- Independent Work Report Fall, 2016 -

De-Anonymizing the Bitcoin Blockchain

Bharath Srivatsan
Partner: Christina Huang
Advisor: Arvind Narayanan

Abstract

Bitcoin is the most popular of a collection of cryptocurriences that have risen to prominence in recent years, promising a de-centralized approach to currency. A belief in transaction anonymity has accompanied, and often propelled, this growth, especially among users who desire privacy. In this paper, we investigate whether one method of 'anonymizing' transaction histories in practice - namely, utilizing different addresses for every transaction - is successful, by attempting to cluster together addresses owned by the same entity. We use a novel approach that involves unsupervised clustering based on structural and attribute similarities in an augmented transaction graph. In doing so, we integrate and build upon past efforts to cluster user addresses using either structural properties of the bitcoin transaction graph or network-level data on user behavior. We ultimately find this methodology to be successful in proof-of-concept tests.

1. Background

Since its inception in January 2009, bitcoin has experienced rapid, aggressive growth in its popularity and usage. [15] Bitcoin is a decentralized digital currency that operates using a Peer-to-Peer (P2P) network. In the system, users engage in transactions by publicly declaring them and cryptographically signing them with associated private keys. These transactions are then verified by miners, special types of entities in the network who perform computationally difficult

calculations (using the concept of Proof-of-Work) before storing transaction information in 'blocks.' After being checked by a vast network of bitcoin nodes, these blocks are linked to all previous blocks in the blockchain, at which point they are propagated through the network as confirmed. Almost all of this work is public on the P2P network – a central benefit of the bitcoin protocol is the ability to independently verify transaction occurrences and the like. [21]

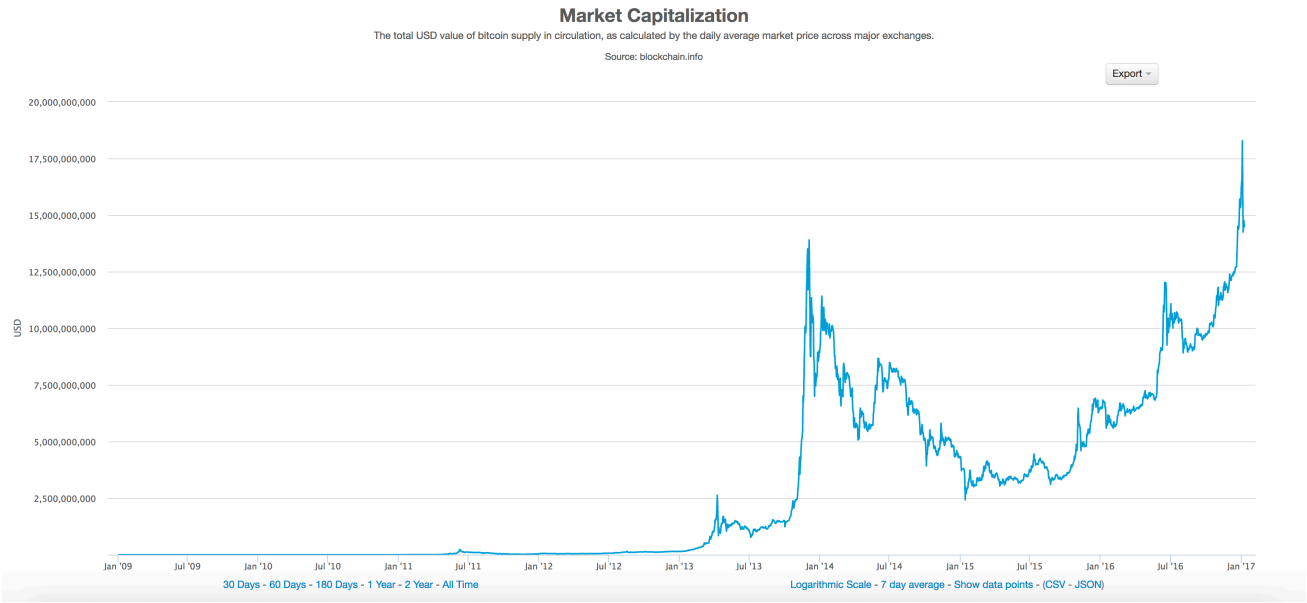


Figure 1: Steady, rapid growth in bitcoin's market capitalization to present.

1.1 Anonymity in Bitcoin

One of the largest debates about the bitcoin ecosystem centers around its anonymity, both in theory and in practice. A belief in the currency's anonymity, at least practically, pervades popular culture and is often used as a reason for switching to the technology. [3][5][7][30] Perhaps this is due to the seemingly random nature of bitcoin addresses – while bitcoin transactions are widely known to be non-private (a record of all transactions is stored in the blockchain), they seem on face anonymous. After all, transactions are stored with user addresses, hashes of letters and numbers not intrinsically linked with the bitcoins coming in to or out of them.

Figure 2: This record of a transaction, for example, contains addresses that look random and meaningless. Source: [28]

Regardless of the reasons for this belief in anonymity, though, its implications are clear and serious. Bitcoins, crucially, are not cash. While it is true that bitcoin's implementation means that user identities aren't directly or automatically tied to specific transactions, it does not automatically follow that user identities *cannot* be so 'tied'. Transactions are merely pseudonymous; if a user's name is ever linked to their addresses, their transaction record (in the past, present, and future) is known to the real world. [21] This isn't a purely philosophical concern - the owner of the Silk Road, Dread Pirate Roberts, once revealed his address in an online forum. [1][9]

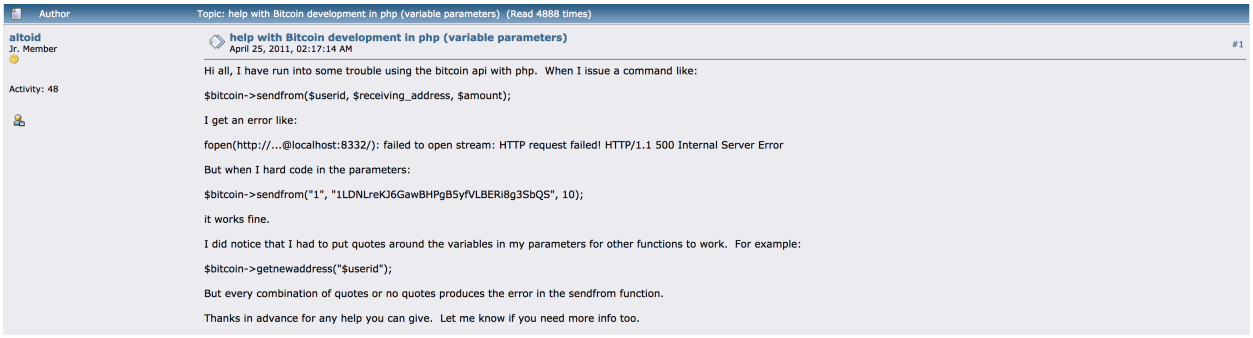


Figure 3: Dread Pirate Roberts, founder of the Silk Road, posts his bitcoin address in this help query. Source: [1]

Which immediately meant that the transactions he'd completed with that address became immediately knowable (even today):

