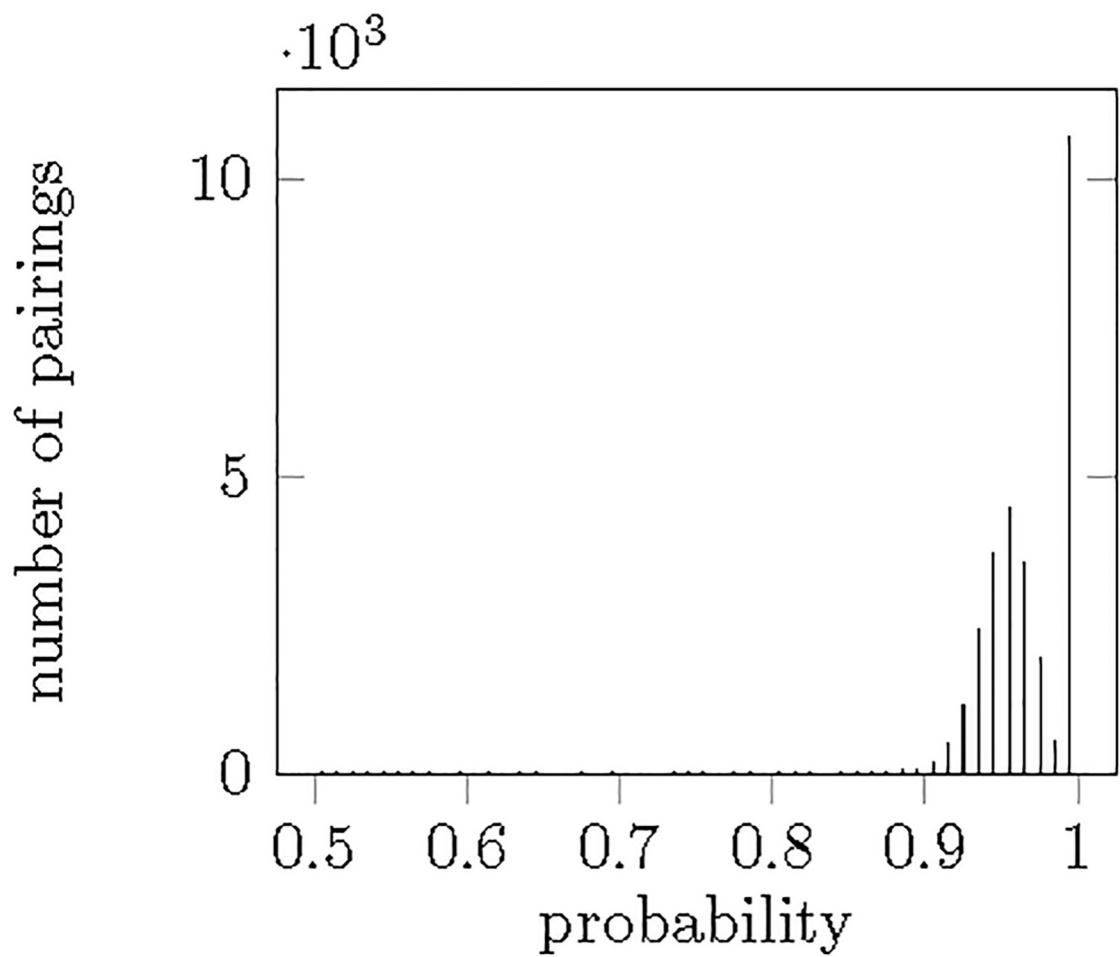


Fig 6. Distribution of the probabilities assigned to the accepted user—IP address pairings.

