A Traceability Method for Bitcoin Transactions Based on Gateway Network Traffic Analysis

Dapeng Huang, Haoran Chen, Kai Wang, Chen Chen, Weili Han
Laboratory for Data Analytics and Security, Fudan University
Email:{18110240052, haoranchen20, wangk20, chenc, wlhan*}@fudan.edu.cn

2022 International Conference on Networking and Network Applications (NaNA) | 978-1-6654-6131-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/NANA56854.2022.00037

Abstract—Cryptocurrencies like Bitcoin have become a popular weapon for illegal activities. They have the characteristics of decentralization and anonymity, which can effectively avoid the supervision of government departments. How to de-anonymize Bitcoin transactions is a crucial issue for regulatory and judicial investigation departments to supervise and combat crimes involving Bitcoin effectively. This paper aims to de-anonymize Bitcoin transactions and present a Bitcoin transaction traceability method based on Bitcoin network traffic analysis. According to the characteristics of the physical network that the Bitcoin network relies on, the Bitcoin network traffic is obtained at the physical convergence point of the local Bitcoin network. By analyzing the collected network traffic data, we realize the traceability of the input address of Bitcoin transactions and test the scheme in the distributed Bitcoin network environment. The experimental results show that this traceability mechanism is suitable for nodes connected to the Bitcoin network (except for VPN, Tor, etc.), and can obtain 47.5% recall rate and 70.4% precision rate, which are promising in practice.

Index Terms—Bitcoin Transactions, Traceability Method, Traffic Analysis

I. Introduction

Cryptocurrencies like Bitcoin have been widely used as payment tools in recent years and attract the attention of users and researchers. Digital asset management firm CoinShares shows that inflows into bitcoin products and funds hit a record $6.4 billion as of November 2021. Bitcoin's anonymous and decentralized nature makes it is difficult to be regulated by the government. At the same time, Bitcoin is also widely used by criminals. With the help of cryptocurrencies, the crimes of money laundering, fraud, and crypto-extortion involving cryptocurrencies are increasing dramatically. Analyzing and tracing the Bitcoin transaction data are the keys for government departments to supervise illegal activities involving cryptocurrencies effectively. Our proposed transaction traceability mechanism can analyze and track the creator's identity of a specific transaction. This mechanism helps to improve the regulator's ability such as malicious transaction tracking and special transaction discovery.

In the existing research on Bitcoin transaction regulation technology, traceability based on the analysis of Bitcoin network data flow is one of the most important research directions[1]. However, existing traceability technologies have low precision and poor practicability. In order to improve the precision and practicability, the main contributions of this paper are as follows:

1) Based on gateway network traffic analysis, our method can trace Bitcoin transactions in a specific range of Bitcoin networks and associate Bitcoin transaction hashes with the IP addresses of transaction originating nodes.
2) Associate the IP address of transaction originating node with the input transaction address.
3) The general Bitcoin network nodes are suitable for this traceability mechanism except for the use of VPN or Tor technologies. The mechanism can achieve traceability precision of 47.5% recall rate and 70.4% precision rate, which is better than the existing traffic analysis based on the Bitcoin network transaction traceability methods.

The rest paper is organized as follows: In Section II we provide the transmission approach of Bitcoin transactions and the related works. In Section III we describe the details of our traceability method. The collection of datasets and the experiment's environment were showed in Section IV, the results of our experiment were evaluated in section V. In Section VI we outline our conclusions and future work.

II. Background and Related Work

A. Background

Bitcoin uses an Internet-based P2P network architecture which is decentralized[2]. Each node in the network both provide and use resources. Although the various nodes in the Bitcoin network are equal to each other. They may have a different division of labor depending on the functions provided. The most common types of nodes in the Bitcoin network are as follows: core clients include wallets, miners, complete blockchain databases, network routing nodes, and complete blockchain nodes. Wallet nodes can be divided into Lightweight (SPV) Wallet and Lightweight (SPV) Stratum Wallet. Miners nodes can be divided into independent miners nodes, mining pool protocol servers, and mining nodes[3].

Two types of core clients and lightweight(SPV) wallets[4] can create Bitcoin transactions. They can be divided into static IP address nodes and dynamic IP

978-1-6654-6131-3/22/$31.00 ©2022 IEEE
DOI 10.1109/NaNA56854.2022.00037

address nodes. The static IP address nodes are the backbone nodes of the Bitcoin network. They are online for a long time and provide external services such as information forwarding, verification transactions, etc., and maintain complete blockchain data files, which can initiate transactions; The dynamic IP address nodes have a short online time, mainly for initiating transactions.

