Chapter 5

A GRAPH-BASED INVESTIGATION OF BITCOIN TRANSACTIONS

Chen Zhao and Yong Guan

Abstract The Bitcoin global cryptocurrency system has been the subject of several criminal cases. The Bitcoin network is a peer-to-peer system that has participants from all over the Internet. The Bitcoin protocol requires participating nodes to retain and update all transaction records; this ensures that all Bitcoin activities are accessible from a consistent transaction history database. This chapter describes a graph-based method for analyzing the identity clustering and currency flow properties of Bitcoin transactions. The method, which is scalable to large Bitcoin graphs, focuses on transactions relevant to criminal cases such as Mt. Gox. The analysis, which is performed on two years of Bitcoin transaction data, provides insights into the nature of anonymity provided by Bitcoin and how currency flows between selected users and communities of users.

Keywords: Cryptocurrency, Bitcoin, graph, investigation

1. Introduction

Bitcoin is by far the most widely used online cryptocurrency. The most intriguing distinction between virtual cryptocurrency systems such as Bitcoin and traditional online transactions is the decentralized semantics [1, 7]. Bitcoin is a highly-scalable peer-to-peer network that has no responsible authority or central management; it is accessible to anyone with an Internet connection.

As a self-complete economic system, Bitcoin provides two major functionalities. First, it supports coin generation as the source of Bitcoin currency. The Bitcoin protocol features a computationally-intensive "coin mining" process that brute forces the hash of new block data onto the Bitcoin blockchain. The network accepts a new block if the newly mined block hash is below a specific threshold. Second, Bitcoin makes it feasible

© IFIP International Federation for Information Processing 2015
G. Peterson, S. Shenoi (Eds.): Advances in Digital Forensics XI, IFIP AICT 462, pp. 79–95, 2015.
DOI: 10.1007/978-3-319-24123-4_5

to conduct global transactions. A valid Bitcoin transaction is broadcast in a relay manner to the network, after which it is accepted by Bitcoin miners and written onto the blockchain.

The Bitcoin currency exists in a different form from paper money or online bank balances. No user balance is recorded in the blockchain. Every transaction is a record of a sum of money paid from a source to a destination. Bitcoin transactions are identified and authenticated based on Bitcoin addresses, which are hashes generated from user public/private key pairs [2] that cannot be manipulated without the correct private keys.

A Bitcoin address is generated based on a randomly selected key and is the only unique way of identification in the system. Due to the random property of Bitcoin addresses and the extremely low probability of key collisions in the enormous elliptic curve digital signature algorithm (ECDSA) key space [8], the number of Bitcoin addresses grows rapidly because hundreds of different addresses can be generated for different transactions without address reuse. This address generation scheme provides adequate anonymity for Bitcoin transactions, making it difficult to track Bitcoin transactions conducted by real-world entities.

This chapter focuses on the transaction component of the Bitcoin system in an attempt to construct a low-cost transaction data analysis model and analytical approaches that support forensic investigations of Bitcoin transactions and relationships between Bitcoin users. The results reveal that different Bitcoin addresses belonging to the same entity can be effectively clustered to reduce the forensic effort required to track Bitcoin users involved in criminal activities. The accompanying graph-based analysis of Bitcoin transactions helps discern transaction behaviors and currency flows involving Bitcoin users. The method can expedite forensic investigations of criminal cases such as Mt. Gox that involve double-spending, theft, money laundering or fraudulent transactions.

2. Bitcoin System Overview

In the Bitcoin system, a single transaction is defined as a single record of a one-time payment. It is a proof of payment of a specific amount of Bitcoins (BTC) from a payer address to a receiver address. A transaction payer spends money by referring to payments received from other entities. Bitcoin has no notion of an account balance – payments are always about spending previous payments received from other entities.

A single transaction has two major parts: (i) input list; and (ii) output list. Each input includes information about previous transactions and

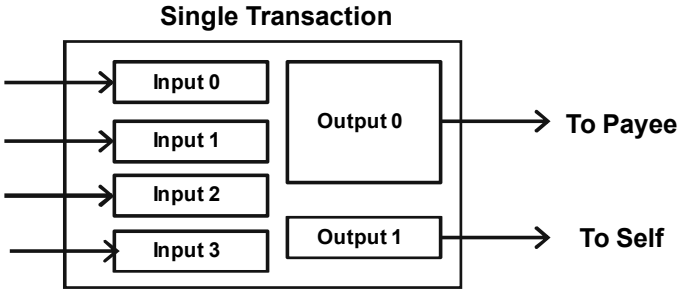


Figure 1. Transaction with change.

signature verification information. A transaction output denotes the amount of money paid and the address information of the recipient. A Bitcoin transaction is uniquely identified by a transaction ID defined as the hash value of the transaction.

