Bitcoin and U.S. Capitol Riot

0. Introduction

On January 6, 2021, thousands of former President Donald Trump's supporters gathered at the U.S. Capitol and attacked the United States Congress in an attempt to overturn his defeat in the 2020 United States presidential election. Reportedly, on Dec. 08. 2020, one month before this capitol riot, there was a large bitcoin transaction. The doner was a programmer based in France, and the receivers involved several far-right activists. Motivated by this incident, we were curious about the following questions:

(1) identifying the receiver addresses, and determine if all addresses belong to different wallets;

(2) investigating who these riot wallets often sent money to and received money from.

It was found that the common union-find algorithm for aggregating wallets may be more accurate in identifying wallets that belong to large trading websites and less accurate in distinguishing those that belong to individuals. In terms of behaviors, right-wing activists tend to transact with each other and gambling websites.

1. Background

1.1. Bitcoin transaction graph

Bitcoin transactions are recorded in the ledger. For each transaction, there are multiple input addresses and output addresses. As shown in Fig. 1, tx1 and tx2 each represent one transaction, and tx1 has a1, a2 as its input addresses, and a2, a3 as output addresses; tx2 has a2, a5 as input addresses, and a3, a4 as output addresses.

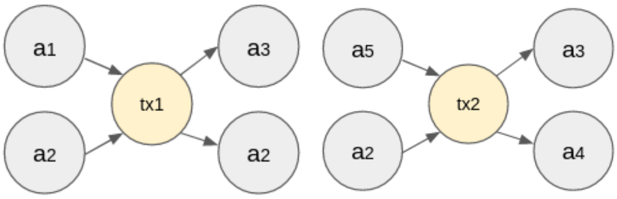


Figure. 1 Bitcoin transaction graph

1.2. Wallet aggregation heuristics

The Bitcoin transaction network allows users to have multiple addresses in his/her wallet, and each user can leverage the money from different addresses as input in one transaction. Due to this mechanism, to study user behavior, it is reasonable to aggregate addresses into wallets, and use one wallet as a unit of analysis.

The method that regarded all addresses that can be spent together as one wallet is called **transitive closure**, which can be implemented with **Weighted Quick-Union with Path Compression (WQUPC)** algorithm. For example, after aggregation, the addresses in Figure. 1 can be illustrated in Figure. 2.

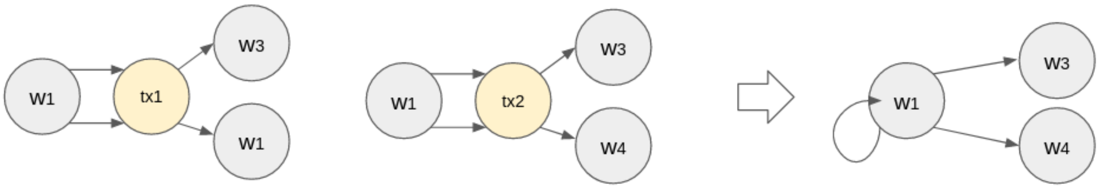


Figure. 2 Address-aggregated bitcoin transaction graph

Table 1. WQUPC Algorithm

Requirements:
(i) An edge list that records addresses that were ever spent together such that we can read in the edge list do the Union operation (e.g. if address 1 and address 2 were spent together, we will do Union(1,2))
(ii) Records of **the parent of each node** and **the rank of each node**
*p in the following procedure means parent

```
Make-Set(x)
  x.p=x
  x.rank=0

Union(x,y)
  Link(Find-Set(x),Find-Set(y))

Link(x,y)
  if x.rank > y.rank
    y.p=x
  else x.p=y
    if x.rank == y.rank
      y.rank = y.rank+1

Find-Set(x)
  if x != x.p
    x.p = Find-Set(x.p)
  return x.p
```

Table 2. Steps of assigning wallet IDs to addresses

Requirements:
(i) An edge list that records addresses that were ever spent together
(ii) The parent record resulted from the WQUPC Algorithm

Step 1: read in edge list and run WQUPC algorithm on all addresses

Step 2: Use the parent record to find the root for each address

Step 3: Export the root as wallet ID for each address

1.3. Bitcoin & far-right party

DuPont (2017) has summarized that the politics of Bitcoin is right-wing. Moreover, according to multiple online resources, bitcoin has become a popular transaction tool for right-wing activists and hate groups at the end of 2017. The skyrocketed trend can be largely attributed to Richard Spencer, the white nationalist's declaration on Twitter: "Bitcoin is the currency of the alt-right."