<https://doi.org/10.1371/journal.pone.0207000.g006>

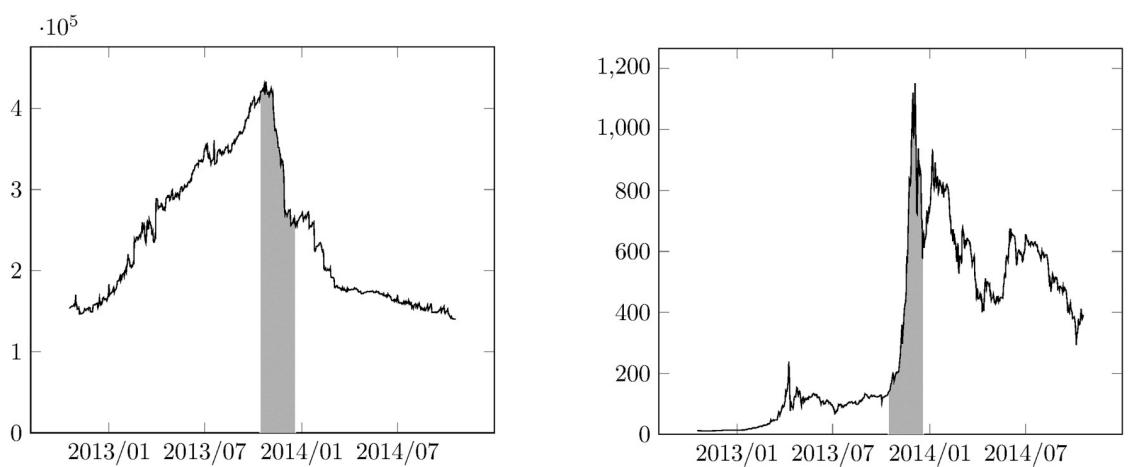


Fig 7. Balances of Bitcoin users identified in our study and the Bitcoin exchange rate. The shaded area corresponds to our data collection period.

<https://doi.org/10.1371/journal.pone.0207000.g007>

measurement campaign. After the measurement, some of the identified Bitcoin addresses were not used anymore, and other new unidentified Bitcoin addresses took over their place. The steep drop during the measurement is probably due to the significant increase of exchange rate in this time interval, which probably inspired the users to sell their Bitcoins for traditional currencies. We found a significant, -0.91 linear correlation coefficient between the total amount of Bitcoin owned by the identified users and the exchange rate during the measurement period.

The total number of Bitcoins in use is constantly increasing as time goes by. At the time of the measurement ~ 13500000 Bitcoins were in circulation. The amount of Bitcoins owned by the identified users reached a maximum of 432666 on 10/25/2013, which corresponds to $\sim 3.2\%$ of the total amount of Bitcoins. We believe that this ratio is a statistically representative sample, if the data were collected with uniform random sampling. However, systematic differences could have affected the data collection as users in different parts of the world, with different intentions and technical backgrounds were possibly operating differently in the network. The users could be protected by firewalls, thus banning incoming connections, and they could also obscure their operation by using VPN, proxy service or TOR.

Geographical distribution of Bitcoin and the cash flow

The location of IP addresses can be determined from publicly available databases such as MaxMind [30], which contains approximate locations of the IP addresses. If the Bitcoin users use additional tools to hide their IP addresses, or if the IP addresses are located at other positions than they are registered to, the database gives false location results. However, these inaccuracies are not relevant in the vast majority of the cases.

Fig 8 shows the distribution of the identified Bitcoin clients. The coloring represents the logarithmic value of the density. The identified Bitcoin clients are mostly located on the more developed regions of the world. Note that in some countries, such as Russia or China, the Internet is regulated, therefore some interference of the connected clients (and their messages) can occur.

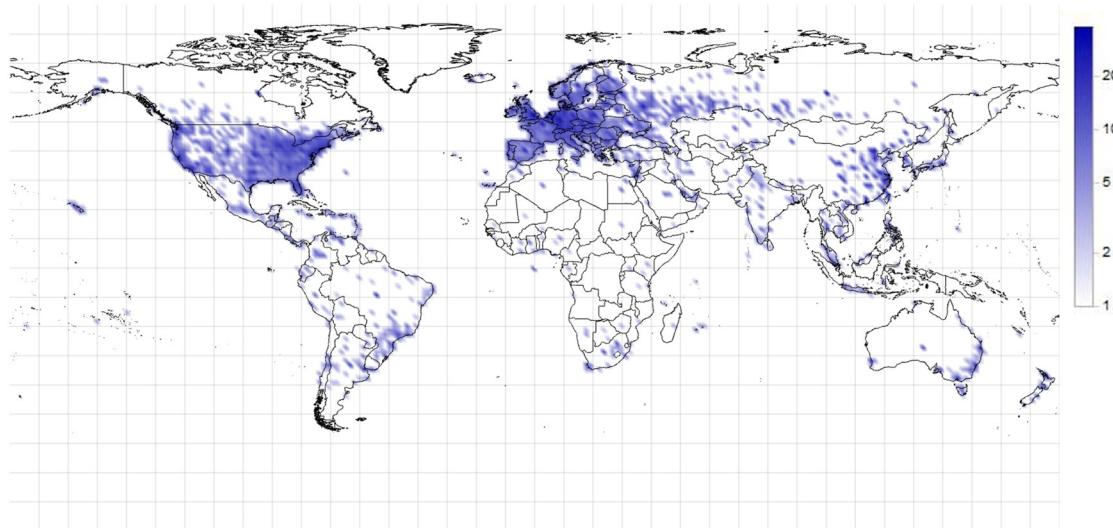


Fig 8. Distribution of the identified Bitcoin clients (1/100 km²).