A single transaction may contain multiple previous transactions and multiple recipient addresses. If the sum of the input values is larger than the amount that is to be paid, the payer designates a new address for the remaining change as shown in Figure 1. As mentioned in the previous section, a Bitcoin address is generated from a private/public key pair, so an entity cannot claim Bitcoin payments designated to other entities unless he/she owns the corresponding private keys (without which valid signatures cannot be created). In general, peers in the Bitcoin network do not accept illegal transactions.

```
{"inputs":
[{"value": 2.5, "address": "13vjvKHDeFhtwRGseVkLF1eXA7ZrJA2Uo9"},
{"value": 1.0092427, "address": "1EyH7htSWjSyuXgofuE9gKX5Sr4yEyDcD2"}],
"blocktime": 1389490523,
"outputs":
[{"value": 3.0, "address": "1Q5ztGyLj7KxL2rs7bmVcKYDfTYmpQs5bo"},
{"value": 0.5090427, "address": "1G7HKqqTUCrTpMDEYV5SdzcarFPhKKHb12"}]
```

Figure 2. Single transaction.

All Bitcoin transactions are stored in the blockchain that consists of a sequence of individual blocks. Blocks are for both storage as well as coin mining. Other than transaction records as in Figure 2, each block also contains other data fields, including a timestamp, last block hash and nonce. Mining Bitcoins involves the brute force computation of the nonce in the new block based on the current chain's latest block so that the new block hash value is equal to some value within a specified threshold.

```
{"inputs":
[{"address": "coinbase"}],
"blocktime": 1354133545,
"blockhash": "0000000000000498e426effa08fc54070cd7ca70e80206f8763e7ceaa
              ab734bc",
"outputs":,[{"value": 25.0512, "address":"1811f7UUQAkAejj11dU5cVtKUSTfo
                                          SVzdm"}]}
```


Figure 3. Coinbase transaction.

A Coinbase transaction has only one input. Previous transactions are not included in the Coinbase input because it is an indication of currency generation instead of a payment received from elsewhere. The input simply consists of a "coinbase" parameter without any authentication information (Figure 3).

A Bitcoin transaction is not confirmed until it is committed by a specific miner and issued onto the blockchain. Since this work focuses on the relationships between addresses, a Bitcoin transaction is considered to be a verified record containing addresses, payment amount and confirmation time.

Figures 2 and 3 show the main content of Bitcoin transactions in JSON notation. Figure 3 shows an example of Bitcoin generation for which there is no incoming address. It indicates the creation of a new block with the hash value shown. In the transaction, 25.0512 BTC (as one coin unit) was mined and it belongs to the output address in the output array. This single transaction is also stored in the blockchain with the hash value shown.

3. Transaction Data Collection

The Bitcoin dataset used in this study comprises blockchain data downloaded using the compiled full Bitcoin client source `bitcoind` in C++ [4]. The dataset is genuine and its integrity is assured by the cryptographic schemes used by the Bitcoin system. Transaction and block data were queried with `getrawtransaction` and `getblock` using the command line RPC tool provided by `bitcoind`. Alternatively, the data can be retrieved by directly invoking the `getrawtransaction` and `getblock` methods from the source. Both methods are globally visible in files `rpcrawtransaction.cpp` and `rpcblockchain.cpp`. Single transactions in the JSON format were parsed and written into a file as shown in Figures 2 and 3.

The dataset contains confirmed Bitcoin transactions. The transactions were collected from the blockchain maintained by the Bitcoin system from block 210,000 to block 314,700 corresponding to block creation

times from `15:24:38 2012-11-28` to `10:45:14 2014-08-09`. During this 20-month period, a total number of 34,839,029 Bitcoin transactions were successfully released and globally confirmed. A total of 35,770,360 distinct Bitcoin addresses were involved in the transactions.

4. Graph-Based Bitcoin Transaction Analysis

The graph-based transaction analysis involved three main steps. The first step was to parse the transactions in the dataset into the input-output based format shown in Figure 2. The addresses involved in the dataset were divided into groups, each group of addresses corresponding to a single Bitcoin entity. Such an entity could be a single individual, a group of Bitcoin users who share a Bitcoin wallet (private key) or Bitcoin miners who cooperate in mining activities.