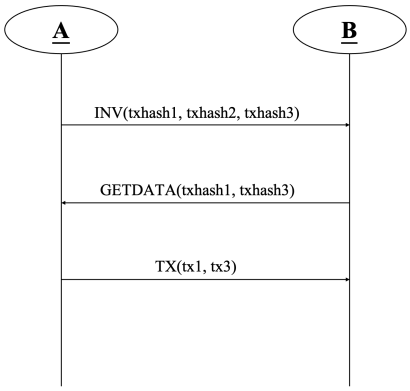

Fig. 1: Bitcoin transaction message transmission diagram

The brief flow of Bitcoin transaction sending is shown in Fig. 1. When node A sends transaction information to node B, it first sends the inv message, which contains a number of transaction hash lists that node A is going to broadcast to B. After B receives the inv, it returns the getdata response message to node A. The message contains the transaction hash list required by B. After node A receives the transaction hash list in getdata message, it uses the TX message to send these transactions' content to node B. Nodes that receive transactions, after validating those transactions, will also forward the transactions in the same way. It is worth mentioning that the code of the transaction hash contained in the inv message and the getdata message needs to be converted before it can correspond to the transaction hash on the Bitcoin blockchain. For example, the transaction hash code contained in the Getdata message is a4aea61c6a23ae8d13c16b7f629e53cd518674525a76cba45ece2c66709426b7, and the corresponding transaction hash code on the blockchain is b7269470662cce5ea4cb765a52748651cd539e627f6bc1138dae236a1ca6aea4.

B. Related Work

In order to maintain the stable operation of the Bitcoin network, Bitcoin static IP nodes usually need to provide external services, such as helping other client nodes to connect to the Bitcoin network, forwarding transaction information and verifying transactions. Therefore, Bitcoin static IP nodes usually accept connection requests initiated by any other node and will broadcast transaction information to these connected nodes. The Bitcoin transaction traceability technology based on Bitcoin network data analysis is to use the openness of the Bitcoin network to monitor data by joining the Bitcoin network through special nodes, collect transaction information forwarded in the network, and infer the broadcast path of transaction information in the network.[5][6] The origination node of the transaction to realize the traceability of the Bitcoin transaction. The Bitcoin system introduces two mechanisms: delayed forwarding and blacklisting to increase the difficulty of traceability. Delayed forwarding means that Bitcoin nodes use different random delays when forwarding transactions to prevent attackers from distinguishing between originating nodes and non-originating nodes by using the difference in transaction time points.

Traffic analysis is applied in many other fields.[7][8][9][10] In Bitcoin, Abu et al[11] leveraged the Bitcoin traffic to determine the nodes' states. Guo et al. [12]proposed an efficient Bitcoin client tracing mechanism to trace from Bitcoin server to the client through traffic analysis. Huang et al.[13] proposed a malicious node detection method based on behavior pattern clustering, which can quickly locate and eliminate malicious nodes. Imtiaz et al.[14] provided experimental evidence that the vast majority (97%) of Bitcoin nodes exhibit only intermittent network connectivity.Gervais et al.[15] introduced the Bitcoin blacklist mechanism in detail. The blacklist mechanism refers to the behavior of the Bitcoin system to identify abnormal other nodes in the system. If the node harms the operation of the network, it will be blacklisted to prevent the connection of such nodes.

Tracing technology based on special broadcast mode: It refers to inferring the originating node of the transaction by analyzing the broadcast law of transaction information in the network layer and using the special broadcast mode generated in some special circumstances. For example, Koshy et al. [16] analyzed the broadcast law of Bitcoin transactions in the network layer and found that normal transaction information will be forwarded by multiple nodes once in the blockchain network, while transactions with problematic transaction formats will only be sent by the originating node. Once forwarded, the originating node can be inferred through this special forwarding mode. However, the proportion of transactions with special broadcast patterns is small, and the proportion of the experimental results of the paper is less than 9%. This traceability technology is less practical.

Tracing technology based on transaction broadcast path: It refers to analyzing the broadcast path of the transaction by collecting the information transmitted by the blockchain transaction at the network layer so as to track the IP address of the server that created the transaction. Kaminsky[17] proposed at the Black Hat Conference in 2011 that "the first node that tells you a transaction may be the originating node of the transaction". Analysts only need to connect as many Bitcoin server nodes as

possible and record the transaction information forwarded from different nodes, and then they can determine that the node that forwards the information to the probe first is the originating node. This method only relies on the first node as a judgment feature, while the precision is low. Biryukov et al.[18]proposed a transaction traceability mechanism based on neighbor nodes, which can improve the traceability precision by using neighbor nodes as a judgment basis. However, the solution needs to continuously send a large amount of transaction information to all nodes in the Bitcoin network, which is prone to serious interference in the Bitcoin network and is less practical.

III. A Traceability Method for Bitcoin Transactions Based on Gateway Network Traffic Analysis

The Bitcoin network is a P2P network that operates on the Internet. The current Internet is composed of sub-networks dominated by a star structure. These sub-networks are connected through gateway devices such as routers and switches. Most large enterprises and institutions have their own Intranet which connects to the Internet. So these gateway devices often become the only way for the sub-networks to access the Internet. Fig. 2 shows the traceability system architecture of Bitcoin transactions based on gateway network traffic analysis. The sub-networks in the figure converge upward in a star-shaped structure, mirroring the traffic of the core switch to our parsing server. The parsing server parses and records the network traffic data to a Bitcoin log which will be stored in the log server.

In order to track the Bitcoin transactions effectively, we design the traceability technology based on log storage and query on the physical link of the Bitcoin network: by using switch mirroring, network traffic parsing server, log database server and other equipment. The network log is formed by analyzing the transmission information of the Bitcoin network. The log server record only some key information but all network traffic data to reduce the requirment of harddisk.