1.4. Identify important wallets in the network

To investigate some of the critical wallets for those riot wallets in the transaction network is equivalent to finding vertices that are most relevant to a specific node in a graph. This kind of problem setup is one of the applications for Personalized PageRank (PPR). As the bitcoin transaction network is large, and running PPR on the whole graph would be inefficient, we adopted a faster modification of PPR, the approximate PPR algorithm, to solve the problem.

Table 3. Approximate Personalized PageRank Algorithm

```
Requirement: undirected graph G, PPR vector p$\in[0,1]^N$ , preference vector $\pi$, teleportation constant $\alpha$, and tolerance $\varepsilon$
  Initialize p$\leftarrow$0, r$\leftarrow\pi$, $\alpha'\leftarrow\alpha/(2-\alpha)$
  while $\exists$ u$\in$V such that $r_u \geq \varepsilon d_u$ do
    uniformly sample a vertex u satisfying $r_u \geq \varepsilon d_u$
      $p_u \leftarrow p_u + \alpha' r_u$
    for v:(u,v)$\in$E do
      $r_u \leftarrow r_v + (1-\alpha') r_u/(2d_u)$
    end for
      $r_u \leftarrow (1-\alpha') r_u/2$
  end while
  return $\varepsilon$-approximate PPR vector p
```

2. Method & Data

2.1. Data preprocessing and filtering

The blockchain is split and stored in multiple blk*.dat files which starts with blk00000.dat, blk00001.dat, …, up until now there are more than 2000 blk*.dat files. The maximum file size for each .dat is 128 MB, which consists of several blocks of data, and around 2000 transactions on average.

2.1.1. Convert binary data to csv files

The blockchain .dat files are recorded in binary form. Thus, we wrote several scripts to convert the binary data into JSON [ledger-to-json-script], and further into CSV form [json-to-csv-script] in several batches. In the end, there are multiple CSV files being created:

(1) addrxxxxx.csv files - store the addresses that occur in each blockxxxxx;
(2) blockxxxxx.csv files - record the transaction data in each blockxxxxx, containing block height, transaction index, the transaction type, and transaction value;
(3) block_info.csv files - each record information about the batch of blocks that were being processed, e.g. the timestamp, number of transactions, number of BTCs, miner address, difficulty, and number of new addresses;
(4) multi-sig.csv files - each record information about multi-sig addresses, e.g., the block height, transaction index(i), the i-th transaction type(input or output), number of addresses involved in this multi-sig case.

2.1.2. Insert CSV filse into SQLite database

For each addrxxxxx.csv file, we dropped duplicates and further insert all addresses into a SQLite database with a unique constrain, such that we only store each address once. The table was called the **address** table. Other CSV files were inserted into the SQLite database without further processing, respectively called the **blockTransaction** table, the **blockInfo** table, and the **multiSig** table.

After the tables were created, we leveraged the SQLite query to map row ID as address ID from the address table to the address in rest of the tables, i.e. each **blockTransaction**, **blockInfo**, **multiSig** table now has an extra column called addressID. Then, we exported the mapped **blockTransaction** tables as csv to create input-spent-together edge lists.

2.1.3. Create input-spent-together edge lists [scripts]

This step was done in batches. Each **blockTransaction** table contains a range of block heights, where each block height consists of thousands of transactions on average. Each transaction involves several input addresses, with Numpy, we extracted all possible combinations of the input addresses as edge lists.

2.1.4. Aggregate addresses into wallets [scripts]

We leveraged the WQUPC algorithm described in 1.2 to aggregate addresses into wallets. The edge list from 2.1.3 will stream through, and each row of the list consists of the address ID that needs to be unioned together. In the end, the root of each cluster will be the wallet ID for all addresses within that cluster. Finally, we exported the address-wallet ID table, and inserted into SQLite database to mutate a wallet ID column for each **blockTransaction** table. Additionally, we left joined blockInfo with **blockTransaction** table by timestamp, such that each row of the **blockTransaction** table includes wallet ID, timestamp, block height, transaction index, transaction type, and transaction value.