<https://doi.org/10.1371/journal.pone.0207000.g008>

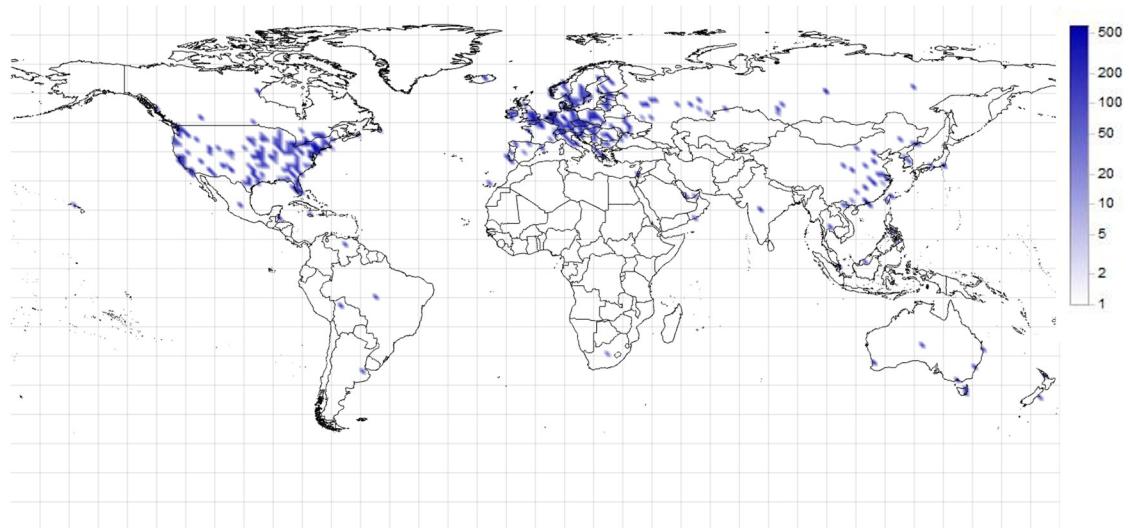


Fig 9. Distribution of Bitcoin owned by the identified clients on 12/20/2013 (1/100 km²).

<https://doi.org/10.1371/journal.pone.0207000.g009>

By the localization of the IP addresses, the geographical distribution of Bitcoin can also be determined (Fig 9). This figure only shows the distribution of the Bitcoins that are owned by the identified clients; the coloring is logarithmic. The snapshot belongs to the end of the data collection period, 12/20/2013.

The analysis detailed in Section 4 results in a data set of transactions and identified originators. It is worth to examine if some originator addresses can be mapped to receiver Bitcoin addresses as well. There are 68973 transactions in which both sides could be found, and altogether 196971 Bitcoins were transferred in the identified transactions. In these transactions 7372 users appear as senders and 6170 appear as receivers.

The transactions are visualized on a world map (Fig 10). The thickness, opacity and saturation of the arrows express the amount of Bitcoin transferred in the related transaction. The