The second step was to convert the transaction data into address-based graph notation. The goal was to clearly display the transaction flow between groups of addresses (each group corresponding to a single entity). The graph was designed so that each node corresponded to a single group. A transaction sent from one of the addresses in a group was drawn as an outward edge and a transaction paid to an address in a group was drawn as an incoming edge.

The third step was to answer some key questions: How efficient is it to track currency flow using an address graph? How can activities such as money laundering and large-scale Bitcoin theft or fraud be identified in a graph? If such activities are found to exist in a graph, how can evidence about Bitcoin entities engaged in the activities be collected and presented.

The remainder of this section describes the three steps and attempts to answer the key questions.

4.1 Address Clustering

The goal is to use a fast algorithm to check and group all the addresses into distinct sets as precisely as possible. Thus, given an arbitrary Bitcoin address, it would be possible to identify the entity group to which the address belongs. Meiklejohn et al. [5] have developed an algorithm that characterizes Bitcoin transactions into usage categories. They also proposed a heuristic that different Bitcoin addresses in the input of a single transaction are guaranteed to be associated with a single entity. This is because the Bitcoin transaction signature scheme does not permit an entity to spend a transaction that was not previously paid to the entity. Therefore, multiple input addresses in a transaction are a strong

indication that the entity that issued the transaction actually owns or shares the private keys corresponding to all the addresses.

In the case of Coinbase transactions discussed in Section 2, only one output address is present in the transaction output (Figure 3), so this address alone receives all the mining rewards. However, multiple addresses are allowed in Bitcoin transactions, including Coinbase transactions. Bitcoin mining intrinsically supports collaboration so that mining pools formed by multiple nodes can participate in mining efforts and the addresses in the pool simply share the rewards. For these reasons, the output addresses in a given Coinbase transaction are always pool-related. Thus, Coinbase transactions can help identify entities that cooperate. In this work, addresses belonging to entities in a given mining pool are treated as an atomic entity.

Another hint with regard to address grouping is the support for change shown in Figure 2, where a payer sends the change amount to his/her own address called the change address. Generally a change address is a one-time generated address for the sole purpose of receiving one's own money according to the implementation by the standard Bitcoin client software Bitcoin-Qt [4]. Using this property, it is possible to be a bit aggressive and assume that there are many one-time change addresses. One necessary condition for an output address to be classified as a change address is that it is not used as output ever again in the dataset. Along with some other conditions summarized in [10], it is possible to identify change addresses with low false-positive rates.

Based on the discussion above, the following three rules are specified:

1. All input addresses in the same transaction must be in the same set.

2. All output addresses in the same Coinbase transaction must be in the same set.

3. An output address that is only used as output once is placed in the same set as its transaction input addresses if the following conditions are met:

   - The output address is not the only address in the transaction output (a payment is not valid if it only sends change).
   - The output address does not exist in the transaction input list (this prevents the rare case of a self-loop).
   - None of the remaining output addresses in the transaction output list is used as an output only once (while a change address probably exists in this case, it is not possible to tell which address in the output list is used to receive the change).

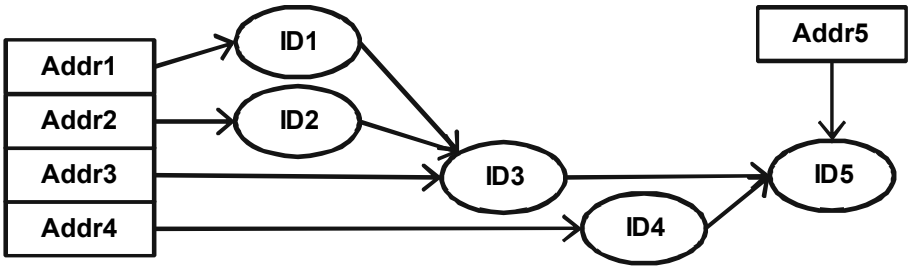


Figure 4. Conceptual graph of address grouping.

The Bitcoin dataset classification effort is substantial – identifying and rearranging nearly 36 million addresses, each of them a string of length 26-34 bytes. The problem arises when iterating all the addresses in each transaction and assigning them to the correct sets so that addresses always belong to the same entity if they are in the same set. The process involves set union operations: whenever an address set intersects with two or more existing sets, it is necessary to union all the sets to create a new atomic set. Unless hash tables are used, the worst case address access time complexity is $O(N^2)$ and the memory consumption is unacceptable. Therefore, instead of using this intuitive process, a much more efficient algorithm was employed to obtain the same classification results.

An open-source database developed by Google with a lightweight on-disk key-value mapping tool called LevelDB [9] was employed. In fact, LevelDB is used by the standard Bitcoin client Bitcoin-Qt to rapidly query massive quantities of Bitcoin blockchain data.