2.1.5. Graph database

For better computational efficiency, a graph database, Neo4j, was introduced. The design of the database is as follows:
(1) Node
- Wallet: wallet ID
- Transaction: transaction ID (concatenated with block height and transaction index), timestamp (inherit from block height's timestamp)

(2) Edge
- Input: value
- Output: value

After the database was constructed, we filtered to keep transaction data after January 2017 (until June 2020), due to the fact that it was believed that bitcoin started to become popular among the right-wing activists after 2017.

Moreover, we further simplified the wallet-transaction-wallet graph into a **wallet-wallet graph**. If there exists a path between two wallets, merge the path and name the edge **SENT_TO**. For PPR computation efficiency (faster on a more connected graph), we removed self-loops and nodes whose total degree ≤ 1. Then, we exported the wallet-wallet graph to a csv file, and imported it into another graph database. Eventually, the wallet-wallet graph contains 206,637,177 nodes and 503,589,462 edges.

2.2. Down-stream and up-stream Personalized PageRank

To investigate who the riot wallets often sent money to and receive money from, we leveraged the Personalized PageRank algorithm as described in 1.4, respectively called down-stream PPR and up-stream PPR. Specifically, the built-in PPR function from Neo4j was implemented.
Down-stream PPR and up-stream PPR are similar, here we illustrate the parameter setting for the down-stream PPR.
(1) create an in-memory graph that encodes **SENT_TO** as **SENT** (for up-stream, encodes **SENT_TO** as **RECEIVE**)
(2) for each riot wallet, treat it as the seed node, and run PPR with α=0.15, and iteration=20
(3) return wallet IDs with PPR score ranked top 10

2.3. External information

We collected some external information from the Internet to help interpret the owners of the addresses.
(1) Alt-right activists' bitcoin addresses that are public online
(2) Top rich addresses scraped from well-known cryptocurrency transaction websites
(3) By observation
(4) Twitter and Medium posts' discussions

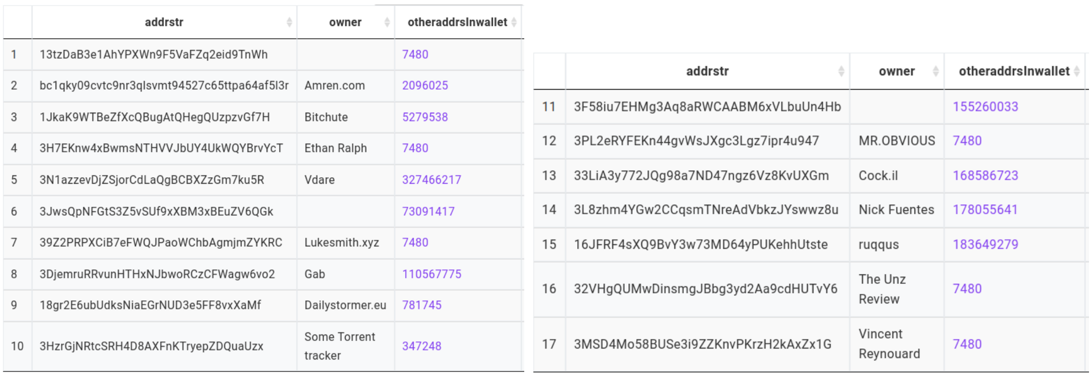
3. Results

A simple visualization website was created for this project, and the analysis is discussed below.

3.1. Address/Wallet owners

Among all 23 addresses that occurred in the original transaction, 17 of them are in our database. Through wallet aggregation, it was found that the addresses belong to 12 different wallets (Figure 3). Indeed, most of them belong to right-wing activists. The reason that we could verify the ownership was through the website BitcoinWhosWho, and some of them publicly broadcasted their addresses for people to donate.