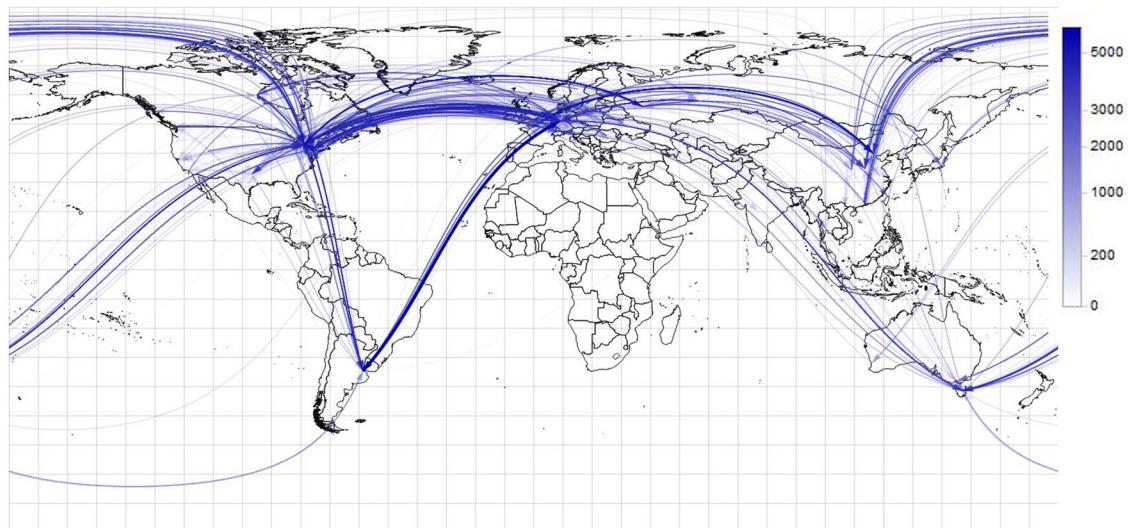


Fig 10. The flow of Bitcoin.

<https://doi.org/10.1371/journal.pone.0207000.g010>

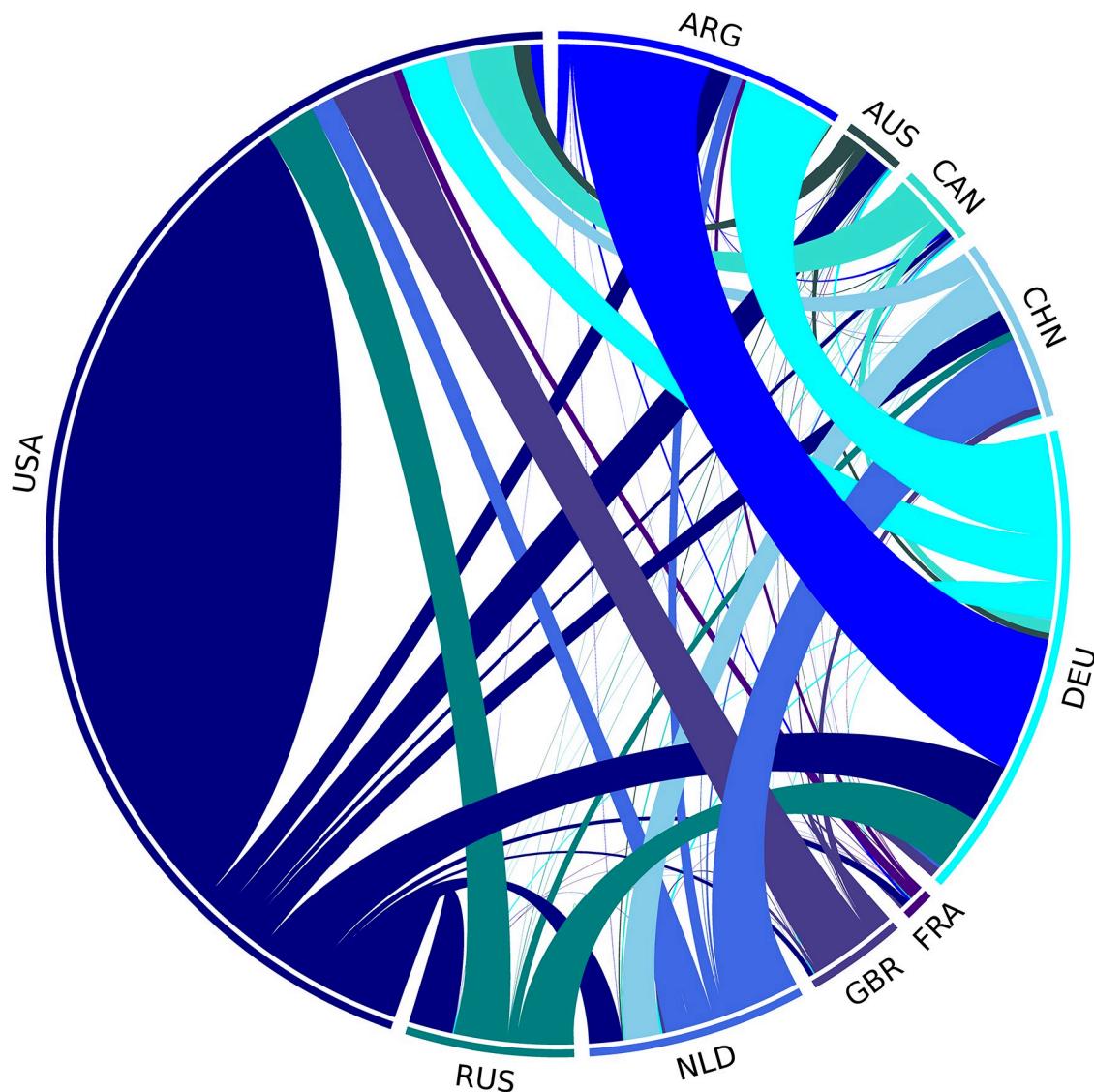


Fig 11. The flow of Bitcoin between countries.

