Bitcoin De-Anonymization

ChengMing 517030910287

June 21, 2020

1 Introduction

Bitcoin uses a pseudonym system. Although all transactions of bitcoin are public, what is publicly displayed is only an account code of the transaction user, that is, "address". This address can be generated at will, so it can't be connected with real life identity. It objectively provides anonymity, but it is also used by lawbreakers. However, bitcoin can not guarantee complete anonymity. When bitcoin transactions have contact with the real world, it may disclose real information (such as using bitcoin for shopping, or using bitcoin to exchange for legal currency). Because all transactions on the block are public, this provides a breakthrough for the anonymization of bitcoin.

There are two ways to de-anonymize bitcoin network. One is de-anonymize on the network layer. Using the broadcast of bitcoin transaction in P2P network, we can determine the broadcast source by manipulating multiple nodes and other means, so as to associate the user address with the IP address and realize de-anonymity. The second is the application layer de anonymization, which mainly relies on the transaction map analysis, builds the transaction network in the blockchain, and matches the real transaction information obtained in various ways, so as to realize the de anonymization. In this project, I did some research on the application layer de anonymization

2 Basic idea

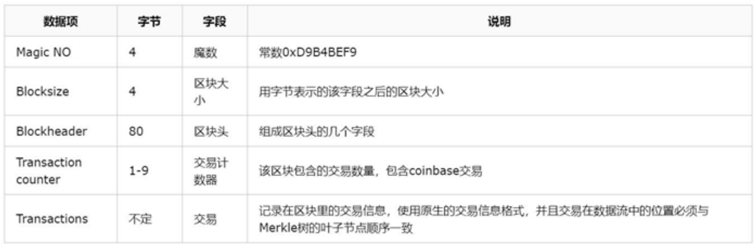
In the specific implementation process, we first need to obtain the transaction information on the bitcoin blockchain, then extract useful node and edge information from the transaction information, so as to build the bitcoin transaction network. Secondly, we need to obtain the information about the connection between the bitcoin transaction and the real world. I selected the record of transactions leaked on bitcoin transaction platform Mt.Gox in 2014. The account in this transaction record is registered with real world identity. If this part of transaction record can be matched with the transaction record of bitcoin blockchain, then our de anonymization work is successful

3 Data Preparation

3.1 bitcoin blockchain

There are two way to get the full data of bitcoin blockchain:

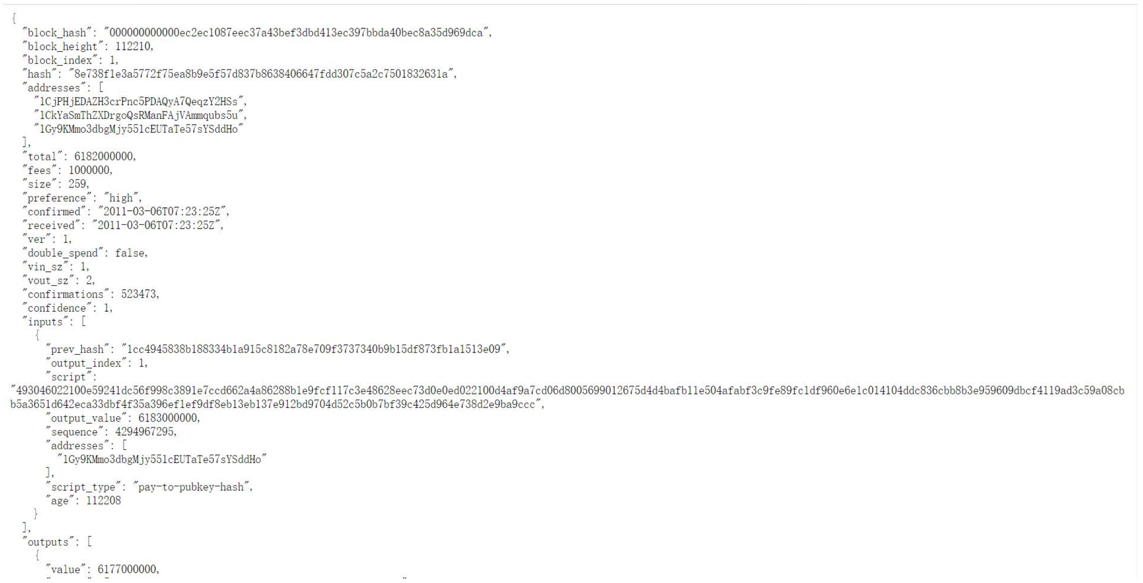
1. download bitcoin wallet, then it will download all blockchain automatically. The data we download is in the .dat file. The file has a very standard format. The specific format can be seen in the official document. There are multiple blocks in a .dat file. Each part of the block occupies the corresponding number of bytes, which can be divided easily and read in order.



