An Analysis of Anonymity in Bitcoin while Leveraging Social Networks

Carolyn Atterbury
catterbury@unm.edu
The University of New Mexico, Department of Computer Science
Albuquerque, New Mexico

Keira Haskins
wasp@unm.edu
The University of New Mexico, Department of Computer Science
Albuquerque, New Mexico

ABSTRACT

Bitcoin, a peer-to-peer digital currency introduced in 2009 by Satoshi Nakamoto, has since gained a large degree of popularity around the world. Bitcoin is a pseudo-anonymous system where miners add transactions to a global ledger, blockchain, by solving moderately difficult puzzles. The system itself was designed to allow users to maintain some degree of pseudonymity. However in recent years research has been done which has shown that there are a number of problems with this assumption: It is possible to use various techniques to de-anonymize users via public key reuse, multi-input transactions, and or interactions with other users and or entities. Our work seeks to de-anonymize users by considering social networks and those who use them. In this case we focus on identification of users who post information about their public keys on social networks, we then use this information to build user graphs which may then be linked to actual transactions in the blockchain.

CCS CONCEPTS

• **Cybersecurity and Cryptocurrency**;

KEYWORDS

Bitcoin, Cryptocurrency, Privacy, Anonymity

ACM Reference Format:
Carolyn Atterbury and Keira Haskins. 2019. An Analysis of Anonymity in Bitcoin while Leveraging Social Networks. In *Proceedings of UNM CS 591.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Privacy and anonymity have been a major goal in both the design and creation of Bitcoin as well as with every other alt-coin which has since followed. Although Bitcoin does provide some degree of pseudonymity; given that users of the system may create any number of new identities or public keys as they please, the system itself does not enforce the creation of new keys for each new transaction. (While although it is highly recommended). On top of this there is of course nothing to prevent users from ignoring various security concerns in general: As mentioned, reuse of keys, failing to adhere to general computer security practices; not using a VPN, dynamic DNS, secure browser/client, sharing their public key hashes on forums, various social media networks, etc. In this paper we specifically present a framework for linking identities between various social networks; Twitter, Reddit, BitcoinTalk, etc, and the Bitcoin blockchain.

2 RELATED WORK

Here, we present a brief overview of the work most closely related to ours.

Arvind Narayanan and Vitaly Shmatikov[5] demonstrate an algorithm for de-anonymizing social networks. Specifically they focus on the de-anonymization of users between Twitter and Flickr, or rather that of the users who have accounts on both social networks. One thing that we came across in their work discusses how much easier it is for advertisers to get access to users' data as it is now the case that most social media platforms charge some amount of money in order to get more and more detailed access to their API. They also talk about three different attack models:

1. Government-level agencies who have interests on a global level.
2. Advertisers and abusive marketing.
3. Private investigators and or stalkers who choose to target individual users.

Given their work, we realize that it should be possible for the first two attack models to be possible in Bitcoin, however for our work we focus on carrying out more of a targeted attack.

Chen Zhao and Yong Guan[2] focus on created a graph-based analysis on Bitcoin transactions in order to investigate transaction behaviors and currency flow involving Bitcoin users. They use bitcoind, which is the client source for Bitcoin, written in C++, in order to collect and generate a transaction graph. They go a step further and further refine this data using an address clustering algorithm in order to generate entity groups for addresses. Our work borrows the notion of address clustering, however as of the writing of this we have yet to fully implement algorithms for doing so in our framework.

Amitabh Saxena, et al.[1] present a framework for composite signatures in order to obscure the links between inputs and outputs. Their work in contrast to ours is instead concerned with enhancing anonymity in a system such as Bitcoin, whereas our work focuses on measuring anonymity between various networks including Bitcoin.