<https://doi.org/10.1371/journal.pone.0207000.g011>

time course of the transactions is demonstrated on a video that can be found at <https://figshare.com>.

Let us have a look at the flow of Bitcoin between the different countries, which is illustrated in Fig 11. As the vast majority of the identified Bitcoin transactions belongs to a few countries, only the top ten most significant ones are shown in the figure. 87.5% of the Bitcoins in our data set were transferred between these countries.

The different countries are indicated by arcs on the perimeter of the figure. The colors of the links are identical with the color of the country where the Bitcoins were sent from. A high amount of the Bitcoins (24250 Bitcoins, more than 12.3% of the total amount) are transacted internally in the United States. There are several interesting connections: the second largest flow is between Germany and Argentina (25508 Bitcoins, 13.0% of the total amount), and there is a significant Bitcoin flow between China and the Netherlands as well.

7 Conclusions

In this paper we examined the problem of user identification in the Bitcoin network. While Bitcoin provides a significant level of anonymity as Bitcoin addresses can be generated freely and without providing any form of personal identification, the requirement to announce new transactions on the peer to peer network opens up the possibility of linking Bitcoin addresses to the IP addresses of clients. Our main goal was evaluating the feasibility of this procedure.

We installed a modified Bitcoin client program on over a hundred computers, which recorded the propagating messages on the network that announced new transactions. Based on the information propagation properties of these messages, we developed a mathematical model using naive Bayes classifier method to assign Bitcoin addresses to the clients that most probably control them. As a result, Bitcoin address—IP address mappings were identified. Through the IP addresses of the clients, we could determine their geographical location, which enabled the spatial analysis of distribution and flow of Bitcoin.

The method is cheap in terms of resources, the used algorithms are relatively easy to implement and can be combined with other Bitcoin-transaction related information.

All monitoring clients behaved as regular Bitcoin clients during the measurement. Although they did not generate any transactions, the source code can be modified to do so if a better concealment is required. Furthermore, the monitoring clients do not need to be connected to other Bitcoin users in any detectable way (i.e. communication among them is trivially achieved outside the Bitcoin protocol), making it virtually impossible to reveal their monitoring activity. This raises the question if the Bitcoin network might already be monitored by a similar methodology. It can be implied that Bitcoin users should take further steps to adequately disguise their real IP addresses and preserve their anonymity.

Appendix—Derivation of naive Bayes classifier method

The model classifies the clients into the *originator* and *non-originator* classes (C_o and C_n respectively) based on their IP addresses and by considering m transactions. Transactions are denoted by $\mathbf{tx} = \{\mathbf{tx}_i\}$, ($i \in [1; m]$).

Consider a single IP address, and let us examine the probabilities that the different transactions assign to it. Using the Bayes theorem the probability of belonging to the originator class is

$$\mathbb{P}(\text{IP}_i \in C_o | \mathbf{tx}) = \frac{\mathbb{P}(\text{IP}_i \in C_o)}{\mathbb{P}(\mathbf{tx})} \mathbb{P}(\mathbf{tx} | \text{IP}_i \in C_o) \quad (9)$$

where $\mathbb{P}(\text{IP}_i \in C_o)$ is the frequency of C_o class (a priori probability). By assuming that the probabilities $\mathbb{P}(\mathbf{tx} | \text{IP}_i \in C_o)$ are conditionally independent, the expression can be simplified.

$$\mathbb{P}(\text{IP}_i \in C_o | \mathbf{tx}) = \frac{\mathbb{P}(\text{IP}_i \in C_o)}{\mathbb{P}(\mathbf{tx})} \prod_{i=1}^m \mathbb{P}(\mathbf{tx}_i | \text{IP}_i \in C_o) \quad (10)$$

Bayes theorem can be applied again to the factors in the product.

