Discovering Bitcoin Mixing Using Anomaly Detection

Mario Alfonso Prado-Romero[1(✉)], Christian Doerr[2], and Andrés Gago-Alonso[1]

[1] Advanced Technologies Application Center (CENATAV), 7a # 21406, Rpto. Siboney, Playa, 12200 Havana, Cuba
{mprado,agago}@cenatav.co.cu
[2] Delft University of Technology (TU Delft), Mekelweg 4, 2628CD Delft, The Netherlands
c.doerr@tudelft.nl

Abstract. Bitcoin is a peer-to-peer electronic currency system which has increased in popularity in recent years, having a market capitalization of billions of dollars. Due to the alleged anonymity of the Bitcoin ecosystem, it has attracted the attention of criminals. Mixing services are intended to provide further anonymity to the Bitcoin network, making it impossible to link the sender of some money with the receiver. These services can be used for money laundering or to finance terrorist groups without being detected. We propose to model the Bitcoin network as a social network and to use community anomaly detection to discover mixing accounts. Furthermore, we present the first technique for detecting Bitcoin accounts associated to money mixing, and demonstrate our proposal effectiveness on real data, using known mixing accounts.

Keywords: Bitcoin · Bitcoin mixing · Anomaly detection

1 Introduction

Bitcoin is an electronic currency designed to be decentralized and to provide anonymity to users [1]. In the Bitcoin network, accounts are identified by public keys and each user can own multiple accounts. All transactions in the Bitcoin system are stored in a public ledger called the blockchain which is the mechanism used by the system to prevent double spending. Low transaction fees, easiness to create accounts and anonymity have all influenced in the fast increase in Bitcoin popularity with a current market capitalization of around 14 billion dollars in 2017[1]. Despite the alleged privacy, there is an understanding amongst Bitcoin more technical users that anonymity is not a primary design goal of the system [2]. It is possible to use the blockchain to trace money from one user to another, and it has been demonstrated that the identity of the users can be uncovered using information external to the Bitcoin network.

[1] https://blockchain.info/charts/market-cap.

M. Mendoza and S. Velastín (Eds.): CIARP 2017, LNCS 10657, pp. 534–541, 2018.
https://doi.org/10.1007/978-3-319-75193-1_64

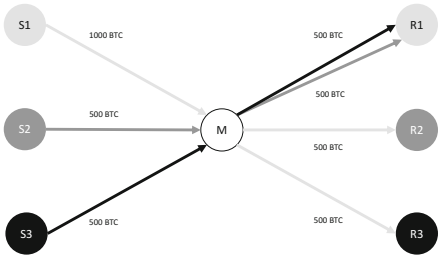

Fig. 1. Bitcoin mixing example

In recent years, many services intended to provide further transaction anonymization have emerged such as BitLaundry, BitFog and the Send Shared functionality of Blockchain.info. These are known as Mixing Services and some of them routinely handle the equivalent of 6-digit dollar amounts [3]. The idea behind Mixing Services is to be an intermediary in user transactions. They take the money of many senders and then for each one, send the desired amount of money to the receiver using money coming from other senders. The goal of Bitcoin mixing is to make it impossible to link the sender with the actual receiver of the money. The use of these services imply some inconveniences for the users, like a delay of many days in the transactions, the payment of an extra fee for the operation, and the risk of having their money stolen by a fraudulent mixing service. Bitcoin has always attracted the attention of the criminal world due to its decentralized nature [4] and Mixing Services can be used for money laundering or to finance terrorist groups without being detected.

Figure 1, displays an example of mixing where senders $S1$, $S2$ and $S3$ want to transfer money to the receivers $R1$, $R2$ and $R3$ respectively. To avoid being related to the receivers, the senders use a mixing service M which transfers the desired amount to $R2$ and $R3$ using the money from $S1$, and the bitcoins from $S2$ and $S3$ are sent to $R1$. This is a basic example, actual Mixing Services use many evasion techniques to avoid money tracing. Two of the most used tactics for this end are delaying transactions, to avoid be linked by time, and splitting the money into small transactions, to make impossible to relate the transactions by amount. Also, a common practice is to use many accounts for moving the money before performing the actual mixing.

Tracing money through mixing services has been demonstrated to be an extremely difficult task in most cases [3]. In the other hand, the discovery of Bitcoin mixing accounts is still possible and worthwhile in its own. Mixing Services create new accounts regularly, and the whole set of accounts belonging to them is not known by users. While tracing is capable of discovering some mixing accounts used by a particular person of interest, other accounts from the service remain unknown. Mixing detection can identify which accounts from the network are related to these services. Once the mixing accounts are discovered, users related to them could be identified and further analyzed to determine if they are

involved in criminal activities. Existing works for detecting malicious activities in Bitcoin [3,5–7] are not focused in identifying unknown mixing accounts.

Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior [8]. Due to the inconveniences of mixing sites, they are not used by the majority of people in the Bitcoin network, for this reason mixing accounts and its users are an anomaly from the perspective of the network topology. We take advantage of this property and use anomaly detection to identify mixing accounts. Most existing anomaly detection techniques are for vector data [9] and cannot be used for this task. From the techniques designed to work in graphs [10], only a small number considers communities of elements [11–14] and they are not designed to identify bitcoin mixing accounts. The main contributions of our work are:

- **We tackle the problem of Bitcoin mixing detection:** Many works have focused in studies of anonymity and criminal activities in Bitcoin. To the best of our knowledge this is the first work focused on detecting mixing accounts, which can be helpful to uncover potential money laundering.
- **We discover mixing accounts using community outlier detection:** We propose to build the Bitcoin user network where members are structured in communities. Then, we use community anomaly detection to identify mixing accounts. Furthermore, we present the first algorithm for this purpose.
- **We validate our approach using real data:** We test our algorithm in real data and the results demonstrate the effectiveness of our proposed approach for identifying mixing accounts.

The remainder of this paper is structured as follows: In Sect. 2, our proposal is presented. In Sect. 3, the effectiveness of our proposal is demonstrated on real data and the results are analyzed. Finally, in Sect. 4, the work is concluded and some open challenges are discussed.

2 Discovering Bitcoin Mixing

People normally have a tendency to use the services they known and like, and to exchange money with the same group of known people. We believe this behavior is also present in the Bitcoin network.

Bitcoin transactions are linked to each other, and can be naturally modeled as a network where the transactions will be the vertices, and the money stream among them will be represented by the edges. This graph, called the transaction network, can be used to trace money or to identify double-spending, but it is not very useful to recognize user behavior. Also the user behavior is spread across many accounts. Due to this, our first step is to merge accounts into user identities to build a user network following the process described in [2].

The user network behaves as a social network where users are organized in communities. The idea of mixing is to merge money from different users, and for each sender, give the desired amount of money to the target receiver using the

money coming from another sender. For this reason, mixing violates the community structure of the network relating users that have nothing in common. As we mention in the introduction, the people using mixing sites are a minority of Bitcoin users. We affirm that users having many more inter-community connections compared to the rest of users belonging to its same community are probably mixing sites. We propose a new algorithm named InterScore designed to identify this kind of outliers. Due to unavailability of labeled data and the difference between user groups, our algorithm finds the communities in the network and analyzes each element in its community in an unsupervised fashion. As result, it returns an outlier ranking of Bitcoin users.

Definition 1 (Outlier ranking). *An outlier ranking from a graph G is a set $R = \{(v, r)|v \in V, r \in [0, 1]\}$ of tuples, each one containing a vertex from G and its outlierness score.*

The input of our algorithm is a user graph G_U. In a first stage, the Louvain community detection method [15] is used on G_U to identify groups of related users, returning a clustering C of vertices from G_U. Any state-of-the-art graph clustering algorithm could be used in this stage. The Louvain method was selected based mainly in its performance and applicability in large graphs.

In the second stage, our algorithm iterates over each community $C_i \in C$ and for each vertex calculates the number of inter-community links it has, using a function $l : V \to \mathbb{R}$. Then, for each community C_i is calculated the mean difference among the number of inter-community links from its elements as defined below:

$$IMD(C_i) = \frac{\sum_{v_j \in C_i} \sum_{v_k \in C_i, v_j \neq v_k} |l(v_j) - l(v_k)|}{|C_i|} \tag{1}$$

Once the inter-community links mean difference is calculated for each community, our algorithm iterates over the elements of each C_i and determines its anomaly score using the next function:

$$r(v, C_i) = \frac{\sum_{u \in C_i, u \neq v} d(v, u, C_i)}{|C_i|} \tag{2}$$

where $d : V \times V \times 2^V \to \{0, 1\}$ is a function that determines if the inter-community links difference between two vertices is greater than its community mean. The function is defined as below:

$$d(v, u, C_i) = \begin{cases} 0 & |l(v) - l(u)| \leq IMD(C_i), \\ 1 & |l(v) - l(u)| > IMD(C_i) \end{cases} \tag{3}$$

Intuitively, the score function measures with how percent of the community the user has a difference in the amount of inter-community links greater than the mean difference for that community. This function adaptively ranks users outlierness according to their context, and detects anomalies that cannot be identified from a global point of view. In the Algorithm 1, the steps of the InterScore method can be observed in more detail.