数据项	字节	字段	说明
Magic NO	4	魔数	常数0xD9B4BEF9
Blocksize	4	区块大小	用字节表示的该字段之后的区块大小
Blockheader	80	区块头	组成区块头的几个字段
Transaction counter	1-9	交易计数器	该区块包含的交易数量，包含coinbase交易
Transactions	不定	交易	记录在区块里的交易信息，使用原生的交易信息格式，并且交易在数据流中的位置必须与Merkle树的叶子节点顺序一致

Figure 1: .dat file format

2. get block data or transaction data by requesting API, like https://api.blockcypher.com/v1/btc/main/txs/txhash. The data get from API is in json format. Here is an example:

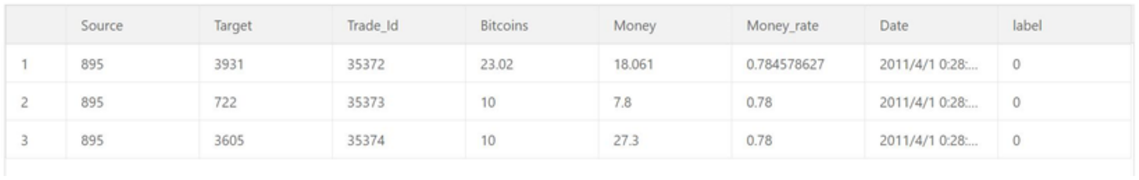


```
{
  "block_hash": "000000000000ec2ec1087eec37a43bef3dbd413ec397bbda40bec8a35d969dca",
  "block_height": 112210,
  "block_index": 1,
  "hash": "8e738f1e3a5772f75ea8b9e5f57d837b8638406647fdd307c5a2c7501832631a",
  "addresses": [
    "1CjPHjEDAZH3crPnc5PDAQyA7QeqzY2HSs",
    "1CkYaSmThZXDrgoQsRManFAjVAmmqubs5u",
    "1Gy9KMmo3dbgMjy551cEUTaTe57sYSddHo"
  ],
  "total": 6182000000,
  "fees": 1000000,
  "size": 259,
  "preference": "high",
  "confirmed": "2011-03-06T07:23:25Z",
  "received": "2011-03-06T07:23:25Z",
  "ver": 1,
  "double_spend": false,
  "vin_sz": 1,
  "vout_sz": 2,
  "confirmations": 523473,
  "confidence": 1,
  "inputs": [
    {
      "prev_hash": "1cc4945838b188334b1a915c8182a78e709f3737340b9b15df873fb1a1513e09",
      "output_index": 1,
      "script":
"493046022100e59241dc56f998c3891e7ccd662a4a86288b1e9fcf117c3e48628eec73d0e0ed022100d4af9a7cd06d8005699012675d4d4bafb11e504afabf3c9fe89fc1df960e6e1c014104ddc836cbb8b3e959609dbcf4119ad3c59a08cb
b5a3651d642eca33dbf4f35a396ef1ef9df8eb13eb137e912bd9704d52c5b0b7bf39c425d964e738d2e9ba9ccc",
      "output_value": 6183000000,
      "sequence": 4294967295,
      "addresses": [
        "1Gy9KMmo3dbgMjy551cEUTaTe57sYSddHo"
      ],
      "script_type": "pay-to-pubkey-hash",
      "age": 112208
    }
  ],
  "outputs": [
    {
      "value": 6177000000,
```

Figure 2: API json format

3.2 Mt.Gox leaked DataSet

Mt.Gox leaked DataSet can be download in many dataset website, such as kaggle and github. here is one source: https://www.kaggle.com/xblock/mtgox-leaked-transaction
The data I need in this project is in complet_edge_v2.csv, the format of this file is shown below.