$$\mathbb{P}(\text{IP}_i \in C_o | \mathbf{tx}) = \frac{\mathbb{P}(\text{IP}_i \in C_o)}{\mathbb{P}(\mathbf{tx})} \prod_{i=1}^m \frac{\mathbb{P}(\text{IP}_i \in C_o | \mathbf{tx}_i) \mathbb{P}(\mathbf{tx}_i)}{\mathbb{P}(\text{IP}_i \in C_o)} = \underbrace{\frac{\prod_{i=1}^m \mathbb{P}(\mathbf{tx}_i)}{\mathbb{P}(\mathbf{tx})}}_{\text{const}} \cdot \frac{\prod_{i=1}^m \mathbb{P}(\text{IP}_i \in C_o | \mathbf{tx}_i)}{\prod_{i=1}^{m-1} \mathbb{P}(\text{IP}_i \in C_o)^{m-1}} \quad (11)$$

The first factor is constant (depends only on the data), and it can be eliminated by the

normalization of the probabilities. As $\mathbb{P}(\text{IP}_i \in C_o | \text{tx}) + \mathbb{P}(\text{IP}_i \in C_n | \text{tx}) = 1$ is valid,

$$\begin{aligned} \mathbb{P}(\text{IP}_i \in C_o | \text{tx}) &= \frac{1}{\underbrace{\frac{\prod_{i=1}^m \mathbb{P}(\text{IP}_i \in C_o | \text{tx}_i)}{\mathbb{P}(\text{IP}_i \in C_o)^{m-1}} + \frac{\prod_{i=1}^m \mathbb{P}(\text{IP}_i \in C_n | \text{tx}_i)}{\mathbb{P}(\text{IP}_i \in C_n)^{m-1}}}_{\text{const}}} \cdot \frac{\prod_{i=1}^m \mathbb{P}(\text{IP}_i \in C_o | \text{tx}_i)}{\mathbb{P}(\text{IP}_i \in C_o)^{m-1}} = \end{aligned} \quad (12)$$

$$= \frac{1}{\frac{\prod_{i=1}^m \mathbb{P}(\text{IP}_i \in C_o | \text{tx}_i)}{\mathbb{P}(\text{IP}_i \in C_o)^{m-1}} + \frac{\prod_{i=1}^m (1 - \mathbb{P}(\text{IP}_i \in C_o | \text{tx}_i))}{(1 - \mathbb{P}(\text{IP}_i \in C_o))^{m-1}}} \cdot \frac{\prod_{i=1}^m \mathbb{P}(\text{IP}_i \in C_o | \text{tx}_i)}{\mathbb{P}(\text{IP}_i \in C_o)^{m-1}}$$

The expression can be simplified further.

$$\begin{aligned} \mathbb{P}(\text{IP}_i \in C_o | \text{tx}) &= \frac{\frac{\prod_{k=1}^m \mathbb{P}(\text{IP}_i \in C_o | \text{tx}_k)}{\mathbb{P}(\text{IP}_i \in C_o)^{m-1}}}{\frac{\prod_{k=1}^m \mathbb{P}(\text{IP}_i \in C_o | \text{tx}_k)}{\mathbb{P}(\text{IP}_i \in C_o)^{m-1}} + \frac{\prod_{k=1}^m (1 - \mathbb{P}(\text{IP}_i \in C_o | \text{tx}_k))}{(1 - \mathbb{P}(\text{IP}_i \in C_o))^{m-1}}} = \\ &= \frac{1}{1 + \frac{\frac{\mathbb{P}(\text{IP}_i \in C_o)^{m-1}}{(1 - \mathbb{P}(\text{IP}_i \in C_o))^{m-1}} \cdot \frac{\prod_{k=1}^m (1 - \mathbb{P}(\text{IP}_i \in C_o | \text{tx}_k))}{\prod_{k=1}^m \mathbb{P}(\text{IP}_i \in C_o | \text{tx}_k)}}{1 + \frac{\mathbb{P}(\text{IP}_i \in C_o)^{m-1}}{(1 - \mathbb{P}(\text{IP}_i \in C_o))^{m-1}} \cdot \frac{\prod_{k=1}^m (1 - \mathbb{P}(\text{IP}_i \in C_o | \text{tx}_k))}{\prod_{k=1}^m \mathbb{P}(\text{IP}_i \in C_o | \text{tx}_k)}}} \quad (13) \end{aligned}$$