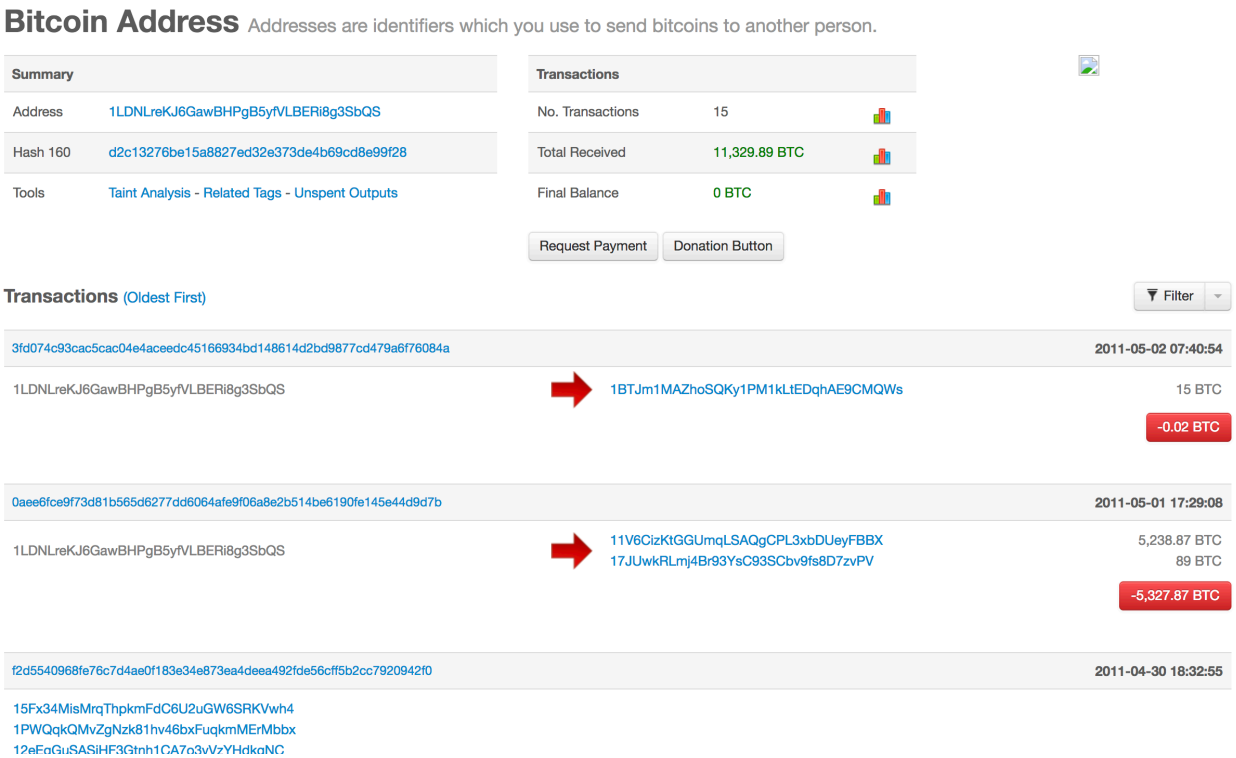


Figure 4: The first few transactions carried out by Roberts (via blockchain.info). Source: [10]

Of course, this doesn't necessarily give us much. We still might not know, for example, who Roberts was interacting with when he made the above transactions, or what the transactions were for. What's clear, though, is that knowledge of a user's addresses (and therefore, transaction history), is the first step towards eroding any conception of anonymity in bitcoin.

Furthermore, it is not necessarily so difficult to find this type of information. Beyond public self-identification, users may inadvertently be signing away their privacy in two other ways. First, many companies in the bitcoin ecosystem have legal and fiduciary responsibilities to collect information when transacting with clients. Thus, a bitcoin bank or exchange could know these crucial links to the real world, placing user information at risk of being exposed to hackers, government officials, or indeed, researchers examining transaction graphs. Similarly, stores sending goods or other real world interactors might also know identifying information like appearances and addresses. [21]

To facilitate a stronger degree of anonymity, Satoshi Nakamoto[1] recommended simply creating a new address for each transaction performed. [20] That way, even though the risk of individual identification still exists, users are protected from their entire transaction histories being released due to any one slip-up.[2] This is crucial, because it would (in theory) make it much harder if not impossible to link together different user transactions. In this world, when Dread Pirate Roberts accidentally posts an address, all but one of his illicit activities remains hidden. Similarly, when a currency exchange gives the government information linking me to one of my addresses or my local shopkeeper describes my appearance to an investigator, the Feds do not then automatically know all other transactions I've engaged in.

The crucial question, then, is whether this approach holds up. Is this enough to anonymize user behavior in practice?

1.2 Motivation

The final question above sets up the motivation for this paper. If there indeed was a way to group together clusters of addresses belonging to the same user, the hard-earned sense of anonymity we'd have created by generating a multitude of addresses would disappear. This could have huge implications. For example, users who have, by nature of their multi-address strategy, never linked their legal and illegal activities explicitly might now find themselves at risk of legal sanction. Others who may have wanted to hide individual transactions for privacy or security reasons could be hurt. Anonymity is powerful, and so the potential to unveil and understand millions of transactions and users at once is important to investigate.

[1] The pseudonymous creator of bitcoin and author of the initial paper about the proposed bitcoin protocol
[2] Most bitcoin wallet providers now automatically perform such an action, automatically and randomly generating addresses for each transaction a client wishes to engage in.

The remainder of this paper is organized as follows. Section 2 surveys some of the ethical implications of this kind of research. I discuss past approaches and other related works in Section 3, and our own approach in section 4. Section 5 contains evaluation results, while section 6 outlines several avenues for proposed future work. I conclude in section 7 by acknowledging the invaluable help of a few individuals.

2. Ethical Implications

Before diving into the details of our research, it is important to sketch its ethical implications. After all, questions of privacy and anonymity are quite morally charged, and so it is important to keep such considerations in mind. There are two relevant questions I will discuss: first, whether anonymity in the abstract is a good thing for the bitcoin ecosystem, and second, whether de-anonymizing users post-facto is ethically legitimate.

Bitcoin today is often associated by laypeople with the seedy underbelly of the internet - it is believed to facilitate the trade in drugs, arms, and people, and was (until recently) the underlying financial system used by the Silk Road (an online store for things illegal). [3] Furthermore, there exist legitimate fears that the un-traceability of bitcoin means that users can launder money or evade taxes by transacting exclusively on the network. Anonymity in the blockchain, then, does not come without its costs.

On the other hand, anonymity facilitates many of the behaviors that bitcoin advocates are so proud of. It is because bitcoin is pseudonymous, for example, that dissidents and companies like Wikileaks can continue to be funded even when other routes of obtaining payment dry up. More generally, many financial interactions that we engage in daily are intended to be private; given that bitcoin transactions aren't, they must at least be shielded from *identification* by the public eye.

On net, it is thus difficult to establish whether anonymity in bitcoin is good or bad from an objective moral standpoint.[3] Research like mine does not attempt to answer this question; instead, this paper aims to investigate whether the benefits proclaimed are actually being accrued by the cryptocurrency's users. After all, the positives of anonymity backfire if certain actors (whether private or public) gain the ability to identify users in secret.

The second question is both narrower and more topical. On one hand, it seems questionable to be generating and discussing methods of de-anonymization that would operate on a system widely believed to be anonymous in the status quo. Indeed, as mentioned in section 1.1, many users were in fact drawn to the currency because of this perception. Thus, de-anonymization seems in some ways to be a breach of trust, an academic analysis that will have all-too-real implications for individuals on the network who have staked their livelihoods and perhaps even security on this belief. Further, it may strike observers as particularly wrong to be performing these studies on old datasets from time periods before papers like "Fistful" prompted a general acknowledgement (within at least the technical community) of the cryptocurrency's lack of true anonymity.

These concerns do not stand deep scrutiny, though; for any that do, I explain later how we attempt to avoid them in this paper. First, the blockchain is incontrovertibly public. Any rational bitcoin user should understand that analyses like the one I am performing can and do occur either way - Nakamoto himself warned that this would be the case. [20] Furthermore, the application of academic scrutiny to this question might be able to energize efforts to increase the security of the protocol or serve as a signaling mechanism for others to switch to alternative providers. Thus, in the long term, discussions like this one could be useful in protecting users from the worst excesses of intrusive governments or private actors. Finally, the mere presumption of security ought not

[3] Not so from a financial standpoint, where anonymity is clearly incredibly lucrative

invalidate any attempts to investigate or illustrate the truth of such claims - it is perfectly legitimate (indeed, almost obligatory) to identify the flaws in a widely-trusted system, just as it would be to point out glaring security holes in a bank's website code.

The above defense notwithstanding, in this paper, we do not employ any methods of tagging users that are not public - we use only information scraped from public sites on the web. We also do not attempt to publish personal information about individual users, nor do we identify methods of such identification. As such, hopefully any remnant privacy concerns surrounding this issue are mitigated.