Algorithm 1. InterScore

Input: A users network G_U
Output: An outlier ranking R of the vertices from G_U

$R \leftarrow \emptyset$
$C \leftarrow$ Clustering(G_U)
foreach *community* $C_i \in C$ **do**
  $m \leftarrow IMD(C_i)$
  foreach *user* $v \in C_i$ **do**
    $r_v \leftarrow 0$
    foreach *user* $u \in C_i$ *with* $u \neq v$ **do**
      if $|l(u) - l(v)| > m$ **then**
        $r_v \leftarrow r_v + 1$
      end
    end
    $r_v \leftarrow \frac{r_v}{|C_i|}$
    $R \leftarrow R \cup \{r_v\}$
  end
end
return R

The InterScore algorithm has two fundamental stages, the community detection stage and the anomaly detection stage. In the former, we use the Louvain algorithm whose exact computational complexity is unknown because it is an heuristic, but the authors said it appears to be $O(V \log(V))$ based in the experiments. The outlierness score used in the last stage has a $O(V^2)$ complexity. As a result, the InterScore algorithm has a computational complexity of $O(V^2)$. It is important to mention that the Bitcoin network is sparse, so calculate the inter-community links of all vertices is less expensive than $O(V^2)$ in practice. Also, the anomaly score is calculated only in relation to the users in the same community, for this reason is less expensive when the number of communities is higher.

3 Experiments

We use two subsets from the blockchain in our analysis, the first one contains all transactions ranging from 2012-09-01 to 2012-10-01, and the second one, all transactions from 2013-04-19 to 2013-05-31. The former subset will be referred as the 2012 data set and the later as 2013 data set. It is important to note that the algorithms for Bitcoin mixing detection should be capable of finding interesting results using only a subset of the data because the size of Bitcoin network growths exponentially.

As ground truth we use six accounts, identified in [3], as involved in money mixing. The authors found these accounts while trying to trace money through three known mixing services BitLaundry, BitFog and the Send Shared functionality of Blockchain.org. The problem of tracing money through this services is complex and the authors cannot guarantee all six transactions are related to the mentioned mixing services.

We propose two different methods for determining the inter-community links of a user. The first one will be called Relative Inter Links and consist in for each neighbor u of the analyzed node v, adds 1 if the community of u is different from that of v. The second will be called Total Inter Links and it is a very similar process, but instead of increase 1, increases in the number of transactions between the users v and u. The former method is less sensitive to misclassified users during the community detection stage, but the later is better in identifying a common practice among Mixing Services of dividing transactions in smaller ones.

The user networks for the 2012 and 2013 data set where built. The result was a graph with $412,330$ vertices and $885,808$ edges in the former, and with $942,204$ vertices and $2,835,807$ edges in the later. These values evidenced the sparse nature of the user network.

The results of our algorithm can be observed in Table 1. The advantages of our proposal in identifying candidate mixing accounts, for being further analyzed, can be appreciated. The number of elements identified as mixing services in both data sets is less than a 0.6% of total users. Furthermore all known mixing accounts are identified by our algorithm demonstrating the effectiveness of our approach.

Table 1. InterScore performance

Database	Interlink counting function	Total users	Discovered users	Known mixing users	Known mixing users identified
2012	Relative Inter Links	412330	100	2	2
2012	Total Inter Links	412330	133	2	2
2013	Relative Inter Links	942204	2114	4	4
2013	Total Inter Links	942204	5422	4	4

The ground truth used in this section are four transactions and two accounts reported in [3] as involved in money mixing. As a threshold in the outlier ranking, we use the outlierness score of the element from the ground truth with the lowest score. It is important to mention that a transaction could have many sender and receiver accounts. We say our algorithm detects a known mixing user if it identifies a user that owns at least one account involved in a transaction from the ground truth. Not all accounts involved in mixing transactions are identified as anomalous. Some of them are accounts used only to move money and made it harder to trace. Also, some new mixing accounts that at the moment of the analysis have been used only by users in the same community are hard to identify. Despite the previous cases, at least one account involved in each transaction for the ground truth was identified by our algorithm.

It is interesting to analyze the difference in the number of detected elements depending on the function used to count the inter-community links. The function that counts the total number of links detects more elements, especially in

the 2013 data set. This increase can be the result of some elements misclassified by the community detection algorithm due to the increase in the complexity of the network. Also, could indicate an increase in the activity of mixing services. The last explanation is possible due to an increase in the number of Bitcoin users and the fact that the number of detected elements greatly increased using both functions. Additionally, the function that counts the total number of inter-community links better captures the common mixing sites behavior of dividing big transactions into many smaller ones, being capable of detecting more mixing accounts. It should be mentioned that we cannot ensure that all detected elements are mixing accounts, but cannot ensure neither that those of them which are not known mixing accounts are normal users.