Figure 4 illustrates the intuition behind the clustering of address sets. Every new address is mapped to a new node. Consider the five transactions, Tx1, Tx2, Tx3, Tx4 and Tx5 (see Table 1). First, Tx1 is selected, which has only one transaction input address (Addr1). Mapping Tx1 and Tx2 is straightforward because Addr1 and Addr2 are from different transactions. However, Tx3 intersects with both Tx1 and Tx2, which implies that Addr1, Addr2 and Addr3 should be in the same set. Instead of modifying Addr1 and Addr2, it is only necessary to search for the node ID to which they map. Then, all the nodes that are found are mapped to the new node ID3 (Figure 4). Thus, all the sets are unioned without sequentially accessing the addresses in each set.

In the case of Tx4, there is no intersection with the other address sets, so it is mapped to a new node ID4. However, Tx5 intersects with several previous addresses, so the database is updated with the operations shown in Table 1. After iterating from Tx1 to Tx5, all five addresses are finally

Table 1. Operations on sample transaction inputs.

Transaction ID	Input Address List	Database Operation
Tx1	Addr1	(Addr1, ID1)
Tx2	Addr2	(Addr2, ID2)
Tx3	Addr1, Addr2, Addr3	(Addr3, ID3), (ID1, ID3), (ID2, ID3)
Tx4	Addr4	(Addr4, ID4)
Tx5	Addr1, Addr1, Addr5	(Addr5, ID5), (ID4, ID5), (ID3, ID5)

mapped to ID5. Using this approach, given any sets of addresses, it is possible to correctly union all the sets with intersections regardless of the order of the iteration (e.g., Tx5 to Tx1 instead of Tx1 to Tx5).

However, in the case of 36 million addresses, the intuitive algorithm is very inefficient because the directed path in the tree structure of Figure 4 tree can be long enough to require a single address search to traverse thousands of node IDs. The algorithm is improved by remapping a graph node to the root whenever a path is traversed.

Table 2. Improved operations on sample transaction inputs.

Transaction ID	Input Address List	Database Operation
Tx1	Addr1	(Addr1, ID1)
Tx2	Addr2	(Addr2, ID2)
Tx3	Addr1, Addr2, Addr3	(Addr3, ID3), (ID1, ID3), (ID2, ID3)
Tx4	Addr4	(Addr4, ID4)
Tx5	Addr1, Addr1, Addr5	(Addr5, ID5), (ID4, ID5), (Addr1, ID3), (ID3, ID5)

Table 2 shows the improvement in operations at Tx5. Because Tx5 has Addr1, a search is required from Addr1 to ID3 to obtain its root node ID. After it is found that Addr1 is actually mapped to ID3, the information is updated for future reference. Similar shortcuts can be applied to intermediate nodes such as ID2 if a traversal through ID2 is needed in the future.

The new algorithm is summarized as follows:

1. Select the next Bitcoin transaction and extract its input addresses (or outputs in case of a Coinbase transaction).

2. Examine the extracted address set. For addresses that are not yet in the database, map each address to a new ID. Ignore all the existing addresses, but traverse to the root node ID to which the

address points and save it in a set. During each traversal, optimize the path using a shortcut, if possible.

3. Examine the root node ID set in Step 2. If the set is empty, the algorithm is terminated. Otherwise, map the root node IDs in the set to the new node ID in Step 2; if no new addresses were found in Step 2, create a new node ID and map the root node IDs in the set to the new node ID.

4. Go to Step 1.

The four steps listed above map transaction input addresses and Coinbase output addresses. Note that one additional step is required to scan for one-time change addresses by applying the three rules listed above.

The application of the algorithm to the Bitcoin transaction dataset yielded good results. Of the 35,770,360 addresses that were processed in the dataset, 35,587,286 addresses are used as outputs. A total of 398,954 were identified as one-time change addresses, which validated the heuristic. In all, 13,062,822 distinct address sets were generated.