Finally, and briefly, a quick note. Certain de-anonymization approaches are met with more backlash than others within the bitcoin community. Unlike static analyses of the blockchain, actively listening for network information to unmask users is sometimes frowned upon. Private blockchain analysis company Chainalysis felt a keen backlash when it was accused of setting up listener nodes across the bitcoin network to monitor and collect user IP address information and more. [22] Since we do not personally collect any such information - we borrow the user tags from other projects, the “Fistful” authors, and from public APIs like that of Blockchain.info - we do not have to deal with these repercussions.

3. Related Work

Past work into investigating anonymity in the bitcoin blockchain has generally proceeded along one of two paths:

1. **Clustering User Addresses:** Many papers (and most implemented clustering systems) find heuristics that roughly correspond to co-ownership of addresses in the blockchain

2. **Identifying Users:** Other papers investigate ways to map real world identities to addresses (or clusters of addresses) [4]
    a. **… using public information:** Some papers (like Meiklejohn et al.) attempt to identify users manually by transacting with companies and parsing forums [19]
    b. **… using network data:** Other researchers have investigated whether attackers could discover user information by exploiting loopholes in the bitcoin network [4]

Though our project is focused on clustering addresses, we build upon some of the work done by network-level researchers in the past as well. As such, I sketch past work done in both the clustering space and the network-based identification space.

3.1 Heuristic-based clusters

Reid and Harrigan in 2011 and Ron and Shamir in 2012 both perform quantitative analyses of the transaction history stored in the blockchain – Reid with an eye towards privacy and examining individual transaction flows from idiosyncrasies like thefts, and Ron looking towards broader anonymity concerns. [23][24] Ron and Shamir develop a clustering heuristic for multi-input transactions (originally mentioned by Satoshi Nakamoto) that identifies all addresses going into a given transaction as controlled by the same user. [24]

In "A Fistful of Bitcoins", Meiklejohn et al. discuss two heuristics constructed around idioms of use in the bitcoin ecosystem that allow them to identify many co-owned addresses. The first of these was the multi-input heuristic discussed by Ron and Shamir; the second relies on identifying 'change addresses' – output addresses that belong to the same owner as the input address. Using these methods, Meiklejohn et al. partition 12,056,684 public keys into 3,384,179

[4] Technically, comprehensive user identification would necessarily come along with address ownership clustering, though this seems like a valid distinction for now

clusters. [19] These two techniques are later used in a whole host of applications, whether academic (Fleder et al. improve the identification and scraping components of the de-anonymization process to name more clusters), applied (btctracker and BitIodine are applications built for real-time heuristic clustering of the blockchain), or commercial (Chainalysis, mentioned above, used these concepts as a launching point into developing a far larger suite of heuristics). [6][8][26]

3.2 Network Analyses

In 2011, security researcher Dan Kaminsky gave a talk at a BlackOps conference that would serve as the backbone for many academic network analyses of bitcoin flows going forward. In order to collect reliable, high-confidence information about ownership of particular addresses, he claimed, an attacker could merely listen to every bitcoin node at once, identifying each transaction as 'owned' by the first IP to relay it. [12] Koshy et al. implemented this approach, but managed to de-anonymize only 1,162 addresses over the course of their 5-month listening period (from 5.6 million transactions). This is largely because their methods relied on exploiting anomalous, single-input transactions, and they could only identify the addresses of bitcoin servers (as opposed to clients, who are 9x as populous). [13][4]

Biryukov et al.'s paper proposed a method to explicitly target clients, including those behind NAT or firewalls. This involves listening to server requests made by transacting parties and uniquely identifying users based on sets of at least 3 'entry nodes' they communicated with. This approach was used to great success on a test network. [4] In the real bitcoin ecosystem, but by using a Bayesian classification approach to IP-identity matching (i.e. by probabilistically guessing identity not purely based off the first relay but also other closely connected nodes), Juhász et al. were able to identify 22,363 distinct addresses. [11] Finally, Androulaki et al. attempted to

cluster addresses in a test network based on user behavior and network characteristics and using unsupervised learning algorithms like k-means clustering. [2]

For the purposes of this paper, we do not engage deeply with the *methods* outlined above – we do not attempt to set up listeners and identify broadcasting IP addresses, etc. That said, we utilize past work and network data from blockchain.info on relaying IP addresses, timestamps, etc., as one input to our clustering attempts.

4. Approach

Our approach involves unsupervised clustering of addresses based on transaction information from the blockchain and the network. In doing so, we extend upon past work done in the space in a few salient ways:

- We contribute a new 'loose heuristic' for address similarity that is distinct from the two clustering heuristics implemented by Meiklejohn et al.
- We represent information from both clustering heuristics and network listening in an augmented structure/attribute similarity graph
- We are the first team (to our knowledge) to apply state-of-the-art unsupervised clustering algorithms that do not require approximations of user numbers to this problem
- We demonstrate the viability of our approach by implementing it in python and applying it to a selected case study and evaluation metric

In this section, I will discuss first the nature of the data we used. I will then expand on the graph we constructed (upon which we would cluster transactions) and the specific heuristics and network-level data we incorporated. Finally, I will outline the technique we used for clustering addresses.

Before moving further, a brief note on the definition of 'ownership.' While we understand that there exist various ways to define and establish ownership of a given address, for clarity and simplicity we align with past papers in defining ownership of addresses as possession of the relevant public and private keys. We use this term interchangeably with control (of addresses).

4.1 Data Collection

We had four sources of data available to us for this clustering approach:

1. Blockchain information
2. Network information
3. Clusters found by Meiklejohn et al.
4. Clusters created by Chainalysis

The first was a dump of the complete bitcoin blockchain, downloaded at around 12:30 AM on 10/7/2016. This included transactional information – transaction hashes, input and output transactions, addresses, script type, etc. – and block-related information – block height, block hashes, etc. Granted that transaction placement in blocks is largely arbitrary, and any useful information we could gain from co-occurrences of transactions in blocks, etc., would be present in timestamp information as well, we did not use the block-related information while clustering.

The second source of information we used was the blockchain.info API. From the public block explorer site, we obtained more network data on transactions, including approximate timestamps and IP addresses. Blockchain.info uses a similar approach to mapping IP addresses to some of the network-level identification research outlined in section 3.2, but was still likely to mismatch addresses at some points. Our approach (as explained below) could handle egregious mismatches, though, so we used the information provided regardless. One problem with this information,

though, was a severe rate-limiting of the API endpoints – we were unable to reach an agreement with them for access to the full blockchain's worth of information.

We also contacted and communicated with Sarah Meiklejohn, and were able to obtain the clustering output that her scripts yielded. Given that we were building directly off of her approach, though, and that there doesn't exist a ground-truth dataset for address clusters, we didn't think direct comparisons with her clusters would be of much use in evaluating our project.

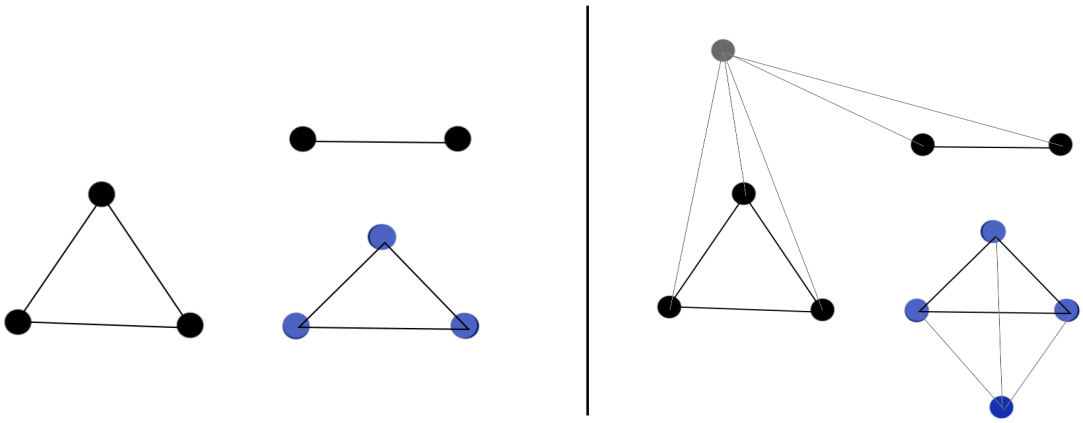
By contrast, our agreement with blockchain analysis company Chainalysis proved far more promising. Beyond the two heuristics developed in the "Fistful" paper (that we also used), the firm has developed thousands of other heuristics based off an understanding of idioms of usage in the bitcoin ecosystem. We decided to use this dataset as a ground truth set.