3 METHODOLOGY

At a high-level our general approach to conducting our research involves first gathering data from a variety of social media networks, forums, and the Bitcoin blockchain. Second, we use the given data to

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

UNM CS 591, May 2019, Albuquerque, NM, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. . . $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

construct user graphs centered around specific users which we have found to post hashes of their public keys for Bitcoin, at the same time we use the data from the blockchain to create a transaction graph which we will then use in combination with the user graphs to measure to what extent we are able to link users to public key hashes in the Bitcoin network.

3.1 Social Media APIs

The first method for collecting social network data that we look at is using APIs for Facebook, Twitter, and Instagram. We realize early on that that doing so will prove to be much more difficult than it would have been a few years ago. In 2018 both Twitter and Facebook changed their developer API requirements which severely restricted the access to their data. Twitter implemented a tiered payed program that makes it expensive for users to get access to their data. They maintained a free tier of access that allows developers to have access to data within the last 10 days, but only after they are accepted through an application process. Facebook implemented a much more rigorous application process that allows only established companies to have the ability to pay for their data. Since Facebook acquired Instagram, the same rules apply to Instagram as well. This has ultimately proved to be a fairly large impediment as it limits the scope of the data which we collect to a much smaller subset for Twitter. In querying the Twitter data, we were not able to identify any users that posted information about their public keys within the last 10 days. This pushed us to seek out data from other sources such as BitcoinTalk.

3.2 Web Scraping

After recognizing difficulty of gathering a substantial amount of data using the APIs for various social media networks, especially Twitter, Facebook, and Instagram, we decide to try various libraries and tools designed for web scraping in an attempt to gather data from other sources of social media such as Reddit, BitcoinTalk, and other forums. To start, we first take a look at using various tools which others have written; Scrapy, bitcointalk_scraper, and GrayHats/btctalkint, however this quickly turns out to be more difficult than originally anticipated as we run into various types of problems; not being able to put the data into JSON, and others either not working easily, and or not working as we need. From here we begin looking at libraries in Haskell such as TagSoup, Scalpel, HandsomeSoup, xml-to-json, among others.

3.3 JSON/HTML Parsing

For parsing JSON we use the Haskell library Aeson which took us a while to fully implement, however after doing so we have found it to be relatively simple to add additional functionality for other JSON key/value types. We are able to use this data in order to start producing user graphs based upon public key hashes which we have come across via both scraping and manual searching.

3.4 Bitcoin API/Blockchain

In an attempt to create the a user graph based on the Bitcoin blockchain transaction graph, we downloaded the entire Bitcoin blockchain onto Wheeler in CARC at UNM. The Bitcoin blockchain took up around 250 GB of space and it took a day to download in entirety. The blocks were downloaded in compressed blk*.dat files. Our original goal was to parse the blk*.dat files to generate the transaction graph, but we ran into difficulty finding a reliable parsing library to assist with the parsing. Eventually we found Bitcoinj, a java library that has block parsing capabilities along with many advanced features for interacting with the Bitcoin protocols. Unfortunately the tools necessary for building the Bitcoinj repository were not available on Wheeler. Due to lack of time we had to abandon this endeavor.

Instead of creating a user graph for the entire blockchain, we instead used Blockchain Explorer and its API endpoints to create a user graph that represents a small subset of the transactions in the Bitcoin ledger. Data from Blockchain Explorer was given in JSON format, so we had to parse the corresponding data to create a user graph representing a small subset of bitcoin transactions.

3.5 Graph Building

In order to compare the user graph in Bitcoin with a social network from a social media site, we needed two build two graphs. Let G_s be the graph representing the social network, and let G_b be the user graph from bitcoin. To build G_b, we are treating each public key in the bitcoin network as a separate user, even though we know this is not always the case. Due to time constraints, we were not able to implement more advanced clustering mechanisms for assigning a user to a group of public keys.

From Blockchain Explorer, we had a set of transactions that we used to create a user graph. Each transaction had a set of input addresses signifying who the bitcoin was being set from. Each transaction also had a set of output addresses signifying where the bitcoin was being sent to. Each of the unique input addresses and output addresses were added to our user graph G_b. For each input address i, and for each output address o, we created an edge i, o and added it to G_b. In G_b, each user shared an edge with another user if they both took part in a transaction.