For the convenience of expression, this paper refers to the network covered by the traceability system as a network jurisdiction. The traceability mechanism in this paper aims to trace the transaction source of Bitcoin nodes within a specific range, that is, identifying the transaction information originating node in the network jurisdiction.

Each node in the Bitcoin network has no more than 125 neighbor nodes. Nodes keep pace with the Bitcoin network by exchanging information with their neighbors. A 2-level node refers to the neighbor node of the neighbor node. By analogy, starting from the originating node, the entire Bitcoin network can be divided into neighbor nodes, 2-level neighbor nodes, ..., n-level neighbor nodes.

When a Bitcoin node initiates a transaction, it will immediately broadcast the transaction to its neighbor nodes. After the neighbor node verifies the transaction, it will broadcast to the next-level neighbor nodes according

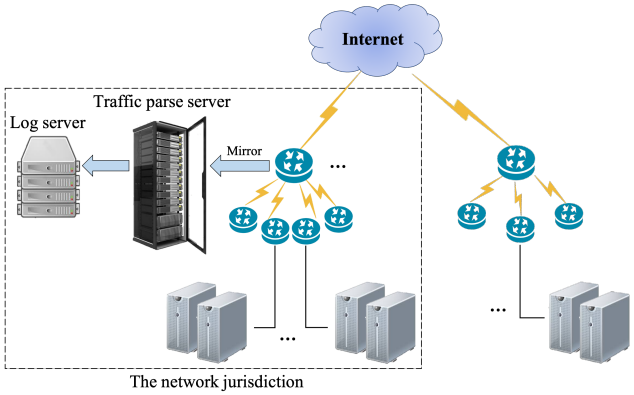

Fig. 2: The architecture of traceability system

to the Bitcoin system Trickling or Diffusion forwarding strategy[19]. Then the transaction is packaged into the blockchain file by the miner node. The transaction will continue to be broadcasted until it reaches every node in the Bitcoin network. Even the Bitcoin system adopts the delayed forwarding strategy, the initiation time of the Bitcoin transaction must be earlier than the time when the transaction is written into the block file, that is, the transaction confirmation time. Before the transaction is written to the block file, the transaction must be broadcasted on the Bitcoin network for a period of time, and we call the transaction broadcasted on the network during this period as an unconfirmed transaction. The traceability system will record this transaction, and the log recording time is between the initiation time and the transaction confirmation time. This paper refers to this recording time as the logging time of unpackaged transactions. We cannot obtain a large number of Bitcoin transactions and the initiation time of transactions, but we can obtain a large number of Bitcoin transactions and the corresponding log time in the traceability system.

Without considering the network delay, data packet leakage and other interference factors, we can analyze the transaction broadcast time and broadcast path in the area covered by the traceability system according to these logs and speculate that the originating node and its neighbor nodes. The transaction, as well as the relevant Bitcoin input address and the IP address, port and time of the originating node are associated with forming a data table as shown in Table I.

TABLE I: Bitcoin transaction network traffic log

logtime	IP_{scr}:port	IP_{dst}:port	txhash
21-9-6 05:09:01:01	202.*.*.130:8333	187.*.*.25:3504	0c76...e482
21-9-6 06:23:11:13	202.*.*.130:6486	154.*.*.187:8333	bd71...3dd1

There are about 9,709 nodes with fixed IP addresses in the Bitcoin network, generating about 388,000 transaction records per day (data in May 2021 https://bitnodes.io/). Analyzing the daily flow of tens of thousands of Bitcoin transactions requires enormous computing and storage resources. This paper adopts three methods to reduce the consumption of computing resources.

1) Only focus on transactions sent out from the jurisdiction (Fig. 2).
2) Only focus on transactions that are earlier than the confirmed time on the chain
3) Only focus on the transaction hash carried by the Getdata message as it is a response to the INV message. The number of transaction hashes it contains is far less than the transaction hash carried by the inv message and the data volume of the TX message, but only the direction of transmission is reversed.

The following describes the operation steps in detail:

1) Determine the transaction set of the pure output of this network. By recording the Bitcoin transaction set TX_{in} entering the jurisdiction and the Bitcoin transaction set TX_{out} leaving the jurisdiction, the transaction set with the same transaction hash in the intersection of TX_{out} and TX_{in} and whose TX_{out} log time is earlier than the corresponding TX_{in} log time is recorded as $(\mathrm{TX}_{out} \cap \mathrm{TX}_{in})'$, record the transaction set purely from this switch as $\mathrm{TX}_{pureout}$, that is shown in Equation 1

$$TX_{\mathrm{pureout}} = (TX_{\mathrm{out}} - TX_{\mathrm{in}}) \cup (TX_{\mathrm{out}} \cap TX_{\mathrm{in}})' \quad (1)$$

2) Determine the set of transactions initiated by the network earlier than the confirmed time. When a Bitcoin transaction is initiated in the network of the jurisdiction, the initiation time must be earlier than the time of interception by the traceability system, and the time when the traceability system intercepts the unconfirmed transaction must be earlier than the confirmation time of the transaction on the blockchain. According to the transaction hash in $\mathrm{TX}_{pureout}$, find the corresponding transaction record on the blockchain to extract the confirmation time of the transaction. The pure outgoing transaction set $\mathrm{TX}_{pureout}$ excludes the transaction set $\mathrm{TX}_{>=blocktime}$ that is later than the confirmation time stamp and forms the pre-confirmation time. The pure outgoing transaction set TX_{early}, that is shown in Equation 2:

$$TX_{\mathrm{early}} = TX_{\mathrm{pureout}} - TX_{>=\mathrm{blocktime}} \quad (2)$$