	Source	Target	Trade_Id	Bitcoins	Money	Money_rate	Date	label
1	895	3931	35372	23.02	18.061	0.784578627	2011/4/1 0:28:...	0
2	895	722	35373	10	7.8	0.78	2011/4/1 0:28:...	0
3	895	3605	35374	10	27.3	0.78	2011/4/1 0:28:...	0

Figure 3: API json format

- **Source:** bitcoin seller
- **Target:** bitcoin buyer/receiver
- **Trade_Id:** Id of this transaction
- **Bitcoins:** the amount of bitcoins in this transction
- **Money:** the amount of dollars for buy bitcoin
- **Money_rate:** Unit bitcoin price
- **Date:** Time of transaction
- **label:** the type of user

4 Process data

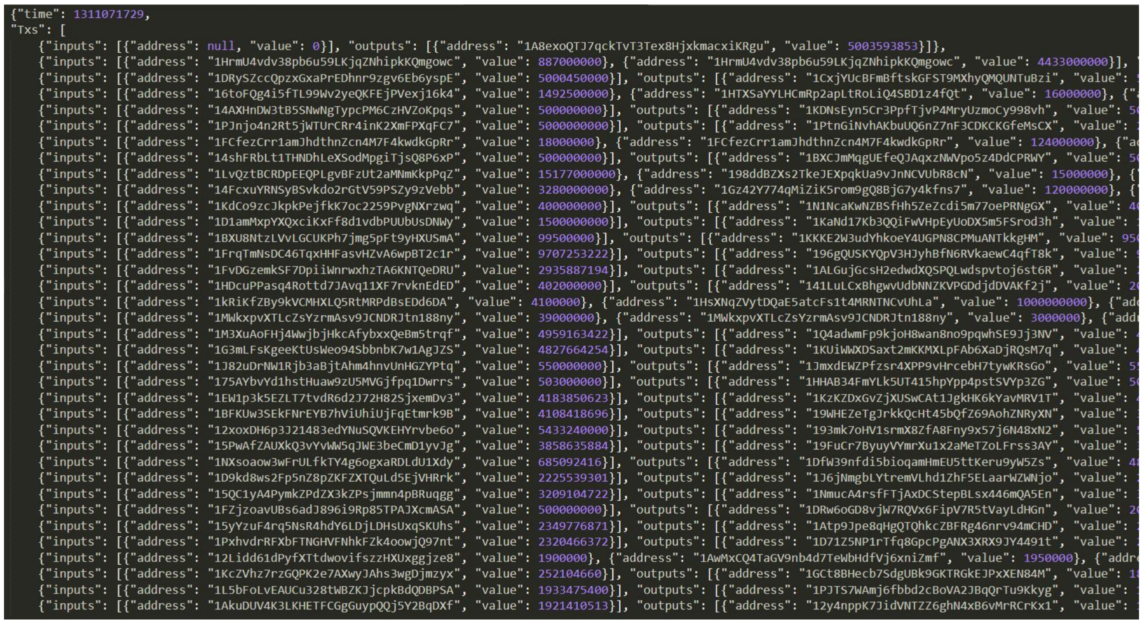
In order to get useful information, we need to process the data.

4.1 bitcoin blockchain

our goal is to obtain particular information of every transction in blockchain, include input addresses, input value, output addresses and output value. Also, we need to obtain the time of each block.

The time of each block is in the blockheader, so can obtain it easily. The output value and public key is in output transaction part. we can obtain them too. and the address can be generate by the public key. But the address and public key of input is not in corresponding transction. we can only get the previous hash, which point to the source of bitcoin. We should search transaction whoes hash is same as the previous hash, and get the address and public key of its output.