4.2 Structural-Attribute Similarity

After collecting the above data, we were looking at, broadly, two types of information. We had attribute-style information on individual transactions (script types used, relaying IP address, etc.), and we had structural information about the transaction graph (which transactions were used as inputs to others, which were likely clustered together, etc.). However, attempting to build an unsupervised clustering algorithm that would somehow learn the weights of some parameters to assign to these different *types* of information would be very difficult (and unlikely to converge). [31] We therefore needed a way to represent both structure and attribute information in a single form for clustering.

To achieve this behavior, we used a setup inspired by SA-Cluster, a graph clustering algorithm described by Zhou et al. in a 2009 paper. This involved resolving all attributes to vertices in the graph, thereby folding all relevant information into the graph's topological structure. [31] To do this, we augmented the existing, structured transaction graph with vertices representing

values for each attribute. Then, we drew edges from existing vertices (transactions) to attribute vertices based on the values associated with each transaction on each attribute. In this way, we built a graph that contained in its structure both the underlying transactional information and measures of attribute similarity that we wanted to consider.



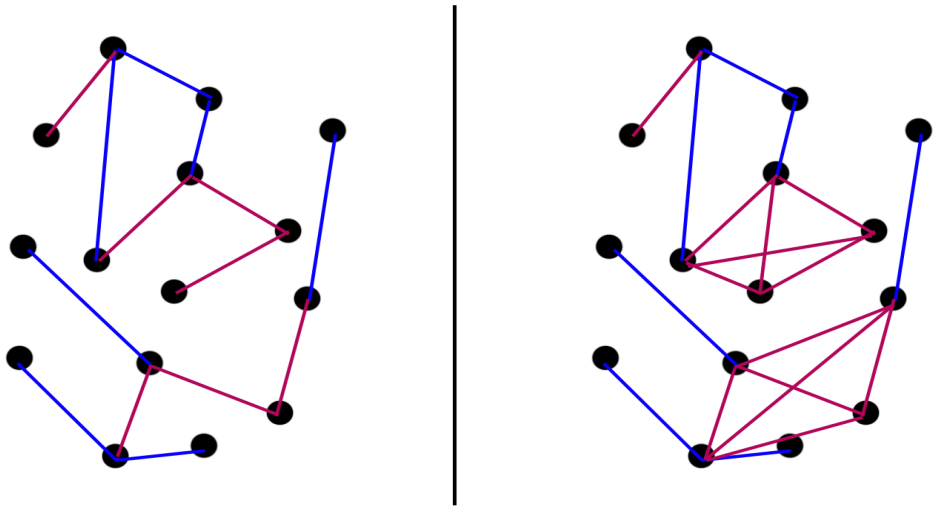
Figures 5, 6: In the left image, we have the original graph. Beyond its structure, it has various attributional qualities that should also be taken into account (represented by the colors). On the right, we create points to represent the 'black' and 'blue' attributes and draw the appropriate edges into the graph. Now, the colors no longer matter – we have represented that information into the structure itself!

Finally, we had to resolve the issue of infinite attribute values. For an attribute like a transaction's timestamp, the set of all possible values would be very, very large. Clearly it would not make sense to clutter the graph with all these points! Instead, we wanted to represent information on similarity – 12:05 should be 'close' to 12:06, for example. To get this, we 'bucketed' these values, subdividing times of day, etc. into a far smaller collection of points and modifying our code to check each transaction's attribute value against these bucketed ranges. While imperfect, this solution allowed us to avoid unnecessary extraneous graph vertices, and better allowed us to measure similarity.

4.3 Graph Setup

Within the scope of the above graph design, we set out to build our graph for clustering. In the graph, nodes represented (and were indexed by) individual transactions, each of which had a corresponding entry in a separate table containing input/output information (which was relevant for analysis and construction but not for clustering). Meanwhile, edges were 'connective links' between transactions that indicated some degree of similarity. For example, heuristic 1 connected nodes with any of the same inputs as each other (since these inputs were presumably all co-owned). Similarly, if two transactions each used the same IP address, they would each have an edge drawn to that augmented IP address vertex. The closeness of these two points (only two steps away) would serve as an indicator of their similarity for the clustering algorithm.

In the above system, edges were indications of similarity, not guarantees of it. As such, we needed a way to indicate the *strength* of particular types of connections – a connection due to heuristic 1 (which we are almost certain is true) should certainly be valued above a connection due to presence in the same time bucket. At the same time, we couldn't set the values of types of edges without needlessly complicating the resulting clustering algorithm. We settled on strongly connecting the links from particularly high-confidence heuristics. This meant that after joining all the transactions flagged by heuristics 1 and 2, the algorithm went through and strongly-connected these clusters, turning them into cliques. In practice, we found that this almost invariably led to those cliques being clustered together, which was perfectly in line with the strong deference we wished to place on such heuristics being good indicators of clustered addresses.



Figures 7,8: In the above image, transactions (dots) are linked by either heuristics 1 or 2 (represented by pink lines) or by other similarity measures like heuristic 3 or the augmented network data (represented by blue lines). The former sets of nodes are strongly connected on the right by our algorithm to represent the strength of their similarity.

Finally, augmented nodes were indexed by a system of negative numbers to represent buckets and values. Augmented edges were not distinguished from regular edges. This meant that the final clustering algorithm would inevitably include augmented nodes in the clusters since they were functionally equivalent to regular nodes, but we ignored their cluster values in the output analysis.

4.4 Graph Characteristics

We included 6 transaction characteristics in our graph: three heuristics that connected transactions directly (structural aspects) and three attributes that were used to indicate transaction similarities.

Heuristic 1: This is the same heuristic mentioned in section 3.1 – it links addresses that are used as co-inputs to a single transaction. Under our definitions of user ownership and control,

knowing the private keys for multiple addresses implies ownership of those addresses, and so multi-input transactions likely involve addresses owned by the same user.[5] [24]

Figure 9: In the above transaction, all of the input addresses would be flagged as co-owned by heuristic 1, since the transaction originator likely knew the private keys for each. Source: [27]

Heuristic 2: The second heuristic is an adapted version of the one outlined in "Fistful," and involves identifying change addresses controlled by the owner of the inputs. As pointed out by Meiklejohn et al., bitcoin uses change addresses as the way to give back excess funds to the originator of a transaction, since the protocol requires that all the funds from the input address(es) be cleared in each transaction. As such, identifying this change address is another way to cluster addresses and establish co-ownership. To do so, we utilize Meiklejohn's approach of flagging output addresses that are new, but build upon this method by requiring that the proposed change address and the original input address be using the same script type. This is because if a single wallet provider was indeed setting up a change address, it would likely use the same script type as the address it already maintains (the input address).[6] [19]

Figure 10: In this transaction with a known change address, the user pays 10 BTC to some service and receives 0.89 BTC to a change address they control. Our heuristic would attempt to join the current transaction with the transaction of the first (change) address. Source: [29]

[5] I say likely as opposed to certainly because new idioms of usage in bitcoin like coin mixing could lead to multi-input transactions from a group of people who each individually sign the transaction, violating this heuristic

[6] Due to time constraints, we were unable to further refine this heuristic in the way that the "Fistful" authors did, by manually examining and identifying exceptions to the norms, and correcting for them

Heuristic 3: Our third heuristic is novel; it isn't a high-confidence heuristic and so doesn't warrant the strong-connection as do the first two, but it is likely to represent possible co-ownership of clusters. In this heuristic, we join single-link transactions, transactions with a single input and a single output. Bitcoin transactions involve the clearing out of the input addresses – all of the bitcoins are spent at once – and so it seems somewhat unlikely that the amount in the input address is exactly how much was needed for a transaction with a *different* output entity. By contrast, individuals moving money around their own accounts or trying to avoid tracking could lengthen their chain of transactions by engaging in single-link transactions.