3) Determine the traceability target in TX_{early}. Calculate the earliest log recording time of different transactions for the transaction set TX_{early}, form a quadruple of Bitcoin transactions (log time, transaction hash, source IP: port, destination IP: port), and calculate the earliest occurrence of the same transaction through time sorting on which IP node. The input address is parsed according to the transaction content, and a quadruple of the transaction input address is formed (logtime, input address, source IP:port, destination IP:port).
4) Calculate the matching degree. We denote the time when the Bitcoin transaction tx_i is confirmed on the blockchain as $\mathrm{T}(\mathrm{tx}_i)$, and the time when tx_i is sent from IP_{scr} to IP_{dst} is recorded by the traceability system as $\mathrm{TR}(\mathrm{tx}_i,\mathrm{IP}_{scr},\mathrm{IP}_{dst})$. $\mathrm{TR}(\mathrm{tx}_i)$ indicates the earliest time the transaction tx_i was recorded by the traceability system. $\mathrm{T}(\mathrm{tx}_i)$-$\mathrm{TR}(\mathrm{tx}_i,\mathrm{IP}_{scr},\mathrm{IP}_{dst})$ is expressed as the difference between the two. As shown in Equation 3, the value of $\mathrm{P}(\mathrm{tx}_i,\mathrm{IP}_{scr},\mathrm{IP}_{dst})$ is less than or equal to 1, and the earliest recorded transaction has $\mathrm{P}(\mathrm{tx}_i, \mathrm{IP}_{scr}, \mathrm{IP}_{dst}))=1$.

$$P(tx_i, IP_{scr}, IP_{dst}) = \frac{T(tx_i) - TR(tx_i, IP_{scr}, IP_{dst})}{T(tx_i) - TR(tx_i)} \quad (3)$$

The transaction tx_i will be sent to different IP_{dst}) addresses from the same IP_{scr}, so $\mathrm{P}(\mathrm{tx}_i, \mathrm{IP}_{scr})$ represents the synthesis of all the propagated P values sent by tx_i from IP_{scr}, as shown in Equation 4.

$$P(tx_i, IP_{scr}) = \sum_{j=1}^{n} P(tx_i, IP_{scr}, IP_{dst_j}) \quad (4)$$

5) Output transaction hash tuple.
The method used in this paper tests the relevant thresholds, selects the best threshold according to the precision, recall and F value in the actual network environment, and outputs the tuples of Bitcoin transactions (tx_i, IP_{scr}). When $\mathrm{P}(\mathrm{tx}_i, \mathrm{IP}_{scr})$ is greater than or equal to the threshold P-Value, the system outputs the tuples of tx_i to the next link and calculates the $\mathrm{K}(\mathrm{input}_i, \mathrm{IP}_{scr})$ of the Bitcoin input address input_i corresponding to tx_i; otherwise continue to detect the network jurisdiction's Bitcoin network log.
6) Output transaction address tuple.
The initial value of $\mathrm{K}(\mathrm{input}_i, \mathrm{IP}_{scr})$ is 0, and the corresponding $\mathrm{P}(\mathrm{input}_i, \mathrm{IP}_{scr})$ of all inputs in tx_i are superimposed to $\mathrm{K}(\mathrm{input}_i, \mathrm{IP}_{scr})$ as Equation 5, once $\mathrm{K}(\mathrm{input}_i, \mathrm{IP}_{scr})$ is greater than or equal to the threshold K-Value , output the input address matching tuple of the suspected Bitcoin transaction; otherwise, continue detecting. The matching address tuple consists of tuple(input address, IP_{scr}).

$$K(input_i, IP_{scr}) = K(input_i, IP_{scr}) + P(input_i, IP_{scr}) \quad (5)$$

IV. Data collection and expriment

A. Acquisition time settings

To control the size of the TX_{early} dataset, we count the interval between Bitcoin transaction initiation and

confirmation. We found through the traceability system that the time difference between the log interception time of the unconfirmed transaction and the transaction confirmed time is shown in Fig. 3.

Through the traceability system, we obtained 20,052 transactions that were earlier than the blockchain confirmation timestamp from 0:00 on May 10, 2020 to 0:00 on May 20, 2020. The abscissa of Fig. 3 is the recording time in log file, and the ordinate is the delay between transaction occurrence and confirmation, which equals the confirmation time on the blockchain minus the logging time. The highest interval was 86395 seconds, the lowest interval was 6 seconds, and the average was 31898.18 seconds. Satoshi Nakamoto did not specify the interval between transaction initiation and confirmation time in the white paper. According to the statistical results of this data and combined with people's daily transaction habits, we retain the network log data of Bitcoin transactions for 24 hours, in order to reduce the cost of the traceability system computing resources and storage resources. The transaction log captured exceeds 24 hours after the transaction occurs is considered to be a confirmed transaction log and should be discarded.