After getting all these information, we should save them in json form. Some more details such as how to generate address by public key and how to read .dat file in certain order will not be mentioned here. You can see the detail of this part is in DataProcess.py



```
{"time": 1311071729,
"Txs": [
    {"inputs": [{"address": null, "value": 0}], "outputs": [{"address": "1A8exoQTJ7qckTvT3Tex8HjxkmacxiKRgu", "value": 5003593853}]},
    {"inputs": [{"address": "1HrmU4vdv38pb6u59LKjqZNhipkKQmgowc", "value": 887000000}, {"address": "1HrmU4vdv38pb6u59LKjqZNhipkKQmgowc", "value": 4433000000}],
    {"inputs": [{"address": "1DRySZccQpzxGxaPrEDhnr9zgv6Eb6yspE", "value": 5000450000}], "outputs": [{"address": "1CxjYUcBFmBftskGFST9MXhyQMQUNTuBzi", "value":
    {"inputs": [{"address": "16toFQg4i5fTL99Wv2yeQKFEjPVexj16k4", "value": 1492500000}, {"address": "1HTXSaYYLHCmRp2apLtRoLiQ4SBD1z4fQt", "value": 16000000}, {"
    {"inputs": [{"address": "14AXHnDW3tB5SNwNgTypcPM6CzHVZoKpqs", "value": 500000000}], "outputs": [{"address": "1KDNsEyn5Cr3PpfTjvP4MryUzmoCy998vh", "value": 5
    {"inputs": [{"address": "1PJnjo4n2Rt5jWTUrCRr4inK2XmFPXqFC7", "value": 5000000000}], "outputs": [{"address": "1PtnGiNvhAKbuUQ6nZ7nF3CDKCKGfeMsCX", "value":
    {"inputs": [{"address": "1FCfezCrr1amJhdthnZcn4M7F4kwdkGpRr", "value": 18000000}, {"address": "1FCfezCrr1amJhdthnZcn4M7F4kwdkGpRr", "value": 124000000}, {"a
    {"inputs": [{"address": "14shFRbLt1THNDhLeXSodMpgiTjsQ8P6xP", "value": 500000000}], "outputs": [{"address": "1BXCJmMqgUEfeQJAqxzNWVpo5z4DdCPRWY", "value": 5
    {"inputs": [{"address": "1LvQztBCRDpEEQPLgvBFzUt2aMNmKkpPqZ", "value": 15177000000}, {"address": "198ddBZXs2TkeJEXpqkUa9vJnNCVUbR8cN", "value": 15000000}, {
    {"inputs": [{"address": "14FcxuYRNSyBSvkdo2rGtV59PSZy9zVebb", "value": 3280000000}, {"address": "1Gz42Y774qMiZiK5rom9gQ8BjG7y4kfns7", "value": 120000000}, {
    {"inputs": [{"address": "1KdCo9zcJkpkPejfkK7oc2259PvgNXrzwq", "value": 400000000}], "outputs": [{"address": "1N1NcaKwNZBSfHh5ZeZcdi5m77oePRNgGX", "value": 4
    {"inputs": [{"address": "1D1amMxpYXQxciKxFf8d1vdbPUUbUsDNWy", "value": 1500000000}], "outputs": [{"address": "1KaNd17Kb3QQiFwVHpEyUoDX5m5FSrod3h", "value":
    {"inputs": [{"address": "1BXU8NtzLVvLGCUKPh7jmg5pFt9yHXUSmA", "value": 99500000}], "outputs": [{"address": "1KKKE2W3udYhkoeY4UGPN8CPMuANTkkgHM", "value": 95
    {"inputs": [{"address": "1FrqTmNsDC46TqxHHFasvHZvA6wpBT2c1r", "value": 9707253222}], "outputs": [{"address": "196gQUSKYQpV3HJyhBfN6RVkaewC4qfT8k", "value":
    {"inputs": [{"address": "1FvDGzemkSF7DpiiWnrwxhzTA6KNTQeDRU", "value": 2935887194}], "outputs": [{"address": "1ALGujGcsH2edwdXQSPQLwdspvtoj6st6R", "value":
    {"inputs": [{"address": "1HDcuPPasq4Rottd7JAvq11XF7rvknEdED", "value": 402000000}], "outputs": [{"address": "141LuLCxBhgwvUdbNNZKVPGDdjdDVAKf2j", "value": 2
    {"inputs": [{"address": "1kRiKfZBy9kVCMHXLQ5RtMRPdBsEDd6DA", "value": 4100000}, {"address": "1HsXNqZVytDQaE5atcFs1t4MRNTNCvUhLa", "value": 1000000000}, {"ad