However, there are two features that make this a low-confidence heuristic. First, some behaviors or idioms of use specifically confuse it. For example, some single-link transactions could simply be the output of a multi-link transaction that created an address containing the exact number of bitcoins spent. Also, while users may manually move their bitcoins around, modern wallet technologies sometimes mean that they do not control such movements, and the typical rationale for single-linking may no longer hold. The strength of our graph clustering methodology, though, lies in the fact that we can include low-confidence heuristics like this one, knowing that they will only serve as one input of many in the graph clustering algorithm. If and when this heuristic coincides with similarities among other metrics, the algorithm would rightly cluster the relevant transactions.

Figure 11: This single-link transaction would be flagged by heuristic 3, and an edge would be drawn to the transactions of the input and output addresses. Source: [28]

IP Addresses: Based on the work done by past authors (see section 3.2), we decided to use information about the first IP address to relay a given transaction when clustering it. This information was obtained from the blockchain.info API and was represented as a string of 10 digits (or zeros). Relaying IP addresses are a reasonable similarity metric for clustering addresses because it is likely that a user in a given location relays their transactions through the same addresses.

Timestamp: Another signal for user behavior similarity is the timestamp of a given transaction. While this isn't stored in the blockchain itself (since time is a hazy concept – transactions might not be confirmed until hours after they are first authorized), we were able to find timestamp information using blockchain.info. We hypothesized that users who owned multiple addresses might be likely to spend from them at similar times of the day.

Script Type: Finally, we extracted information about the script types used by each of the addresses involved in a transaction. This was most immediately implemented in our adaptation of heuristic 2 (above), but was also used as a similarity measure. Users who utilize a given wallet software to manage all their addresses are likely to have the same script type for each address; as such, monitoring addresses that correspond to the same script type seems to be a good first step towards establishing co-ownership.

4.5 Clustering

Now, having described the graph, we turn to the clustering approach used. In searching for an appropriate clustering algorithm, we had five important considerations in mind:

1. **Noisy Data** – We expected the data to be noisy, that is, to contain nodes that wouldn't belong to any cluster, because we expected large numbers of users with only one address. We therefore needed a clustering algorithm that could accommodate noisy data

2. **Varying-Numbers of Clusters** – Because there doesn't exist any ground-truth data on clusters of addresses, we were unable to definitively establish a-priori the number of clusters we aimed to create
3. **Varying-Density Clusters** – These clusters would also likely vary in density, since some user clusters would be connected by a variety of similarity measures while others would only be joined with a heuristic
4. **Floating Data** – The data we had consisted of adjacency information, not positional information; we represented transactions as linked to one another and to attribute values, but never set distances to those links or positions to those transactions (because any such initial setting would have been arbitrary). Thus, we needed a clustering approach that could deal with non-positional data
5. **Large Data Size** – Given that we had originally intended to run these algorithms on a graph constructed upon the entire blockchain (with over 400 million transactions), we needed an algorithm that could accurately and quickly partition this massive dataset into clusters

We then proceeded to evaluate a selection of available unsupervised graph clustering algorithms against these considerations.

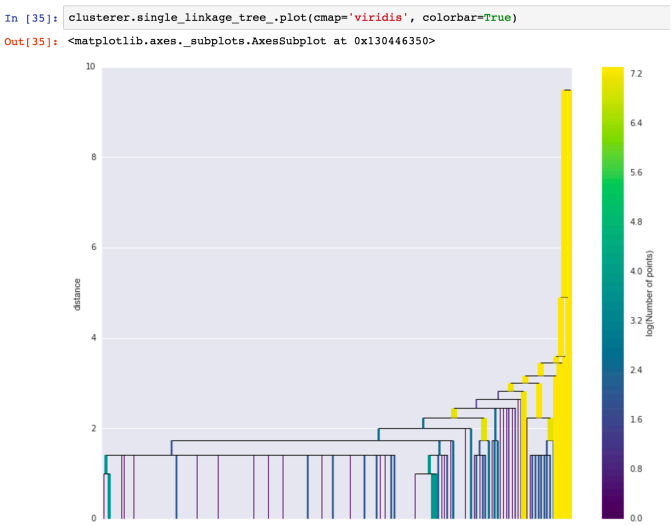
Algorithm	Eliminating Consideration(s)	Explanation
K-Means	1, 2, 3	K-means partitions the data, meaning that it lumps noise into clusters, requires knowledge of cluster numbers, and always looks for globular clusters
Fuzzy C-Means	1, 2	Fuzzy C-means improves upon k-means because its 'fuzziness' factor allows different densities, but it still fails considerations 1 and 2
Affinity Propagation	1, 4, 5	This is another improvement upon k-means by allowing clusters to form naturally and pick 'exemplars', but still includes noise and now requires positional information and smaller data sets
Mean Shift	2, 5	By placing centroids at the highest density points, mean shift can accommodate noise and varying density clusters, but it still requires an initial number of clusters and is quite slow
Spectral Clustering	1, 2	Spectral clustering attempts to embed the graph into Euclidean space before performing k-means. It thus avoids problem 4, but in the process, assumes some of the issues of k-means
Agglomerative (e.g. Single Linkage Clustering)	1, 5	These clustering techniques involve building clusters from the ground up by linking points together. Unfortunately, this means that they often cannot accommodate noise, which skews cluster creation
Divisive Clustering	1, 5	By contrast, these techniques involve dividing the graph into clusters from the top down, but this is not resilient in the face of noise and is relatively slow
DBSCAN	3	DBSCAN transforms the graph to prioritize dense clusters, thereby accommodating noise, generating its own number of clusters, and not requiring positional information. It doesn't accommodate varying-density clusters, though
HDBSCAN	--	

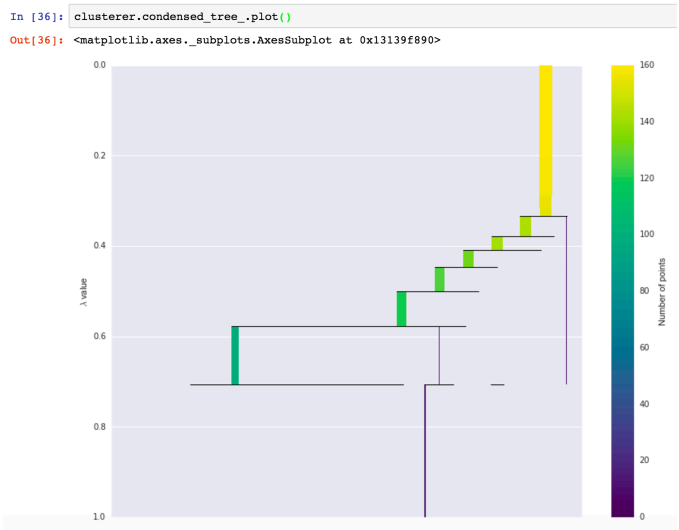
Figure 12: An analysis of the different graph clustering options available to us, with explanations of their failures with respect to our most important considerations [17]

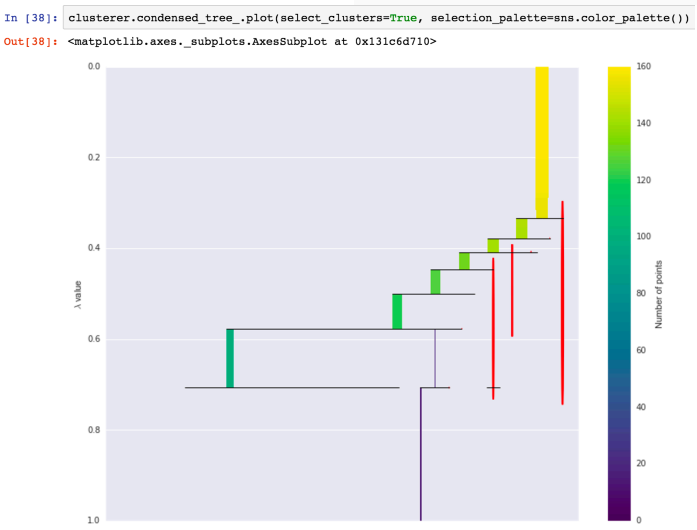
Based on the above analysis, we eventually settled on the clustering algorithm HDBSCAN. This algorithm satisfies all our five considerations – it can register points as noise, it decides on

the number of clusters present in the graph automatically, it improves upon DBSCAN by being able to adequately deal with varying-density clusters, it doesn't require positional information, and it performs adequately quickly. [17]