To generate the social network graph G_s, we used data from the Twitter API. For a given query, the Twitter API returned a list of Tweets which each had an associated user id of the person who published the Tweet. In the metadata about each tweet, there was a field "user_mentions" that listed the users that were tagged in that particular tweet. These were users that were tagged in the message content of the tweet. We added each of the unique user ids from the Tweet publisher as individual nodes in G_s. Each of the unique user ids in the user mentions of each Tweet were also added as nodes in G_s. We added an edge between user id i and user id j if user i mentions user j in a post.

3.6 Public Key Clustering Algorithm

As users in the Bitcoin system can have any number of public keys, we developed a method for clustering public keys that are in the same clique. This method is based on the assumption that Bitcoin users will have multiple public keys, but will need to shuffle bitcoin between their active accounts in order to prepare for transactions with another party, or simply at an attempt to improve anonymity.

Our clustering algorithm is applied to an undirected graph $G = (V, E)$, and begins with an initial public key pk. We find the biggest clique $C = (V', E')$ in G, where $V' \subseteq V$ and $E' \subseteq E$, such that $|V'| \geq 2$, and $pk \in C$. We then replace C with a single node pk', such that every edge $u, v \in G$ where $u \in C$, and $v \notin C$, is replaced with v, pk'. Let the resulting graph be G'. The process repeats on G' and pk', until pk' is no longer in a clique of size 3. Since finding cliques in a graph is a NP-Complete problem, this clustering algorithm is also NP-Complete. It works well for small subsets of the bitcoin transaction graph, but it will not scale well.

We implemented the public key clustering algorithm in Python using the Networkx package for creating and modifying graphs. The Networkx library has a method $cliques\_containing\_node(G, pk)$ that we used to find all the cliques in G that contain the pk. The function terminates when the public key is not in any clique that is bigger than size 2.

```
import networkx as nx

def public_key_clustering(G,pk):
    all_cliques = nx.cliques_containing_node(G, pk)
    cliques = []

    for c in all_cliques:
        if (len(c) > 2):
            cliques.append(c)

    if (len(cliques) < 1):
        return G

    new_G = nx.Graph()

    for edge in G.edges():
        if (edge[0] not in cliques[0] and edge[1] not in
           cliques[0]):
            new_G.add_edge(edge[0], edge[1])

    for node in cliques[0]:
        neighbors = G.neighbors(node)
        for n in neighbors:
            if n not in cliques[0]:
                new_G.add_edge(pk, n)

    return cluster_cliques(new_G)
```

4 RESULTS AND DISCUSSION

Through web scraping BitcoinTalk, we identified many users that post their Bitcoin public key as part of their account profile. We identified a particularly active user that has an associated public key, which we will call user A. As Bitcoin Talk does not have a social media type of network structure, we used the Twitter API to query for Tweets involving users with the same username as user A on Bitcoin Talk. As it turns out, user A is an active member of the discussion on Bitcoin Talk, and also an active member of the Bitcoin discourse on Twitter. We were able to access 100 recent Tweets from Twitter that either mentioned user A or were created by user A. As seen in Figure 1 we were able to build a social media user graph, G_s, based on the activity of Tweets involving user A.

Using the public key from user A, we searched for transactions involving that public key. We could not query Blockchain Explorer's API for user A's public key directly, but we were able to get user A's transaction history from BlockChair. We used the transaction history information from BlockChair to query for specific transaction information through Blockchain Explorer's API. Through user A's transaction history, we were able to build up a user graph of the transactions involving user A, as seen in Figure 2.