    {"inputs": [{"address": "1MWkxpvXTLcZsYzrmAsv9JCNDRJtn188ny", "value": 39000000}, {"address": "1MWkxpvXTLcZsYzrmAsv9JCNDRJtn188ny", "value": 3000000}, {"addr
    {"inputs": [{"address": "1M3XuAoFHj4WwjbjHkcAfybxxQeBm5trqf", "value": 4959163422}], "outputs": [{"address": "1Q4adwmFp9kjoH8wan8no9pqwhSE9Jj3NV", "value":
    {"inputs": [{"address": "1G3mLFsKgeeKtUsWeo94SbbnbK7w1AgJZS", "value": 4827664254}], "outputs": [{"address": "1KUiWWXDSaxt2mKKMXLpFAb6XaDjRQsM7q", "value":
    {"inputs": [{"address": "1J82uDrNW1Rjb3aBjtAhm4hnvUnHGZYPtq", "value": 550000000}], "outputs": [{"address": "1JmxdEWZPfzsr4XPP9vHrcebH7tywKRsGo", "value": 5
    {"inputs": [{"address": "175AYbvYd1hstHuaw9zU5MVGjfpq1Dwrrs", "value": 503000000}], "outputs": [{"address": "1HHAB34FmYLk5UT415hpYpp4pstSVYp3ZG", "value": 5
    {"inputs": [{"address": "1EW1p3k5EZLT7tvdR6d2J72H82SjxemDv3", "value": 4183850623}], "outputs": [{"address": "1KzKZDxGvZjXUSwCAt1JgkHK6kYavMRV1T", "value":
    {"inputs": [{"address": "1BFKUw3SEkFNrEYB7hViUhiUjFqEtmrk9B", "value": 4108418696}], "outputs": [{"address": "19WHEZeTgJrkkQcHt45bQfZ69AohZNRyXN", "value":
    {"inputs": [{"address": "12xoxDH6p3J21483edYNuSQVKEHYrvbe6o", "value": 5433240000}], "outputs": [{"address": "193mk7oHV1srmX8ZfA8Fny9x57j6N48xN2", "value":
    {"inputs": [{"address": "15PwAfZAUXkQ3vYvWW5qJWE3beCmD1yvJg", "value": 3858635884}], "outputs": [{"address": "19FuCr7ByuyVYmrXu1x2aMeTZoLFrss3AY", "value":
    {"inputs": [{"address": "1NXsoaow3wFrULfkTY4g6ogxaRDLdU1Xdy", "value": 685092416}], "outputs": [{"address": "1DfW39nfdi5bioqamHmEU5ttKeru9yW5Zs", "value": 4
    {"inputs": [{"address": "1D9kd8ws2Fp5nZ8pZKFZXTQuLd5EjVHRrk", "value": 2225539301}], "outputs": [{"address": "1J6jNmgbLYtremVLhd1ZhF5ELaarWZWNjo", "value":
    {"inputs": [{"address": "15QC1yA4PymkZPdZX3kZPsjmmn4pBRuqgg", "value": 3209104722}], "outputs": [{"address": "1NmucA4rsfFTjAxDCStepBLsx446mQA5En", "value":
    {"inputs": [{"address": "1FZjzoavUBs6adJ896i9Rp85TPAJXcmASA", "value": 500000000}], "outputs": [{"address": "1DRw6oGD8vjW7RQVx6FipV7R5tVayLdHGn", "value": 2
    {"inputs": [{"address": "15yYzuF4rq5NsR4hdY6LDjLDHsUxqSKUhs", "value": 2349776871}], "outputs": [{"address": "1Atp9Jpe8qHgQTQhkcZBFRg46nrv94mCHD", "value":
    {"inputs": [{"address": "1PxhvdrRFXbFTNGHVFNhkFZk4oowjQ97nt", "value": 2320466372}], "outputs": [{"address": "1D71Z5NP1rTfq8GpcPgANX3XRX9JY4491t", "value":
    {"inputs": [{"address": "12Lidd61dPyfXTtdwovifszzHXUxggjze8", "value": 1900000}, {"address": "1AwMxCQ4TaGV9nb4d7TeWbHdfVj6xniZmf", "value": 1950000}, {"addre
    {"inputs": [{"address": "1KcZVhz7rzGQPK2e7AXwyJAhs3wgDjmzyx", "value": 252104660}], "outputs": [{"address": "1GCt8BHecb7SdgUBk9GKTRGkEJPxXEN84M", "value": 1
    {"inputs": [{"address": "1L5bFoLvEAUCu328tWBZKJjcpkBdQDBPSA", "value": 1933475400}], "outputs": [{"address": "1PJTS7WAmj6fbbd2cBoVA2JBqQrTu9Kkyg", "value":
    {"inputs": [{"address": "1AkuDUV4K3LKHETFCGgGuypQQj5Y2BqDXf", "value": 1921410513}], "outputs": [{"address": "12y4nppK7JidVNTZZ6ghN4xB6vMrRCrKx1", "value":
```