The algorithm works by first modifying the graph, pushing outliers further away to highlight particularly dense clusters. Then, it organizes these nodes into a cluster hierarchy of connected components based on distance, and performs single-linkage clustering on the remaining points. It decides when to stop links from clustering further by building a condensed version of the cluster tree and identifying the most reasonable cutoff points for each branch. [18]







Figures 13, 14, 15: The three figures above illustrate the steps of the HDBSCAN algorithm, as performed on the SA-graph generated for the first 100 transactions on the blockchain. On the top left, the cluster hierarchy tree is first created. On the top right, the tree is condensed to identify cutoff points for single-linkage clustering. Finally, on the bottom, the clusters are identified (and colored appropriately). Plotting commands sourced from: [18]

For the purposes of this project, we used the Python hdbscan package, version 0.8.4. This allowed us to focus on tuning the relevant parameters to best suit our data. After trial and error, we ultimately used parameter settings of 2 for minimum cluster size and 3 for minimum sample size. The former was selected because we knew that owners could own as few as 2 addresses, while the latter was picked because it best allowed us to approximate the number of clusters Chainalysis found for a variety of test data sets. [16]

5. Results and Evaluation

The largest issue we faced in evaluating our project was the lack of a reliable ground truth dataset upon which to test our clusters. Since bitcoin is pseudonymous in the status quo, there was no foolproof way for us to establish whether given addresses were indeed co-owned, especially when the relevant linking transactions had occurred once and far in the past. Furthermore, because of limitations in our ability to procure network-level data from blockchain.info, we were never able to run our methods on massive chunks of transaction data. In this section I outline the intermediary results we obtained, including visualizations of the graph creation process, and two evaluation metrics we used – one qualitative comparison to Chainalysis-created clusters and one case study.

5.1 Visualizations

As we implemented our graph clustering approach, we performed quick sanity checks by examining the outputs of individual steps in the process (for example, individual heuristics). We then visualized these intermediary steps in the process to better demonstrate the mechanics of our

procedure. The following images all show the results of our algorithms applied to the first 100 transaction nodes in the blockchain. The decision to visualize the first 100 was made for clarity's sake – any more numerous and the graphs would come out as large, meaningless blobs of nodes.

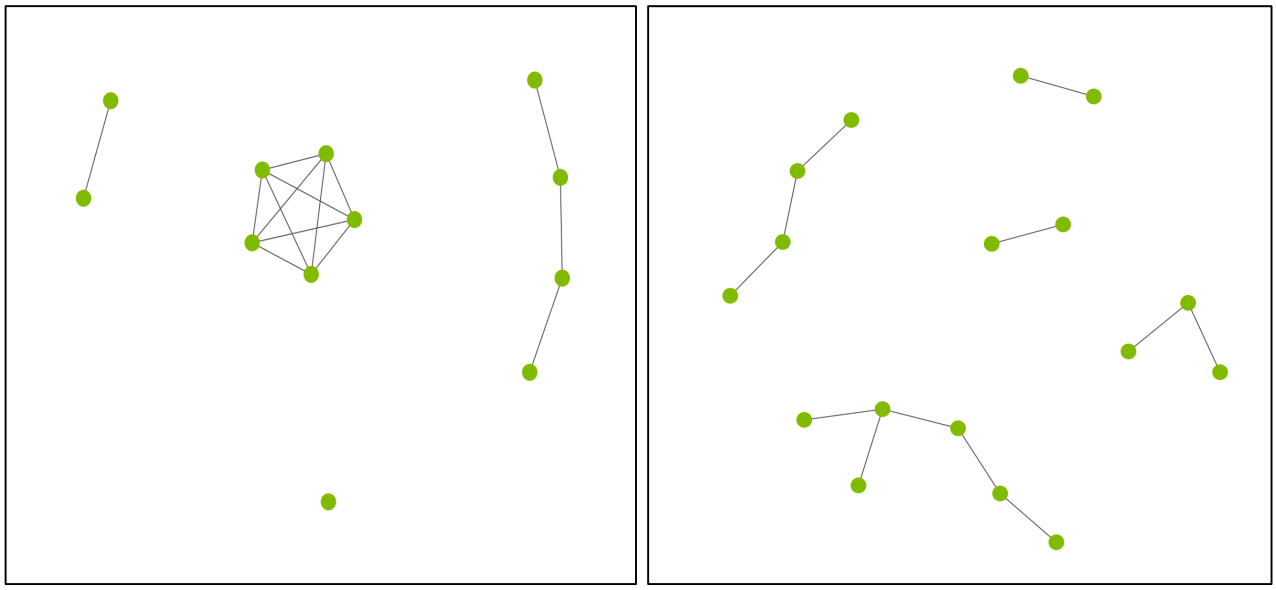


Figure 16, left, shows the results of linking transactions with heuristic 1 alone, while figure 17, right, shows the output of heuristic 2. Note that different transactions have been linked in different ways, which is expected and desired behavior from the two different heuristics

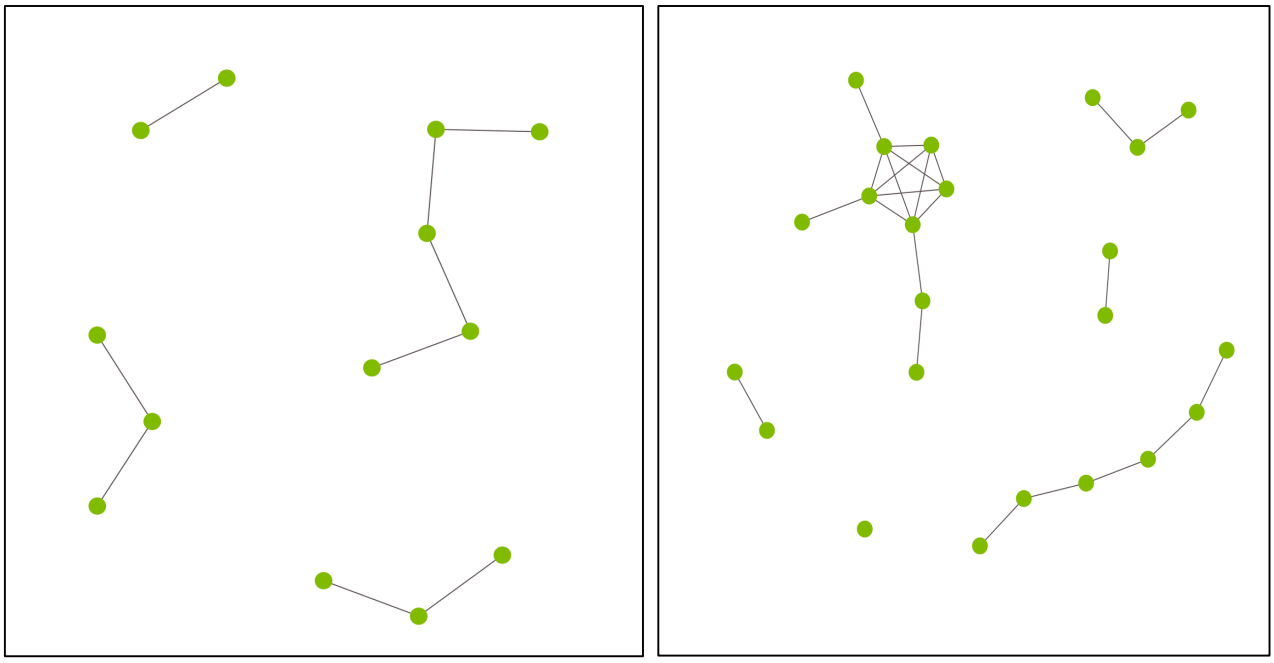


Figure 18, left, shows the results of linking with heuristic 3, while figure 19, right, shows the overall output graph after linking with all three heuristics.