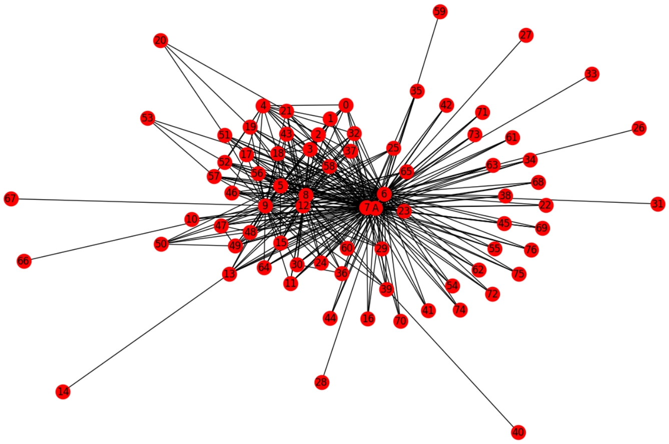

Figure 1: Social media user graph generated from Twitter data based on Tweets by and mentioning user A.

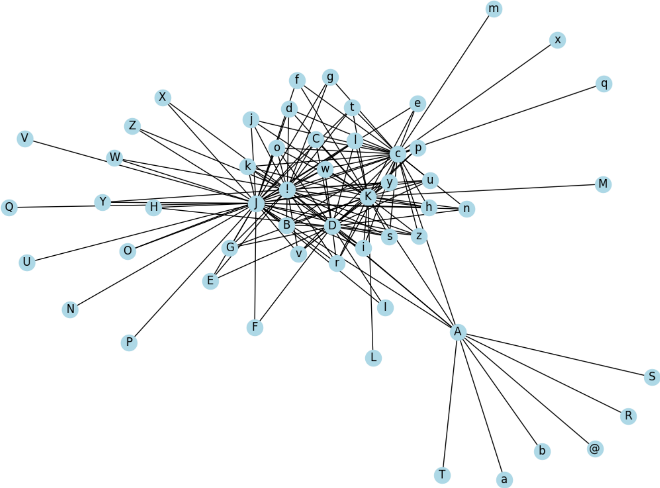

Figure 2: Bitcoin user graph based off of the transaction history from user A. The nodes in this graph are other public keys in bitcoin that either sent bitcoin to user A, or received bitcoin from user A.

As seen in Figure 2, there is a cluster of interconnected nodes in the center of the graph. After further analysis we noticed 17 size 3 cliques in G_b. Only one of the size 3 cliques included user A, and that can be seen in Figure 4. Since any user in the Bitcoin network can have any number of public keys, we applied our clique-based public-key clustering algorithm (described in section 3.6) to create a user graph that contracts all of the public keys that could belong to user A, into one node: A. The modified Bitcoin user graph G'_b can be seen in Figure 3.

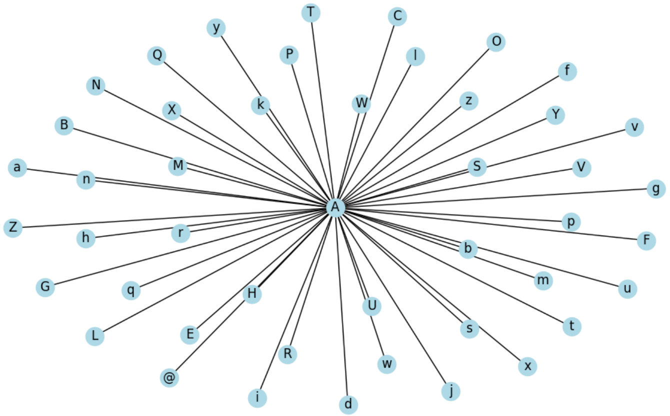

Figure 3: Bitcoin user graph of user A after the public key clustering algorithm has been applied. All of the cliques found in G_b have been consolidated into one node 'A'.

Now that we have a user graph G_b', we can compare it against the Twitter user graph, G_s. In G_s, the biggest clique was of size 9, and there were 4 of them that included user A. One of the cliques in G_s that includes user A can be seen in Figure 5. The fact that the Twitter data has multiple cliques of size 9 that all involve user A, is a good indication that this is a connected group of individuals that are all vocal in the bitcoin community. It seems likely that some of these other users would be associated to, or a part of, user A's transaction history. Between the 4 biggest cliques in G_s, we found that there were 14 distinct members.