Figure 4: blockchain after process

4.2 Mt.Gox leaked DataSet

The Mt.Gox leaked DataSet don't need to change too much, just change the time into the form of seconds since 1970.1.1 which is easily to compare with other time. Then extract useful parts of it.

5 Network De-Anonymization

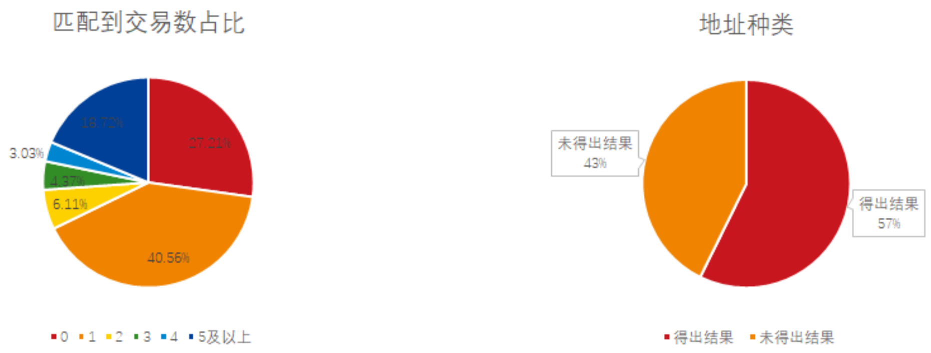
5.1 Algorithm

For each transaction (source, target) in the Mt.Gox data set, looking for a transaction (input, output) in the revenue and expenditure network with the same amount of the bitcoin, while the difference between the transaction time is less than 6 block times. When such a transaction is found, it means that source and input, target and output might belong to the same user, and their confidence degree C_{ij} should plus 1.

Traverse all transactions in Mt.Gox data set, calculate in the above way, the confidence between each account and each address is obtained. For any bitcoin address, find out which Mt.Gox account has the highest level of confidence, then we think the address belongs to this account. If there is no match or there are more then one account with the highest confidence, it will not be assigned, or we say it is not de-anonymized.

5.2 Result

The whole data set is too large for my computer to handle. So I choose all the transaction in 2011.7.19 for experiment. The final result is shown below.



(a) Match Number (b) De-Anonymization rate

Figure 5: Result

As you can see, 40% of the transactions in the data set match the only transactions in the bitcoin network. In addition, 30% of transactions match more than one transaction, probably because there are many integer values that often appear in the transaction (such as: 1, 10,100, etc.)

After complete the algorithm, nearly 60% of the addresses are de anonymous. The rest 40% of unsuccessful de-anonymization is because that the transactions in the dataset do not cover all transactions in blockchain and there are transactions that do not match the address.

6 Future Work

Based on the number of transactions between different addresses, clustering algorithms such as K-means can be used to cluster the addresses. By observing whether the addresses belonging to the same user in the de-anonymization result are in the same cluster, we can analyze the effectiveness of the de-anonymization result.