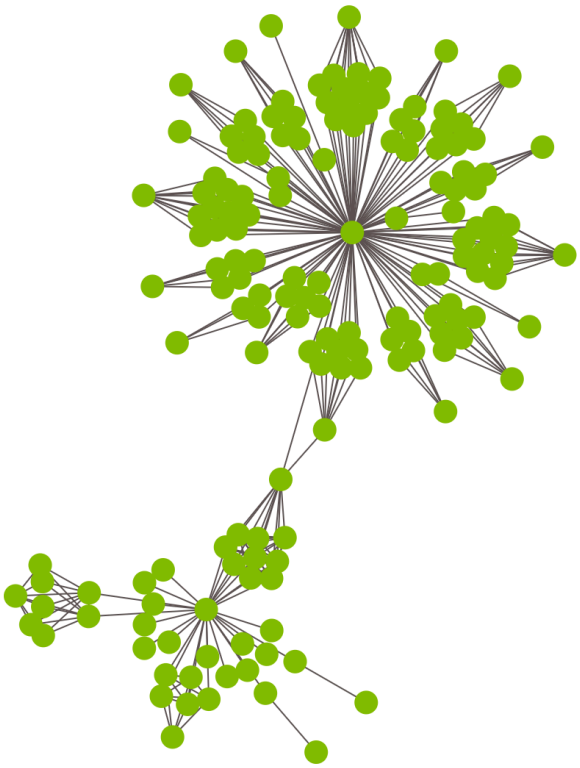


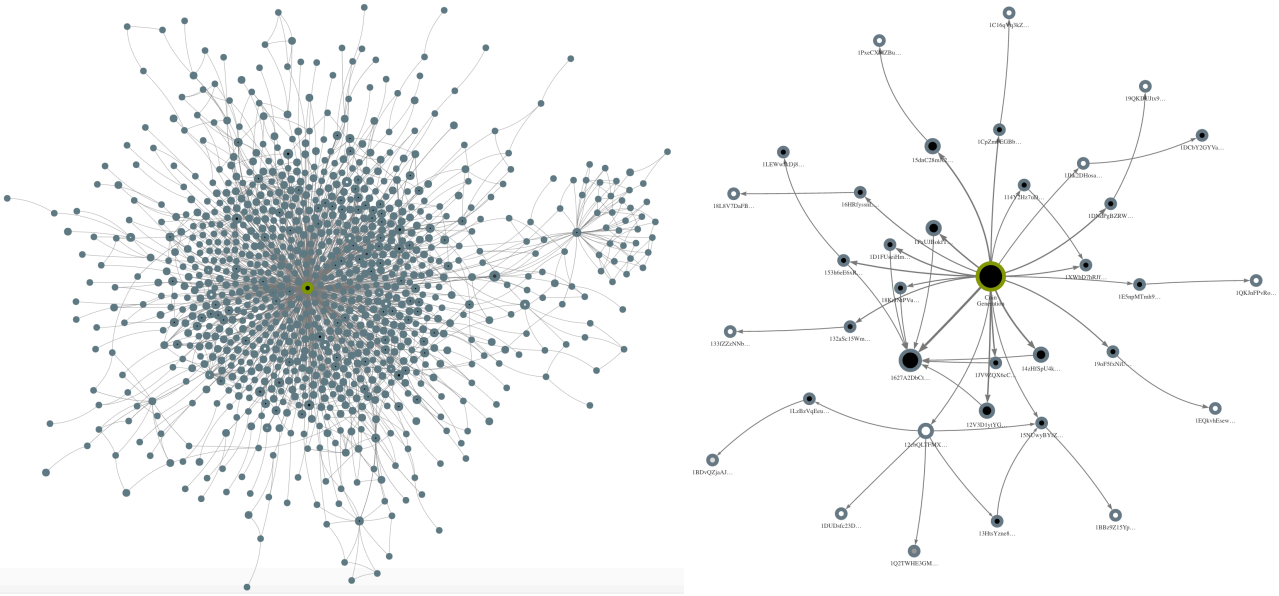
Figure 20: The final, attribute-augmented version of the graph created by the algorithm described in the approach section above, performed on the first 100 transactions. This was the input sent in to the clustering algorithm

```
[-1  8  9 19  0  9 20  5 24  9  5  6  6  6 26 -1 29 30 28 24 27 25 27 -1 23
 22 -1 21 -1 25 15 26 28 11 12 12 13 30 10 14  0 17 18  0  0  0 19  0 14 23
  9 24 15  0 26 26 29 17 14 19  0 16 20 22 16  7  9  9 -1  7  7 23 24 27  3
 13 13 21 30  7  7  8 25 25 28 14  7 15 14 20 21 14 30 10 21 22 21  9 18 29
 26 12 12 11 11  8 30  8 27 27 -1  4 21  4  3 -1 -1 -1  2  4  4  2  4  1  1
 -1  2 -1 -1  2 -1  1  1  2  4  0 10 10 -1 17]
```

Figure 21: The raw output of our clustering algorithm. Each of these numbers represents a cluster assignment for the corresponding points in the transaction graph input. As we can see, this means that HDBSCAN clustered these points into 31 clusters

5.2 Chainalysis Evaluation

The first evaluation metric we used was a qualitative comparison of our clusters against the clusters generated by Chainalysis's heuristics. Because the visualization software that we had access to was unwieldy for massive datasets (see below), we instead evaluated against small subsets of data.



Figures 22, 23: On the left, an illustration of the impracticality of working with large Chainalysis graphs for evaluation purposes (this graph contains 995 nodes). On the right, the corresponding clustered Chainalysis graph for the 100-node SA-graph clustered in stages earlier (section 5.1).

On these subsets, our methodology proved to be quite effective. On the 100-transaction clustering above, for example, we found that both the number of clusters created by our algorithm and the composition of these clusters roughly matched those hypothesized by Chainalysis. We attempted this same approach on slightly larger graphs as well, and found in spot checks of individual clusters that our approach often matched up. Though network-data and evaluation-data limitations prevented us from systematizing these results, we are cautiously optimistic that our algorithm was largely able to approximate the additional heuristic information incorporated into the Chainalysis clusters.

5.3 Case Study: Theft Analysis

The second evaluation metric we used involved testing our clustering methodology on a known, real-world use case – a theft of bitcoins. To do so, we found a series of transactions on the blockchain known to be thefts carried out by the same user, and tested whether our algorithm would cluster them (as desired). The 2012 50BTC theft was a perfect candidate for this analysis – it involved four known transactions, a substantial sum of bitcoins in total stolen (over 1,173), and is as yet unresolved. In this occurrence, an unknown intruder stole thousands of bitcoins by hacking the 50BTC mining pool's billing software. [14]

We began by finding the relevant transactions in the blockchain.

- 9dfdb24667657365c469ff20568fcc820f6f028a125d9c22dc521ae44dcf7c5e
- bd2ad7b49c22d12cf2f8f12ef601952aed2a96907af4df732156fd90165b5ef5
- d0035ad189634e90239cca82eb53f78e08c0179620b2bd24e2cb291478c7d57a
- a2b642bafea45bc128d81314ef33542bc807811ba066329eaa1306bd62bec075

Figure 24: Transaction hashes involved in the 50BTC theft in 2012. Source: [14]

Then, we collected a set of 10,000 transactions from the surrounding time period (including these four). Our goal in doing this was to obscure these transactions among many others, so that it would be harder in theory for our clustering algorithm to pick them out. Finally, we ran the clustering algorithm on this dataset and output the resulting cluster labels.