In Table 2, we show the scores assigned by our algorithm to each account from the ground truth. In general, these accounts get very high scores. It is curious that, in the 2012 data set, the accounts get the same score no matter the link counting function used. In the 2013 data set the results vary accordingly to the function used. This behavior indicates that in 2012 the Mixing Services did not split the senders money into smaller transactions, a practice that seems common in 2013. An increase in the complexity of Mixing Services behavior could be an explanation to the difference in the number of detected elements we got using different link counting functions in 2013 data set.

Table 2. Ground truth accounts outlierness score

Account key	Outlierness score (total)	Outlierness score (relative)	Database
1E8QctAG6oeM6sw9tZccxvUPRg9dhf9VDm	0.999972676849	0.999972676849	2012
1Bw7ohts4BHVPAMBiSAqWeELjTUV1ZY87g	0.999977892736	0.999977892736	2012
13udyfBcdA2PUDCFM69VYDEHRRFnqkjEkx	0.9996	0.999266666667	2013
1MsmThtteKPu6fWxwn2SMDEnmJex3vKSBk	0.996448195994	0.999341002258	2013
152Yd71xMDzVntyD86xWgag5jfeNmyhMyS	0.999980617713	0.999995154428	2013
1KdPv6GWpg6eoj6cxcV65uc1NwufvhtGGQ	0.999995154428	0.999995154428	2013

An interesting result is that we identify as anomalous accounts belonging to all known mixing transactions. The approach used by [3] was focused in trace money through mixing sites, and the fact that we find these same elements using a different approach is indicative of the association of these transactions with Bitcoin mixing.

4 Conclusions

We discussed the problem concerning the detection of mixing accounts in the Bitcoin network. We modeled the Bitcoin user network as a social network where members are associated in communities and mixing sites behave as community anomalies. Furthermore, we proposed the first algorithm to detect Bitcoin mixing accounts and demonstrated its effectiveness on real data.

We will focus on some challenges in future work. First, our algorithm can be naturally parallelized to increase the performance. Also, there is information about the direction of transactions and about the accounts that could be interesting to include in our method analysis. Finally, it is important to mention that the same idea used for detecting Bitcoin mixing can be used to identify spammers or even bots in botnets, being interesting domains for future work.

References

1. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008). http://www.cryptovest.co.uk/resources/Bitcoin%20paper%20Original.pdf
2. Reid, F., Harrigan, M.: An analysis of anonymity in the Bitcoin system. In: Altshuler, Y., Elovici, Y., Cremers, A., Aharony, N., Pentland, A. (eds.) Security and privacy in social networks, pp. 197–223. Springer, Heidelberg (2013). https://doi.org/10.1007/978-1-4614-4139-7_10
3. Möser, M., Böhme, R., Breuker, D.: An inquiry into money laundering tools in the Bitcoin ecosystem. In: eCrime Researchers Summit (eCRS) (2013)
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 (2013)
5. Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J.A., Felten, E.W.: Sok: research perspectives and challenges for Bitcoin and cryptocurrencies. In: 2015 IEEE Symposium on Security and Privacy (SP) (2015)
6. Möser, M., Böhme, R., Breuker, D.: Towards risk scoring of Bitcoin transactions. In: International Conference on Financial Cryptography and Data Security (2014)
7. Spagnuolo, M., Maggi, F., Zanero, S.: BitIodine: extracting intelligence from the bitcoin network. In: Christin, N., Safavi-Naini, R. (eds.) FC 2014. LNCS, vol. 8437, pp. 457–468. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45472-5_29
8. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. (CSUR) **41**, 15 (2009)
9. Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artif. Intell. Rev. **22**, 85–126 (2004)
10. Akoglu, L., Tong, H., Koutra, D.: Graph based anomaly detection and description: a survey. Data Min. Knowl. Disc. **29**, 626–688 (2015)
11. Gao, J., Liang, F., Fan, W., Wang, C., Sun, Y., Han, J.: On community outliers and their efficient detection in information networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2010)
12. Müller, E., Iglesias Sánchez, P., Mülle, Y., Böhm, K.: Ranking outlier nodes in subspaces of attributed graphs. In: 2013 IEEE 29th International Conference on Data Engineering Data Engineering Workshops (ICDEW) (2013)
13. Perozzi, B., Akoglu, L., Iglesias Sánchez, P., Müller, E.: Focused clustering and outlier detection in large attributed graphs. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2014)
14. Prado-Romero, M.A., Gago-Alonso, A.: Detecting contextual collective anomalies at a glance. In: Proceedings of the 23rd International Conference on Pattern Recognition (ICPR) (2016)
15. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech: Theory Exp. **2008**, P10008 (2008)