4.2 Address Graph Observations

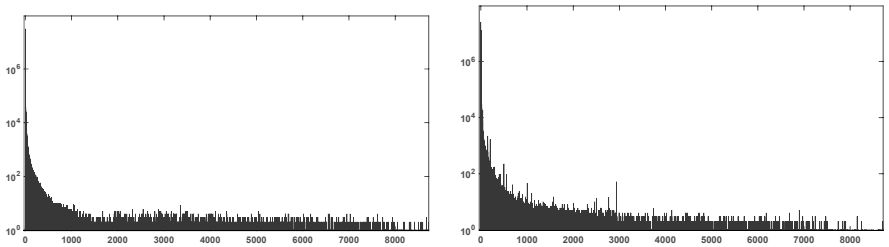
Having mapped Bitcoin addresses to groups, the next problem is to interpret raw Bitcoin transactions in a currency flow model between Bitcoin address sets. The objective is to create a directed graph that reflects monetary relationships between sets of addresses so that tracing incoming and outgoing payments to/from specific Bitcoin entities is possible.

A directed graph is defined as follows:

1. Each vertex in a directed graph represents a set of addresses, which corresponds to a Bitcoin entity that may possess multiple addresses.

2. Each edge in a directed graph reflects a specific Bitcoin payment. An outgoing edge from a vertex is a payment from the current address set to another vertex (address set). Similarly, an incoming edge is a payment received by the vertex from another vertex. Each edge in the directed graph is weighted according to the amount of the associated payment.

In order to study the transaction behavior at fixed time intervals, the Bitcoin transaction data from block #210,000 to block #314700 was divided into multiple samples, each consisting of 20 blocks, to yield a total of 5,236 graph data files. Each graph data file was converted into a directed graph as defined above.



(a) In-degree distribution. (b) Out-degree distribution.

Figure 5. Graph connection distributions.

Figure 5(a) shows the node counts for varying numbers of incoming edges computed over all 5,236 graphs. The majority of the nodes have less than 100 incoming edges as shown by the logarithmic scale on the y-axis. Figure 5(b) shows the node counts with varying numbers of outgoing edges, again computed over all 5,236 graphs. The graph connection distribution for outgoing edges is similar to that for incoming edges (Figure 5(a)).

The graphs have a total of 94,880,896 edges. An overwhelming majority of payments, 89,406,338 (94.2%) out of 94,880,896, were below 10 BTC.

4.3 Currency Flow Analysis

Currency flow analysis focuses on the graph edges to determine what happened to a specific sum of money. In theft and fraud cases, currency flow goes two ways. First, money is actually spent, which is equivalent to a payment being split and sent to multiple addresses. Second, the money can be sent to multiple addresses through different transactions before they are finally collected in a criminal's Bitcoin wallet. This process is, in some sense, similar to money laundering. In order to quantitatively capture the currency flow, a breadth-first search (BFS) of the directed graph was performed to determine the most probable directions of stolen Bitcoin flows.

The algorithm shown in Figure 6 starts with a set of nodes from the graph in fixed time periods. Using these nodes as the starting level, the graph is searched in a breadth-first manner and the node counts and net Bitcoin outputs at each level are saved until no more new nodes exist. The output of the algorithm is a list of node counts and a list of net Bitcoin outputs at each level. The net Bitcoin output is the difference between the total values of the output edges and incoming edges.

```
BFS_Graph(G):
Initialize curLevel={set of starting nodes}, nextLevel=new set()
Initialize countList=new list[], netOutputList=new list[]
  while curLevel is not empty:
    add node in curLevel into countList
    for each node in curLevel:
      add all unvisited neighbors of the node to nextLevel
      mark neighbors as visited
      netOutput=netOutput+sum(out edges)-sum(in edges)
      add netOutput into netOutputList
  curLevel=nextLevel
  nextLevel=new set()
```

Figure 6. Breadth-first search (BFS) algorithm for currency flows.

Bitcoin payments are forward tracked so that, given a set of starting nodes, a search is conducted for patterns of coin transfer to other entities in split sums or money laundering. Due to the potential incompleteness of the address classification database, money laundering can also show up as a large amount of Bitcoins ending up at nodes other than the start nodes. In this case, two graph patterns are expected:

- Starting nodes are involved in cycles consisting of large edges directed to them, with approximately the same total values as the total values of the outgoing edges from them.
- No valid graph cycles are found, but Bitcoins converge into a smaller number of large-valued edges that are gathered by a set of other nodes.

4.4 Mt. Gox Case Study

Having constructed a concrete data model that describes Bitcoin flow patterns, it is applied in a case study involving the Mt. Gox incident.

Mt. Gox was the largest Bitcoin exchange and web wallet service provider in the world. On February 7, 2014, Mt. Gox reported technical difficulties due to hacking attacks. On February 10, 2014, Mt. Gox issued a press release claiming that it had lost more than 850,000 BTC (worth half a billion U.S. dollars at the prevailing market rate). The incident, which resulted in Mt. Gox filing for bankruptcy, has been under investigation for more than a year, but no clear explanation has been provided as yet.