...	**7953108 2**
7953092 1	7953109 39
7953093 1	7953110 -1
7953094 -1	7953111 -1
7953095 2	...
7953096 -1	7953139 37
7953097 -1	-692446476 66
…	7953141 66
7953102 19	**7953142 2**
7953104 -1	7953143 66
7953105 -1	7953144 39
7953106 2	7953145 37
7953107 -1	...

Figure 25: Selected pieces of the clustering output from running our methods on 10,000 transactions relevant to the 50BTC theft. Each row represents a node ID (e.g. 7953092) and its associated cluster ID (eg. 1). The four bolded transactions are the node IDs for the theft transactions mentioned above; all four have been placed into the same cluster.

Ultimately, our methodology successfully managed to cluster the four transactions involved in the 50BTC theft. While this result cannot necessarily be generalized to other thefts or similar occurrences, it serves as a proof of concept for this methodology.

6. Future Work

There are a few avenues for future work in this space. Most centrally, we recommend that additional research be done to further refine our algorithms and heuristics, and to comprehensively evaluate our methodology against the entire blockchain. Doing so will require access to network-level information and evaluation data for far more transactions than we did.

Further research should also examine ways to establish a more resilient anonymity, especially in response to the growing body of address clustering research. Approaches like the use of Mixcoin in or the incorporation of k-anonymity into the bitcoin ecosystem are good potential launching points for these investigations.

7. Conclusion and Acknowledgements

In this paper, we examined a novel approach to clustering bitcoin addresses that used unsupervised clustering on an augmented structure-attribute graph. In doing so, we incorporated and integrated a variety of past methods and implemented a structure that can easily be expanded and applied to other relevant attributes or structural elements in the future. This methodology was successful in clustering the test cases we evaluated, though we recommend that future work be done in the space to obtain a more comprehensive dataset and to thereby run the algorithms on a larger subset of the blockchain.

Through this process, we have received invaluable help from several individuals. We would like to thank Professor Narayanan for guiding us through the process and pointing us in the right directions, and the graduate TAs from our independent work seminar, Harry and Steven, for their help in data collection. We would also like to thank Sarah Meiklejohn for speaking with us on her approach and for providing us with her clustering dataset and algorithms. Finally, we'd like to thank Jonathan Levin at Chainalysis for allowing us to use his product in the evaluation stages of our project.

8. References

[1] Altoid. "Help with Bitcoin development in php (Variable parameters)." *BitcoinTalk*, 25 Apr. 2011, bitcointalk.org/index.php?topic=6460.0.

[2] Androulaki, Elli et al. "Evaluating User Privacy in Bitcoin." *Financial Cryptography and Data Security Lecture Notes in Computer Science*, 2013, pp. 34–51. doi:10.1007/978-3-642-39884-1_4.

[3] "Anonymity." 5 Aug. 2013, bitcoinsimplified.org/learn-more/anonymity/.

[4] Biryukov, Alex et al. "Deanonymisation of Clients in Bitcoin P2P Network." *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14*, 2014.

[5] Chi, Leisha. "Bitcoin digital currency hits three-Year high of $1,000." *BBC News*, 3 Jan. 2017, www.bbc.com/news/business-38495804.

[6] Doll, Aaron et al. "Btctrackr : Finding and Displaying Clusters in Bitcoin." 14 May 2014.

[7] "Donate to WikiLeaks." *Wikileaks*, shop.wikileaks.org/donate.

[8] Fleder, Michael et al. "Bitcoin Transaction Graph Analysis." *CSAIL*, 3 Jan. 2014.

[9] Goodin, Dan. “FBI: Silk Road mastermind couldn’t even keep himself anonymous online.” *Ars Technica*, 2 Oct. 2013, arstechnica.com/security/2013/10/silk-road-mastermind-unmasked-by-rookie-goofs-complaint-alleges/.
[10] “Home View information about a bitcoin address.” blockchain.info/address/1LDNLreKJ6GawBHPgB5yfVLBERi8g3SbQS.
[11] Juhasz, P´eter L et al. “A Bayesian Approach to Identify Bitcoin Users.” *Senseable City Lab*, 20 Dec. 2016.
[12] Kaminsky, Dan. “Black Ops of TCP/IP 2011.” *Dan Kaminsky's Blog*, 4 Aug. 2011, dankaminsky.com/2011/08/05/bo2k11/.
[13] Koshy, Philip et al. “An Analysis of Anonymity in Bitcoin Using P2P Network Traffic.” *Financial Cryptography and Data Security Lecture Notes in Computer Science*, 2014, pp. 469–485. doi:10.1007/978-3-662-45472-5_30.
[14] Dree12. “List of Major Bitcoin Heists, Thefts, Hacks, Scams, and Losses.” *BitcoinTalk*, 19 Apr. 2014, bitcointalk.org/index.php?topic=576337#post_t2012_50_btc_theft.
[15] “Market Capitalization.” *Blockchain.info*, blockchain.info/charts/market-cap?scale=1×pan=all.
[16] McInnes, Leland et al. “Parameter Selection for HDBSCAN*.” *Hdbscan Docs*, 2016, hdbscan.readthedocs.io/en/latest/parameter_selection.html.
[17] McInnes, Leland et al. “Comparing Python Clustering Algorithms.” *Hdbscan Docs*, 2016, hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html.
[18] McInnes, Leland et al. “How HDBSCAN Works.” *Hdbscan Docs*, 2016, hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html.
[19] Meiklejohn, Sarah et al. “A Fistful of Bitcoins.” *Proceedings of the 2013 conference on Internet measurement conference - IMC '13*, 2013, doi:10.1145/2504730.2504747.
[20] Nakamoto, Satoshi. “Bitcoin: A Peer-to-Peer Electronic Cash System.” *Bitcoin.org*, 2008.
[21] Narayanan, Arvind et al. *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton, NJ, Princeton University Press, 2016.
[22] Prisco, Giulio. “Leaked Chainalysis Roadmap Angers Bitcoin Community.” *Bitcoin Magazine*, 28 Apr. 2015, bitcoinmagazine.com/articles/leaked-chainalysis-roadmap-angers-bitcoin-community-1430255443.
[23] Reid, Fergal, and Martin Harrigan. “An Analysis of Anonymity in the Bitcoin System.” 7 May 2012, link.springer.com/chapter/10.1007/978-1-4614-4139-7_10.
[24] Ron, Dorit, and Adi Shamir. “Quantitative Analysis of the Full Bitcoin Transaction Graph.” *Financial Cryptography and Data Security Lecture Notes in Computer Science*, 2013, pp. 6–24. doi:10.1007/978-3-642-39884-1_2.
[25] ShenTu, QingChun, and JianPing Yu. “Research on Anonymization and De-Anonymization in the Bitcoin System.” *ATR Defense Science & Technology Lab*, 27 Oct. 2015.
[26] Spagnuolo, Michele et al. “BitIodine: Extracting Intelligence from the Bitcoin Network.” *Financial Cryptography and Data Security Lecture Notes in Computer Science*, 2014, pp. 457–468. doi:10.1007/978-3-662-45472-5_29.
[27] “Transaction View About a bitcoin transaction.” blockchain.info/tx/6f09102cd8a2ed0f44701dbe728414ead35918c4bf9ab5168d5f8a77a0038656.
[28] “Transaction View information about a bitcoin transaction.” blockchain.info/tx/f6e23454b96869ac102dd07a253387430dc766b26622c247bc0e91dad7be4ba4.
[29] “Transaction View information about a bitcoin transaction.” blockchain.info/tx/dd64b0fe5e951b630be200d7132473fcfcb7e4ed35a4da18625fb48b95744dcd.

[30] Wong, Joon Ian, and Akshat Rathi. “Bitcoin might just be a plausible response to the “war on cash” declared by governments around the world.” *Quartz*, 7 Jan. 2017, qz.com/880460/could-bitcoin-be-the-solution-to-governments-war-on-cash/.

[31] Zhou, Yang et al. “Graph clustering based on structural/Attribute similarities.” *Proceedings of the VLDB Endowment*, vol. 2, no. 1, Jan. 2009, pp. 718–729. doi:10.14778/1687627.1687709.

9. Honor Code

This paper represents my own work in accordance with University Regulations.

Signed: Bharath Srivatsan