$\mathbb{P}(\text{IP}_i \in C_o)$ is the initial frequency of occurrence of the clients in the C_o class, which is $1/|A|$. Although a Bitcoin client can use multiple IP addresses in the network, it is assumed that the $1/|A|$ value is a good approximation for the initial frequency in the vast majority of the cases. The total number of active clients varies with time in the scale of all considered transactions. Thus, the $\overline{|A|}$ average of the different $|A|$ values is used as suggested in [27].

$$\begin{aligned} \mathbb{P}(\text{IP}_i \in C_o | \text{tx}) &= \frac{1}{1 + \left(\frac{\frac{1}{\overline{|A|}}}{1 - \frac{1}{\overline{|A|}}} \right)^{m-1} \cdot \prod_{k=1}^m \left(\frac{1}{\mathbb{P}(\text{IP}_i \in C_o | \text{tx}_k)} - 1 \right)} = \\ &= \frac{1}{1 + (\overline{|A|} - 1)^{1-m} \cdot \prod_{k=1}^m \left(\frac{1}{\mathbb{P}(\text{IP}_i \in C_o | \text{tx}_k)} - 1 \right)}, \end{aligned} \quad (14)$$

This formula brings in a technical problem. Huge numbers are multiplied together in the product, which become significantly biased in regular number representations by rounding and may result in overflow. To relax this problem, the second term of the denominator is written in an exponential form.

$$\begin{aligned} \mathbb{P}(\text{IP}_i \in C_o | \text{tx}) &= \frac{1}{1 + e^\xi}, \\ \xi &= (1 - m) \ln \left(\overline{|A|} - 1 \right) + \sum_{k=1}^m \ln \left(\frac{1}{\mathbb{P}(\text{IP}_i \in C_o | \text{tx}_k)} - 1 \right). \end{aligned} \quad (15)$$

This results in the following practical formula.

$$\mathbb{P}(\text{IP}_i \in C_o | \text{tx}) = \frac{1}{1 + \exp \left[(1 - m) \ln \left(\frac{1}{|A|} - 1 \right) + \sum_{k=1}^m \ln \left(\frac{1}{\mathbb{P}(\text{IP}_i \in C_o | \text{tx}_k)} - 1 \right) \right]} \quad (16)$$

This formula enables us to combine the probabilities assigned to the IP addresses by the transactions.

Acknowledgments

This work has been accomplished at the Department of Physics of Complex Systems, Eötvös Loránd University. Authors thank the Hungarian National Research, Development and Innovation Office under Grant No. KH125280 and the grant of Ericsson Inc.

We thank the PlanetLab community [29] that the servers of their network could be used for the data collection.

We are also grateful to MaxMind [30] for the database that could be used for the localization of the IP addresses.

Author Contributions

Conceptualization: József Stéger, Dániel Kondor, Gábor Vattay.

Data curation: Péter L. Juhász, Dániel Kondor.

Formal analysis: Péter L. Juhász.

Investigation: Péter L. Juhász, József Stéger, Dániel Kondor.

Software: Péter L. Juhász, József Stéger.

Visualization: Péter L. Juhász.

Writing – original draft: Péter L. Juhász, József Stéger, Dániel Kondor.

References

1. Nakamoto S. Bitcoin: A peer-to-peer electronic cash system,” <http://bitcoin.org/bitcoin.pdf>; 2009.
2. Bitcoin Core, GitHub repository; 2009. <https://github.com/bitcoin/bitcoin>.
3. Bitcoin Wiki; 2010. <https://en.bitcoin.it/wiki/Anonymity>.
4. Christin N. Traveling the Silk Road: A Measurement Analysis of a Large Anonymous Online Marketplace. In: Proceedings of the 22Nd International Conference on World Wide Web. WWW’13. New York, NY, USA: ACM; 2013. p. 213–224. Available from: <http://doi.acm.org/10.1145/2488388.2488408>.
5. Androulaki E, Karame GO, Roeschlin M, Scherer T, Capkun S. Evaluating User Privacy in Bitcoin. In: Sadeghi AR, editor. Financial Cryptography and Data Security. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013. p. 34–51.
6. Reid F, Harrigan M. In: Altshuler Y, Elovici Y, Cremers AB, Aharony N, Pentland A, editors. An Analysis of Anonymity in the Bitcoin System. New York, NY: Springer New York; 2013. p. 197–223. Available from: https://doi.org/10.1007/978-1-4614-4139-7_10.
7. Gross R, Acquisti A. Information Revelation and Privacy in Online Social Networks. In: Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society. WPES’05. New York, NY, USA: ACM; 2005. p. 71–80. Available from: <http://doi.acm.org/10.1145/1102199.1102214>.
8. Biryukov A, Khovratovich D, Pustogarov I. Deanonymisation of Clients in Bitcoin P2P Network. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. CCS’14. New York, NY, USA: ACM; 2014. p. 15–29. Available from: <http://doi.acm.org/10.1145/2660267.2660379>.