In looking at the Bitcoin user graph G_s' in Figure 3, we can see that there are 45 nodes that were not clustered as part of user A. It is likely that some of the remaining nodes in G_s' still belong to user A. Since we were only looking at transactions involving user A's pk, we will have missed any clustering behavior that happens one transaction away from user A's pk, but that might still belong to user A. If we had more time with the project, a more robust user graph would be possible. In the case we have a more robust user graph, and assuming that we have nodes exiting user A in G_s' (Figure 3), we can assume that these are different users in the Bitcoin system.

Suppose there are a set B of n other users transacting with user A, that we can see after applying the user clustering algorithm. Suppose we have a set A of n users from Twitter, and a set B of m users in Bitcoin, we can come up estimate of the probability Pa_i, b_j of user $a_i \in A$ in Twitter having public key $b_j \in B$ in Bitcoin, assuming there is a one-to-one mapping between the two sets.

Theorem: Assuming there is a one-to-one mapping between set the Twitter users in set A and the Bitcoin users in set B, then the probability of a Twitter user a having the public key b is $P\left(a,b\right) = \frac{1}{m}$, if $m > n$.

Proof:

Let A be the set of Twitter Users, such that $|A| = n$. Let $a \in A$.

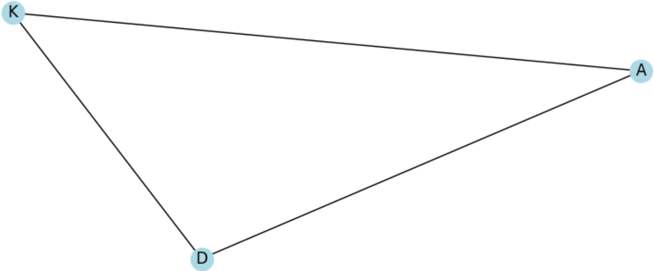

Figure 4: Clique including user A in the bitcoin user graph.

Let B be the set of Bitcoin Public Keys, such that $|B| = m$. Let $b \in B$. Suppose $m > n$.

We want to find $P\left(a,b\right)$, the probability of user a having the public key b.

$$P\left(a,b\right) = \frac{\text{The number of mappings where } f\left(a\right) = b}{\text{Total number of mappings}}$$

Let $F\left(A,B\right) = \{f : A \to B \text{ such that f is an injection}\}$.

We know that $|F\left(A,B\right)| = \binom{m}{n} \cdot n! = \frac{m!}{(m-n)!}$.

If we fix a and b such that $f\left(a\right) = b$,

then let $F\left(A', B'\right) = \{f' : A' \to B' | A' = A - \{a\}, B' = B - \{b\}\}$.

Since $|A - \{a\}| = m - 1$ and $|B - \{b\}| = n - 1$,

then $|F\left(A', B'\right)| = \binom{m-1}{n-1} \cdot \left(n-1\right)! = \frac{(m-1)!}{(m-n)!}$.

So,

$$\begin{aligned} P\left(a,b\right) &= \frac{\text{The number of mappings where } f\left(a\right) = b}{\text{Total number of mappings}} \\ &= \frac{|F\left(A',B'\right)|}{|F\left(A,B\right)|} \\ &= \frac{\left(m-1\right)!}{\left(m-n\right)!} \Big/ \frac{m!}{\left(m-n\right)!} \\ &= \frac{\left(m-1\right)!}{m!} \\ &= \frac{1}{m} \end{aligned}$$

5 CONCLUSION

We presented a framework for analyzing privacy and anonymity between various social media networks and Bitcoin. First, we gather data for specific users from a variety of social networks and forums using both applications and libraries for web-scraping, and APIs from individual networks. As of this point in time we have found it to be overall much more difficult to gather data from most of the larger social media platforms such as Facebook, Instagram, and Twitter, as most of their data now requires paid services. Second,