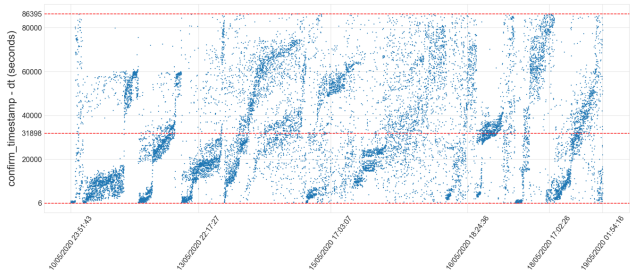

Fig. 3: The delay between transaction occurrence and confirmation

According to the Bitcoin transaction traceability mechanism based on the analysis of Bitcoin network traffic, we established a set of transaction traceability system, the topology structure is shown in Figure 2, and the traceability effect on the Bitcoin network was tested. The traceability system includes a traffic parse server and a log processing server. The traffic parse server can collect the network data flow of the Bitcoin network and store the parsed Bitcoin network logs in the log processing server. The log processing server processes these logs according to the log data and the data on the chain.

B. Experimental data

The threshold P-Value is used to determine whether a transaction is initiated by a node. When the $P(tx_i, IP_{scr})$ of a transaction exceeds this threshold, the transaction is considered to be initiated by the node. Thresholds are obtained experimentally.

The IP address of the node and the hash of the transaction sent are known during the experiment. These transaction network data streams are parsed, recorded and processed by the system. The network status and network delay of each Bitcoin node are different, and the threshold calculated in the experimental environment can be used as a reference. This paper uses the Bitcoin core to initiate transactions in the network jurisdiction.

In order to study the characteristics of the log record transaction initiating node and its neighbor nodes, this paper conducted 40 transactions and designed two statistical items: (1) The probability that the transaction initiating node arrives at the traceability system at the earliest; (2) The probability that the eight neighbor nodes connected by the initiating node reach the traceability system at the earliest. The results are shown in Table II.

TABLE II: Probability comparison of the first record of the node

NO.	number of tx	initial node	neighbor node
1	5	5	0
2	5	4	0
3	5	5	0
4	5	1	2
5	5	1	1
6	5	0	2
7	5	0	2
8	5	3	0
Probability		47.5%	17.5%

The experiment did 8 groups of tests, and each group sent 5 transactions. Table II records the number and probability of the first captured by the traceability system for transactions sent by the originating node and its neighbors. Among them, the earliest probability of the transaction originating node being recorded by the traceability system is 47.5%, and the earliest probability of the neighbor node being recorded is 17.5%.

The experiment initiates 40 transactions through known nodes and uses the traceability system to record these 40 transactions. According to the difference between the log time of the transaction hash recorded in the log and the confirmed time of the transaction on the blockchain, the P-Value is calculated according to Equation 3. Table III records the results of 27 transactions recorded by the logging system. Among them, 1-15 transactions are sent by the same node, and 16-27 transactions are initiated by different nodes. Each row of data contains transaction id, source IP, source port, destination IP, destination port, and P-value. Accoring to the results shown in Table III, we can conclude as follows:

1) Transactions 1 to 15 were initiated by the same known node, and the traceability system found and recorded the transaction hash sent by the node and the log time was earlier than the confirmed time in the blockchain for 22 times. The 10th and 15th transactions are

the transactions forwarded by the neighbor nodes of the known node. They reached the traceability system firstly. The remaining 13 transactions are the node that initiates the transactions, and they reach the traceability system firstly. The log time of the traceability system is an important basis for judging whether the transaction is the first launch of the node.

2) 6 of the 15 recorded transactions were recorded by the traceability system multiple times and sent from the same source IP to different destination IP nodes. They are 3rd, 6th, 7th, 9th, 10th, 14th, and 5 of them were sent by known nodes, which have the earliest log time. The same node sends the same transaction hash to different IP nodes, and the earliest record that reaches the traceability system is also sent by this node. This rule is used to judge whether the node transaction is a more stringent condition for the node to issue first, and its precision rate is higher.
3) 16-27 transactions are initiated by a different IP node for each transaction, and each transaction initiating node will not initiate a transaction for a long time after sending a transaction. The traceability system finds and records these transaction hashes, which log time is earlier than the confirmed time for 17 times. The 17th, 20th, 22nd, 23rd, 24th, 25th and 26th transactions are the transactions forwarded by the neighbor nodes of the known node. They firstly reach the traceability system. The remaining 10 transactions were initiated by the node that initiated the transaction and reached the traceability system at the earliest. The number of times a node initiates transactions may affect data requests from nodes outside its jurisdiction to the initiating node, thereby affecting the records of the traceability system.

In the experiment, the threshold P-Value is set to 1. According to the Equation 4, the two-tuple(tx_i, IP_{scr}) of Bitcoin transactions with P(tx_i,IP_{scr}) greater than 1 is set. Based on the data of the two-tuple (tx_i, IP_{scr}), the experiment analyzes the input address in the transaction and tests the impact of the value of the threshold K-Value on the traceability of the input address of Bitcoin transactions.the data (Table IV) related to the input address was obtained experimentally.The 'input' is the input address corresponding to the transaction hash in the two-tuple (tx_i, IP_{scr}), the 'IP_{scr}' is the source IP address of the transaction, and 'sumK' is the sum of the K values with the same input address and source IP.