However, as shown in Figure 3, the transitive closure wallet aggregation heuristic is more capable of identifying different transaction websites than individuals, that is, different individuals may be aggregated into the same wallet, while different transaction websites are correctly inferred as disparate wallets.



	addrstr	owner	otheraddrsInwallet
1	13tzDaB3e1AhYPXWn9F5VaFZq2eid9TnWh		7480
2	bc1qky09cvtc9nr3qlsvmt94527c65ttpa64af5l3r	Amren.com	2096025
3	1JkaK9WTBeZfXcQBugAtQHegQUzpzvGf7H	Bitchute	5279538
4	3H7EKnw4xBwmsNTHVVJbUY4UkWQYBrvYcT	Ethan Ralph	7480
5	3N1azzevDjZSjorCdLaQgBCBXZzGm7ku5R	Vdare	327466217
6	3JwsQpNFGtS3Z5vSUf9xXBM3xBEuZV6QGk		73091417
7	39Z2PRPXCiB7eFWQJPaoWChbAgmjmZYKRC	Lukesmith.xyz	7480
8	3DjemruRRvunHTHxNJbwoRCzCFWagw6vo2	Gab	110567775
9	18gr2E6ubUdksNiaEGrNUD3e5FF8vxXaMf	Dailystormer.eu	781745
10	3HzrGjNRtcSRH4D8AXFnKTryepZDQuaUzx	Some Torrent tracker	347248
11	3F58iu7EHMg3Aq8aRWCAABM6xVLbuUn4Hb		155260033
12	3PL2eRYFEKn44gvWsJXgc3Lgz7ipr4u947	MR.OBVIOUS	7480
13	33LiA3y772JQg98a7ND47ngz6Vz8KvUXGm	Cock.il	168586723
14	3L8zhm4YGw2CCqsmTNreAdVbkzJYswwz8u	Nick Fuentes	178055641
15	16JFRF4sXQ9BvY3w73MD64yPUKehhUtste	ruqqus	183649279
16	32VHgQUMwDinsmgJBbg3yd2Aa9cdHUTvY6	The Unz Review	7480
17	3MSD4Mo58BUSe3i9ZZKnvPKrzH2kAxZx1G	Vincent Reynouard	7480

Figure 3. Address/wallet ownership

3.2. Down-stream receivers/Up-stream senders of the riot wallets

Overall, many of the wallets shown in 3.1 are only receiving bitcoins, and have never sent bitcoins to others in our selected timestamp. Thus, overall, most wallets' in-degree > out-degree. The observation might be attributed to the factor that many alt-right activists are receiving donations through cryptocurrency because of the pseudo-anonymous property that cryptocurrency possesses.

Figure 4 displays the top 10 downstream and upstream wallets for each riot wallet. Right-wing figures were encoded in red; wallets that contain addresses that were discussed in Twitter were encoded yellow; wallets that possess addresses that were discussed in both Twitter and Medium were encoded green; wallets that contain addresses that belong to well-known transaction or gambling webs were encoded pink; wallets that contain addresses belong to a mixture of well-known webs are encoded purple; the wallet that has property summarized by our observation was encoded gray; the rest unknown wallets were encoded blue. Overall, the riot wallets are not only transacting bitcoins with each other but transacting with famous gambling websites.

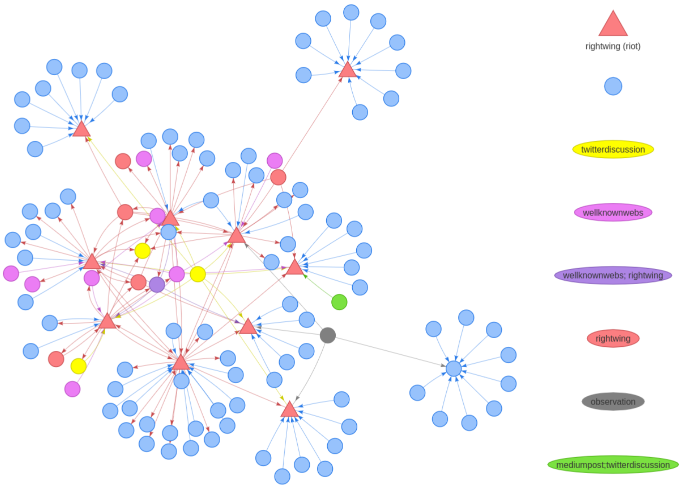


Figure 4. Top 10 Down-stream/Up-stream PPR network of the riot wallets