Decker and Wattenhofer [3] have conducted a study on Bitcoin transaction malleability, which was suspected to have led to the huge loss at Mt. Gox. According to their analysis, malleability attacks did exist

that falsely modified transactions to enable twice the amount of currency withdrawal from Mt. Gox. However, only a very small fraction of the alleged 850,000 BTC loss could be tracked in the malleability detection data. Indeed, it is still not clear how the rest of the Bitcoins were lost. According to a police investigation report [6], there were signs that Mt. Gox's offices had been physically infiltrated and at least one former employee may have pilfered electronic data.

Regardless of the reason, Mt. Gox sustained a massive loss of Bitcoins and this loss should be reflected in the blockchain data and in the directed graph data. The theft clearly involved large currency flows during a short period time; these patterns should be present in the directed graphs. As mentioned above, 94.2% of the edges in the overall graph involve transactions less that 10 BTC. To reduce the size of the dataset, the graphs were pruned to retain only edges of 10 BTC or higher. This simplified the graph sets and renders it more efficient to discover large currency flows.

All the graph nodes and edges in the time period of interest were collected and integrated in a new graph that described global transaction patterns during the incident. According to the Mt. Gox incident timeline, technical difficulties were experienced and withdrawals were suspended on February 7, 2014. Because theft could not have occurred after the suspension of withdrawals, the time window could be restricted to a few days before February 7, 2014. Thus, the analysis started with graph edges with the timestamp `19:49:42 2014-02-03` and ended at the timestamp `08:36:47 2014-02-07`. Figure 7 presents a visualization of the directed graph used in the analysis.

Figure 8 displays the sum of values of all edges, equivalent to the total transaction amount at each timestamp. The sum of edge values provide information about the groups of edges that have a burst of Bitcoin flow and the time when this occurs. A burst of Bitcoin flow indicates a large-scale currency transfer. When computing edge sums, all the edges that started and ended at the same node were removed to focus on currency transfers between different entities.

Figure 8 shows that the maximum peak burst is at the x-axis values 359 to 363, corresponding to the timestamps `23:57:01 2014-02-05` to `00:28:17 2014-02-06`. However, the total output values of these edges only account for about 100,000 BTC, significantly less than the 850,000 BTC allegedly stolen from Mt. Gox. Although, the large loss incurred by Mt. Gox cannot be explained, the maximum peak burst is related to the theft with high probability. Indicators of other theft activities must exist elsewhere, but the maximum peak burst in the graph is certainly the latest possible peak that can be identified. In fact, it is very reasonable

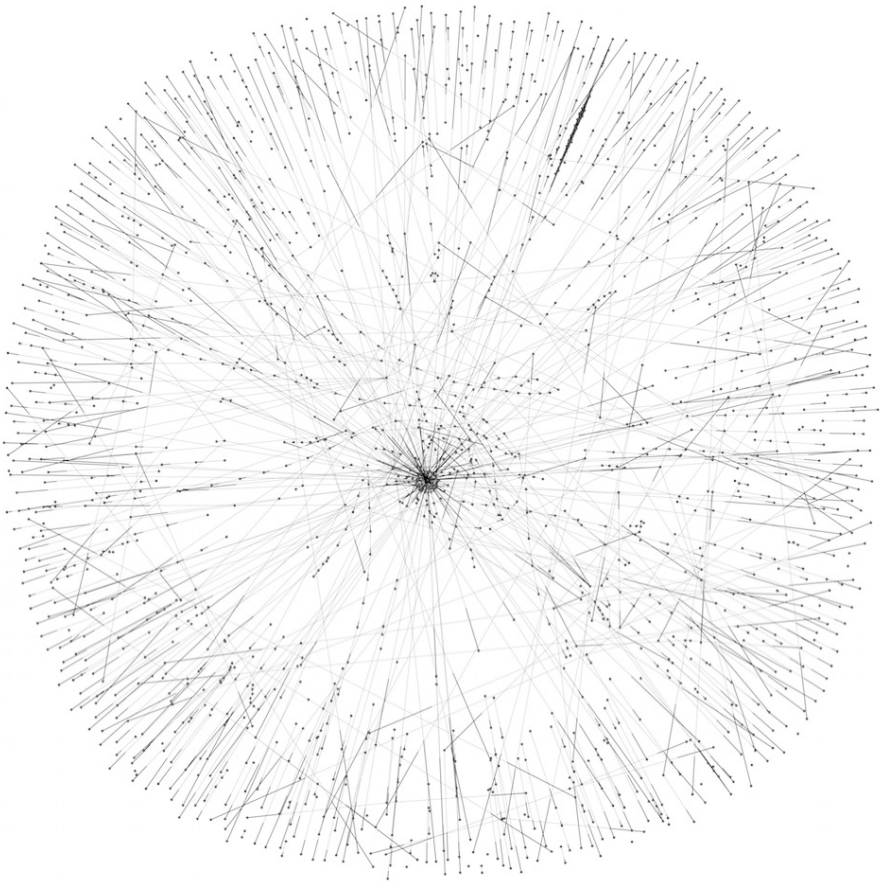


Figure 7. Directed graph used in the Mt. Gox case study.