V. Performance Evaluation

A. The influence of different threshold P-Value on transaction traceability

In order to test the precision and recall of the traceability mechanism based on the network log system under different thresholds, find the optimal threshold. In this paper, we tested 40 transactions in the experimental environment, and the test results are as follows.

TABLE III: Bitcoin transaction transfer log and P-value

#	txhash	IP_{scr}:port	IP_{dst}:port	P-value
1	022e...7367	115.*.*.161:13749	96.*.*.143:8333	1
	022e...7367	60.*.*.86:57909	91.*.*.5:8333	0.995074
2	081e...64cd	60.*.*.86:57909	91.*.*.5:8333	0.991667
	081e...64cd	115.*.*.161:8333	109.*.*.13:14149	1
3	10d1...20fa	115.*.*.161:13354	195.*.*.8:8333	1
	10d1...20fa	115.*.*.161:8333	3.*.*.253:13354	1
	10d1...20fa	60.*.*.86:57829	34.*.*.226:8333	0.980663
	10d1...20fa	115.*.*.161:13354	35.*.*.134:8333	1
4	18fe...a468	115.*.*.161:8333	40.*.*.208:14283	0.904762
	18fe...a468	115.*.*.161:8333	40.*.*.208:14283	1
5	29a3...f3c5	60.*.*.86:57909	91.*.*.5:8333	0.992278
	29a3...f3c5	115.*.*.161:8333	109.*.*.13:14149	1
6	2f04...8ba0	36.*.*.187:49849	202.*.*.130:8333	0.706161
	2f04...8ba0	60.*.*.86:57792	194.*.*.205:8333	1
	2f04...8ba0	115.*.*.161:8333	109.*.*.153:14497	1
	2f04...8ba0	115.*.*.161:8333	195.*.*.8:13354	0.990521
7	33f8...aeb3	115.*.*.161:8333	195.*.*.8:13354	0.995017
	33f8...aeb3	115.*.*.161:8333	40.*.*.208:14283	0.995017
	33f8...aeb3	60.*.*.86:58003	3.*.*.253:8333	0.996678
	33f8...aeb3	115.*.*.161:8333	109.*.*.13:14149	1
8	42a7...d62a	115.*.*.161:8333	109.*.*.153:14497	1
9	753e...1bbd	115.*.*.161:8333	109.*.*.13:14149	0.998814
	753e...1bbd	60.*.*.86:57909	91.*.*.5:8333	0.989324
	753e...1bbd	115.*.*.161:8333	40.*.*.208:14283	1
	753e...1bbd	115.*.*.161:8333	109.*.*.13:14149	1
	753e...1bbd	115.*.*.161:8333	109.*.*.13:14149	0.998814
10*	7583...fe32	115.*.*.38:31736	149.*.*.83:8333	0.999494
	7583...fe32	202.*.*.130:28726	218.*.*.98:8333	1
	7583...fe32	202.*.*.130:56477	77.*.*.195:8333	1
11	7986...1c38	115.*.*.161:8333	109.*.*.153:14497	1
	7986...1c38	60.*.*.86:57909	91.*.*.5:8333	0.997722
12	7998...73fa	60.*.*.86:57909	91.*.*.5:8333	1
	7998...73fa	115.*.*.161:8333	40.*.*.208:14283	1
13	7ee4...4543	115.*.*.161:8333	195.*.*.8:13354	1
14	868f...f2bb	115.*.*.161:8333	195.*.*.8:13354	1
	868f...f2bb	60.*.*.86:57909	91.*.*.5:8333	0.997389
	868f...f2bb	115.*.*.161:8333	109.*.*.13:14149	0.997389
15*	8918...88fd	202.*.*.130:8333	54.*.*.88:56652	1
16	9395...2375	115.*.*.78:17721	94.*.*.119:8333	1
17*	9cc0...5806	61.*.*.106:62002	46.*.*.88:8333	1
18	a6dd...37e8	115.*.*.7:8333	47.*.*.169:13905	1
19	b416...d5da	115.*.*.62:8333	66.*.*.243:22236	1
	b416...d5da	61.*.*.107:61341	93.*.*.162:8333	1
20*	b995...ae8e	60.*.*.86:57792	194.*.*.205:8333	1
21	c7dd...ae80	60.*.*.86:57909	91.*.*.5:8333	1
	c7dd...ae80	115.*.*.161:8333	195.*.*.8:13354	1
22*	cd32...cc4d	60.*.*.86:57792	194.*.*.205:8333	1
23*	d461...1241	60.*.*.86:57792	194.*.*.205:8333	1
24*	d701...464a	60.*.*.86:57792	194.*.*.205:8333	1
25*	fab5...ada9	202.*.*.130:57659	176.*.*.132:8333	1
26	ff81...ba43	115.*.*.61:18147	188.*.*.201:8333	0.99604
	ff81...ba43*	157.*.*.69:8333	125.*.*.42:11299	1
27	e4fd...d414	202.*.*.130:40150	8.*.*.87:8333	0.968944
	e4fd...d414	202.*.*.130:40150	50.*.*.27:8333	0.968944
	e4fd...d414	115.*.*.7:8333	47.*.*.169:13905	1