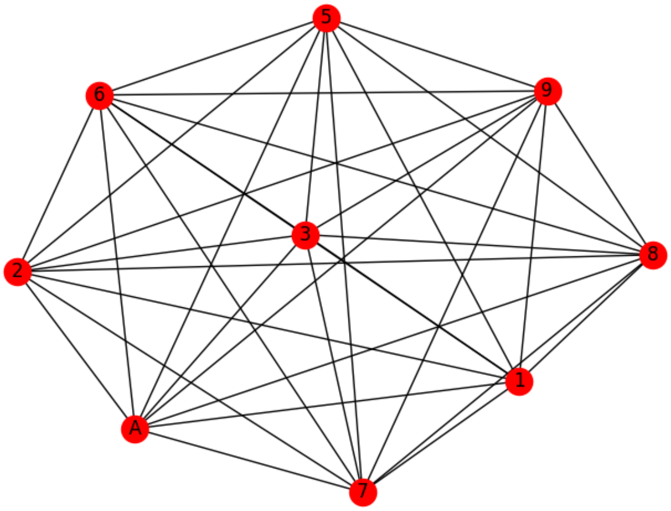

Figure 5: Clique including user *A* in the Twitter user graph. This is one out of 4 different cliques of size 9 that included user *A*.

we implement various algorithms and libraries in JavaScript and Python for generating both user graphs from the data gathered in the first step, as well as a transaction graph from the data gathered from the Bitcoin blockchain. After building the transaction graph, we are able to use a clustering algorithm to group the public keys belonging to one user, user *A*. We were able to compare user *A*'s transaction history in Bitcoin against user *A*'s Twitter user graph, in order to create a metric for which we may attempt to determine the degree to which we may link users in a social network, via public Bitcoin key hashes, (both the associated owners of the public keys, as well as users which are closely linked to them), to public keys and transactions in the Bitcoin blockchain.

REFERENCES

[1] Amitabh Saxena, J. M., and Dhar, A. Increasing anonymity in bitcoin. In *IFCA/Springer-Verlag Berlin Heidelberg 2014 R. BÃűhme et al. (Eds.): FC 2014 Workshops, LNCS 8438, pp. 122âĂŞ139, 2014. DOI: 10.1007/978-3-662-44774-1 9* (October 2014).

[2] Chen Zhao, Y. G. A graph-based investigation of bitcoin transactions. In *Gilbert Peterson; Sujeet Shenoi. 11th IFIP International Conference on Digital Forensics (DF), Jan 2015, Orlando, FL, United States. IFIP Advances in Information and Communication Technology, AICT-462, pp.79-95, 2015, Advances in Digital Forensics XI. <10.1007/978-3-319-24123-4_5>. <hal- 01449078* (January 2015).

[3] DuPont, J., and Squicciarini, A. C. Toward de-anonymizing bitcoin by mapping users location. In *In Proceedings of the 5th ACM Conference on Data and Application Security and Privacy (CODASPY '15). ACM, New York, NY, USA, 139-141. DOI: https://doi.org/10.1145/2699026.2699128* (March 2015).

[4] Michael Fleder, M. S. K., and Pillai, S. Bitcoin transaction graph analysis. In *arXiv:1502.01657 [cs.CR]* (February 2015).

[5] Narayanan, A., and Shmatikov, V. De-anonymizing social networks. *CoRR abs/0903.3276* (2009).

[6] Sarah Meiklejohn, Marjori Pomarole, G. J. K. L. D. M. G. M. V., and Savage, S. A fistful of bitcoins: characterizing payments among men with no names. In *IMC '13 Proceedings of the 2013 conference on Internet measurement conference* (October 2013).

[7] Sweeney, L. k-anonymity: A model for protecting privacy. *IEEE Security and Privacy 1998. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10 (5), 2002; 557-570.* (2002).