9. Koshy P, Koshy D, McDaniel P. An Analysis of Anonymity in Bitcoin Using P2P Network Traffic. In: Christin N, Safavi-Naini R, editors. *Financial Cryptography and Data Security*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2014. p. 469–485.
10. Bojja Venkatakrishnan S, Fanti G, Viswanath P. Dandelion. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*. 2017; 1(1):1–34. <https://doi.org/10.1145/3084459>
11. Decker C, Wattenhofer R. Information propagation in the Bitcoin network. In: *IEEE P2P 2013 Proceedings*; 2013. p. 1–10.
12. Donet Donet JA, Pérez-Solà C, Herrera-Joancomartí J. The Bitcoin P2P Network. In: Böhme R, Brenner M, Moore T, Smith M, editors. *Financial Cryptography and Data Security*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2014. p. 87–102.
13. Neudecker T, Andelfinger P, Hartenstein H. Timing Analysis for Inferring the Topology of the Bitcoin Peer-to-Peer Network. In: *2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/Smart-World)*; 2016. p. 358–367.
14. Goldfeder S, Kalodner HA, Reisman D, Narayanan A. When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies. *CoRR*. 2017;abs/1708.04748.
15. Miller A, Möser M, Lee K, Narayanan A. An Empirical Analysis of Linkability in the Monero Blockchain. *CoRR*. 2017;abs/1704.04299.
16. Conti M, E SK, Lal C, Ruj S. A Survey on Security and Privacy Issues of Bitcoin. *IEEE Communications Surveys Tutorials*. 2018; p. 1–1.
17. Wang L, Pustogarov I. Towards Better Understanding of Bitcoin Unreachable Peers. *CoRR*. 2017;abs/1709.06837.
18. Pel A. *Money for Nothing and Bits for Free: The Geographies of Bitcoin*; 2015.
19. Brown K, Boileau M, Subramaniam G. Bitcoin and Ethereum: Empirical Evidence on Node Distribution; 2018.
20. Bitnodes.io; <https://getaddr.bitnodes.io>.
21. Ron D, Shamir A. Quantitative Analysis of the Full Bitcoin Transaction Graph. In: Sadeghi AR, editor. *Financial Cryptography and Data Security*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013. p. 6–24.
22. Bonneau J, Narayanan A, Miller A, Clark J, Kroll JA, Felten EW. Mixcoin: Anonymity for Bitcoin with Accountable Mixes. In: Christin N, Safavi-Naini R, editors. *Financial Cryptography and Data Security*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2014. p. 486–504.
23. Koshy P. *Coinseer: A Telescope Into Bitcoin*. Pennsylvania State University; 2013. Available from: <https://books.google.hu/books?id=DkyAnQAACAAJ>.
24. Meiklejohn S, Pomarole M, Jordan G, Levchenko K, McCoy D, Voelker GM, et al. A Fistful of Bitcoins: Characterizing Payments Among Men with No Names. *Commun ACM*. 2016; 59(4):86–93. <https://doi.org/10.1145/2896384>
25. Lewis DD. Naive (Bayes) at forty: The independence assumption in information retrieval. In: Nédellec C, Rouveirol C, editors. *Machine Learning: ECML-98*. Berlin, Heidelberg: Springer Berlin Heidelberg; 1998. p. 4–15.
26. Biryukov A, Pustogarov I. Bitcoin over Tor isn't a Good Idea. In: *2015 IEEE Symposium on Security and Privacy*; 2015. p. 122–134.
27. Figueiredo MA. Lecture notes on bayesian estimation and classification. Instituto de Telecomunicações-Instituto Superior Técnico. 2004; 60.
28. Rish I, Hellerstein J, Thathachar J. An analysis of data characteristics that affect naive Bayes performance. *IBM TJ Watson Research Center*. 30; 2001.
29. Chun B, Culler D, Roscoe T, Bavier A, Peterson L, Wawrzoniak M, et al. PlanetLab: An Overlay Testbed for Broad-coverage Services. *SIGCOMM Comput Commun Rev*. 2003; 33(3):3–12. <https://doi.org/10.1145/956993.956995>
30. MaxMind; <http://www.maxmind.com/>.