TABLE IV: Bitcoin input address transfer log and K-value

input	IP_{scr}	sumK
34PT....MQcG	60.*.*.86	1
37L1...JaMh	115.*.*.62	1
37L1...JaMh	115.*.*.78	1
37L1...JaMh	60.*.*.86	1
37L1...JaMh	61.*.*.107	1
3FQg...9Zwz	115.*.*.7	1
3FQg...9Zwz	115.*.*.161	4.904762
3FQg...9Zwz	60.*.*.86	2
bc1q...nuul	115.*.*.7	1
bc1q...rkt8	115.*.*.161	1.990521
bc1q...rkt8	60.*.*.86	1
bc1q...yt8m	115.*.*.161	1
bc1q...6z8v	115.*.*.161	2.990034
bc1q...t4t8	115.*.*.161	1
bc1q...qhet	202.*.*.130	1
bc1q...9e7a	60.*.*.86	1
bc1q...8j62	202.*.*.130	1
bc1q...dcl3	60.*.*.86	1
bc1q...n2v6	157.*.*.69	1
bc1q...3jph	115.*.*.161	1.997389
bc1q...avv9	60.*.*.86	1
bc1q...uwtp	202.*.*.130	1
bc1q...ddu0	115.*.*.161	1
bc1q...ddu0	60.*.*.86	1
bc1q...k2zh	115.*.*.161	3
bc1q...85g0	115.*.*.161	2.998814
bc1q...7rxz	115.*.*.161	1
bc1q...eh88	115.*.*.161	1
bc1q...l90r	61.*.*.106	1
bc1q...3wnv	115.*.*.161	1
bc1q...5x4z	61.*.*.106	1
bc1q...azu0	202.*.*.130	2

The traceability precision and recall described in Table V vary with the change of the threshold value. When the threshold value is greater than or equal to 2, it means that a node sends transactions to different nodes outside its jurisdiction more than three times. One of the transactions is the earliest captured by the traceability system, and its log time is earlier than its confirmed time in the blockchain. The precision rate is 100%, but the recall rate is relatively low, below 7.5%;

when the threshold value is 1.5, it means that a node sends transactions to different nodes outside its jurisdiction more than two times. One of the transactions is the earliest captured by the traceability system, and its log time is earlier than its confirmed time in the blockchain. The precision rate is 85.7%, but the recall rate is 15%;

when the threshold value is 1, it means that a node sends transactions to different nodes outside its jurisdiction more than one time. One of the transactions is the earliest captured by the traceability system, and its log time is earlier than its confirmed time in the blockchain. The precision rate is 70.4%, the recall rate is 47.5%, and the F value obtains the highest value of 56.7%.

According to different traceability requirements, we set different thresholds. In the actual measurement environment, the F value is the highest, and the threshold value is set to 1 as the optimal solution to screen out a large number of suspected originating transactions in the jurisdiction.

TABLE V: Precision and recall of transaction hash traceability with P-Value

P	samples	outputs	correct	precision	recall	F
1	40	27	19	70.4%	47.5%	56.7%
1.5	40	7	6	85.7%	15%	25.5%
2	40	3	3	100%	7.5%	14%
2.5	40	2	2	100%	5%	9.5%
3	40	2	2	100%	5%	9.5%

B. The influence of different threshold K-value on the traceability of the input address

Based on the data of the two-tuple (tx_i, IP_{scr}), the experiment analyzed the input address in the transaction and tested the influence of different threshold K-Values on the precision and the recall of input address traceability, and find the optimal threshold.

Table VI describes the difference in the precision and the recall of input address traceability with different thresholds. When the threshold is above 3, its precision is 100%, but the recall is low, 6.25%; when the threshold is 2.5, its precision is 100%, and the recall is 12.5%; when the threshold is 2, its precision is 66.7%, but the recall is still 12.5%; when the threshold is 1.5, its precision is 75%, the recall is 18.75%, and the value of F value obtains the highest value of 30%. Under the condition that the threshold P-Value is 1, for input address traceability, the threshold K-Value is 1.5 as the optimal solution.

TABLE VI: Precision and recall of transaction input address traceability with K-Value

K	samples	outputs	correct	precision	recall	F
1.5	32	8	6	75%	18.75%	30%
2	32	6	4	66.67%	12.5%	21.05%
2.5	32	4	4	100%	12.5%	22.22%
3	32	2	2	100%	6.25%	11.76%

In the existing research, a transaction tracking mechanism based on neighbor nodes has been proposed, and the neighbor nodes are used as the judgment basis to improve the tracking accuracy. However, this method needs to continuously send a large amount of transaction information to all nodes in the Bitcoin network, which will cause serious interference to the Bitcoin network and is not practical. The tracking method designed in this paper can solve the above problems well and be applied.

According to the design of Bitcoin, there are two types of forwarding transactions in network jurisdiction. One is that the originating node directly sends the transaction

to the neighbor nodes outside the jurisdiction; the other is that the originating node sends the transaction to the N-level node within the jurisdiction, and the N-level node forwards the transaction to the nodes outside the jurisdiction.