that the theft would have occurred about a day or two before Mt. Gox suspended withdrawals and made its formal announcement.

Figure 9 shows the suspicious nodes with the largest edge values. The graph was obtained by collecting all the nodes connected to edges that contributed to the maximum peak burst peak (153 distinct nodes) and ordering the nodes based on increasing edge values (the y-axis value is the sum of values of all the edges at that time). Of particular interest are the last fourteen nodes that received the highest payments (higher than 4,000 BTC).

The breadth-first search algorithm was employed with its traversal list initialized to contain the fourteen nodes. This algorithm terminated after 1,461 levels. Figure 10(a) displays the node counts during the execution of the algorithm. The node count declines drastically after

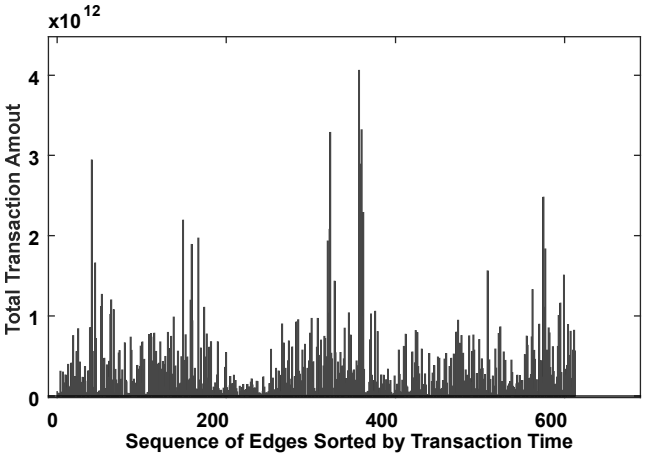


Figure 8. Sum of transaction amounts by time.

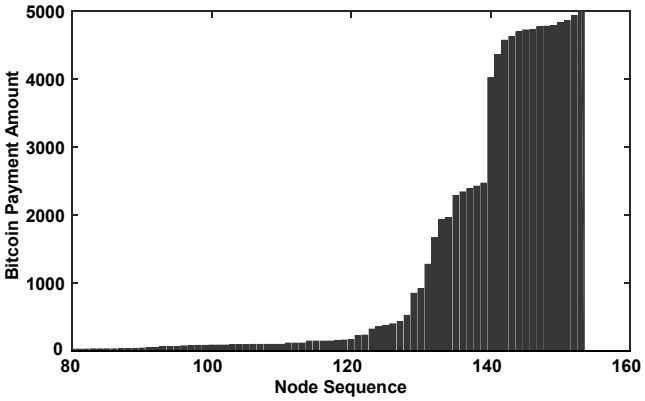
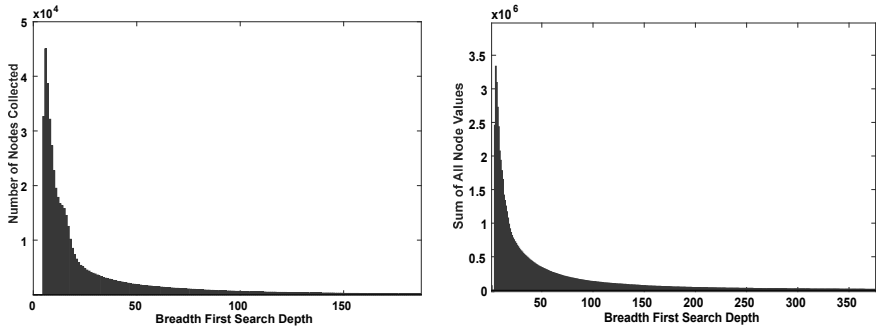


Figure 9. Suspicious nodes with the largest edge values.

the first 20 levels and finally converges to a single node. However, this graph does not provide enough information about the destination of the transaction burst from `23:57:01 2014-02-05` to `00:28:17 2014-02-06` because the Bitcoin outputs also have to be considered.