The experimental observation found that in the state of default settings, when a node with a non-fixed IP initiated a transaction, the node would directly send transactions to neighboring nodes outside its jurisdiction, rather than looking for neighbor nodes within its jurisdiction. The traceability system found that most transaction forwarding belonged to the former type. After the node transaction was generated, the transaction was sent to the nodes outside the jurisdiction at least 2 times and at most 10 times before the confirmed time in the blockchain.

When the non-fixed IP node can keep the IP address unchanged for a long time and be online for a long time, the non-fixed IP node would be connected to the fixed IP address node within its jurisdiction and takes it as its owned 1-level neighbor node. The experimental observation in this paper is that the length of time is more than 96 hours. Nodes with fixed IP, good network status, and long-term online in the jurisdiction had a greater impact on the experimental results.

The experiment found that most of the TX_{early} transaction data came from these 8 nodes except the originating node. By sorting the log time of the transaction and extracting the top 9 nodes, the originating node and its low-level neighbor nodes can be determined. Affected by the confirmed time limit, etc., in the experiment, the records of a transaction sent by different nodes in the jurisdiction to the nodes outside the jurisdiction are often less than 9 times.

VI. Conclusion and Future Work

Bitcoin has attracted widespread attention from industry researchers. The traceability method in this paper adopts the way of passively traffic data collection in the Bitcoin network, which had zero interference to the Bitcoin network. The method achieved high traceability precision that transactions initiated within our observation area. For our traceability method, there are still many works that need to be further researched. For example, it is important to reduce the number of network data collection nodes while meeting the system performance requirements. We also need to apply our mechanism to a more complexe network environment and improve its performance.

Acknowledgement

This paper is supported by the National Key R&D Program of China (No. 2019YFE0103800) and STCSM (No. 21511101600).

References

[1] P. Treleaven, R. G. Brown, and D. Yang, "Blockchain technology in finance," Computer, vol. 50, no. 9, pp. 14–17, 2017.

[2] F. Franzoni, X. Salleras, and V. Daza, "Atom: Active topology monitoring for the bitcoin peer-to-peer network," Peer-to-Peer Networking and Applications, vol. 15, no. 1, pp. 408–425, 2022.

[3] A. M. Antonopoulos, Mastering Bitcoin: Programming the open blockchain. " O'Reilly Media, Inc.", 2017.

[4] L. Zhou, C. Ge, and C. Su, "A privacy preserving two-factor authentication protocol for the bitcoin spv nodes," Science China Information Sciences, vol. 63, no. 3, pp. 1–15, 2020.

[5] S. Park, S. Im, Y. Seol, and J. Paek, "Nodes in the bitcoin network: Comparative measurement study and survey," IEEE Access, vol. 7, pp. 57 009–57 022, 2019.

[6] S. G. Motlagh, J. Mišić, and V. B. Mišić, "An analytical model for churn process in bitcoin network with ordinary and relay nodes," Peer-to-Peer Networking and Applications, vol. 13, no. 6, pp. 1931–1942, 2020.

[7] J. Holland, P. Schmitt, N. Feamster, and P. Mittal, "New directions in automated traffic analysis," in Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 3366–3383.

[8] E. Papadogiannaki and S. Ioannidis, "A survey on encrypted network traffic analysis applications, techniques, and countermeasures," ACM Computing Surveys (CSUR), vol. 54, no. 6, pp. 1–35, 2021.

[9] M. Abbasi, A. Shahraki, and A. Taherkordi, "Deep learning for network traffic monitoring and analysis (ntma): a survey," Computer Communications, vol. 170, pp. 19–41, 2021.

[10] P. Nerurkar, D. Patel, Y. Busnel, R. Ludinard, S. Kumari, and M. K. Khan, "Dissecting bitcoin blockchain: Empirical analysis of bitcoin network (2009–2020)," Journal of Network and Computer Applications, vol. 177, p. 102940, 2021.

[11] A. Abu Ghanem and S. Eftekhari, "A study of the network traffic between bitcoin nodes," 2021.

[12] W. Guo and J. Zhang, "Towards tracing bitcoin client using network traffic analysis," in 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP). IEEE, 2019, pp. 1–5.

[13] B. Huang, Z. Liu, J. Chen, A. Liu, Q. Liu, and Q. He, "Behavior pattern clustering in blockchain networks," Multimedia Tools and Applications, vol. 76, no. 19, pp. 20 099–20 110, 2017.

[14] M. A. Imtiaz, D. Starobinski, A. Trachtenberg, and N. Younis, "Churn in the bitcoin network," IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1598–1615, 2021.

[15] A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun, "Tampering with the delivery of blocks and transactions in bitcoin," in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015, pp. 692–705.

[16] P. Koshy, D. Koshy, and P. McDaniel, "An analysis of anonymity in bitcoin using p2p network traffic," in International Conference on Financial Cryptography and Data Security. Springer, 2014, pp. 469–485.

[17] D. Kaminsky, "Black ops of tcp/ip 2011," Black Hat USA, p. 44, 2011.

[18] I. Pustogarov, "Deanonymisation techniques for tor and bitcoin," Ph.D. dissertation, University of Luxembourg, Luxembourg, Luxembourg, 2015.

[19] A. Biryukov and S. Tikhomirov, "Transaction clustering using network traffic analysis for bitcoin and derived blockchains," in IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2019, pp. 204–209.