Figure 10(b) plots the net output of nodes at each level. The initial fourteen nodes transfer 70,510 BTC to other entities. Next, it is necessary to search for a level such that nodes spend all 70,510 BTC. The negative outputs in Table 3 indicate nodes at the current level that receive more income from other sources. When the algorithm reaches the level (row 4) where twelve nodes have a much larger output than all previous income, it appears that the initial 70,510 BTC being tracked is split to the following 32,621 nodes. The decreasing output pattern in



(a) BFS node counts.

(b) BFS depth outputs.

Figure 10. Node count at each breadth-first search level.

Table 3. Initial breadth-first search stages.

Number of Nodes	Network Output
12	70,510
14	-31,205.998
17	-49,545
12	2,458,376
32,621	3,335,101

Figure 10(b) does not provide adequate evidence of money laundering – such evidence would correspond to a large amount of Bitcoins converging to a small set of nodes or flowing back to the starting nodes in a reasonably short period of time.

5. Discussion

Certain observations can be made based on the experiments conducted with the Bitcoin dataset. First, with regard to the large Bitcoin transfer, there is no significant evidence of money laundering activities for which edge-convergent patterns or graph cycles consisting of edges of equivalent values would be seen. Second, the experimental results show with high probability that a large portion of the suspected stolen Bitcoins was spent and split using many payments to different entities instead of being split among the conspirators.

However, there are two major restrictions. First, address classification is transitive so that future transactions potentially affect the group database by having multiple intersections with existing address sets. The latest block transaction data available in the dataset is 347,000, but it is expected that future transactions would provide stronger hints

for existing address sets to be unioned. Moreover, the application of anti-anonymity approaches is recommended before address classification because it can significantly reduce the number of sets (13,062,822 distinct address sets were generated in this work).

The second major restriction relates to the address graph experiment. Because the dataset did not contain the complete blockchain data, it was unable to determine the exact Bitcoin value that a graph vertex holds at each timestamp. By tracing the relative in-flow and out-flow of currency, the breadth-first search results only reveal the flow of suspected currency without providing an accurate list of addresses of potential conspirators.

6. Conclusions

The graph-based method for analyzing the identity clustering and currency flow properties of Bitcoin transactions can expedite forensic investigations of criminal cases such as Mt. Gox that involve double-spending, theft, money laundering or fraudulent transactions. Experiments performed using two years of Bitcoin transaction data reveal that different Bitcoin addresses belonging to the same entity can be effectively clustered to reduce the forensic effort required to track Bitcoin users involved in criminal activities. Furthermore, the graph-based analysis of Bitcoin transactions can help discern suspicious transaction behaviors and currency flows involving Bitcoin users.

Future work will focus on enhancing the address group database to accommodate new transaction streams. Additionally, efforts will be made to augment the Bitcoin address library with real-world Bitcoin entities to help reduce the size of address sets during forensic investigations.

References

[1] J. Bergstra and P. Weijland, Bitcoin: A Money-Like Informational Commodity (`arxiv.org/pdf/1402.4778v1.pdf`), 2014.

[2] Bitcoin Community, Technical background of Bitcoin addresses, *Bitcoin Wiki* (`en.bitcoin.it/wiki/Technical_background_of_Bitcoin_addresses`), 2015.

[3] C. Decker and R. Wattenhofer, Bitcoin transaction malleability and Mt. Gox, *Proceedings of the Nineteenth European Symposium on Research in Computer Security*, pp. 313–326, 2014.

[4] GitHub, Bitcoin core integration/staging tree (`github.com/bitcoin/bitcoin`).

[5] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. Voelker and S. Savage, A fistful of Bitcoins: Characterizing payments among men with no names, *Proceedings of the Conference on Internet Measurement*, pp. 127–140, 2013.

[6] T. Mochizuki and E. Warnock, Mt. Gox head believes no more Bitcoins will be found, *The Wall Street Journal*, June 29, 2014.

[7] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System (`bitcoin.org/bitcoin.pdf`), 2009.

[8] National Institute of Standards and Technology, Digital Signature Standard (DSS), FIPS PUB 186-4, Gaithersburg, Maryland, 2013.

[9] py-leveldb Project, Thread-safe Python bindings for LevelDB (`co.de.google.com/p/py-leveldb`).

[10] D. Ron and A. Shamir, Quantitative analysis of the full Bitcoin transaction graph, in *Financial Cryptography and Data Security*, A. Sadeghi (Ed.), Springer-Verlag, Berlin Heidelberg, pp. 6–24, 2013.