JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 40, 375-396 (2024)
DOI: 10.6688/JISE.202403_40(2).0011

A Bitcoin Transaction Analyzing and Tracking Mechanism in Specified Network Zone*

DAPENG HUANG, CHEN CHEN, HAOWEI LUO, KAI WANG AND WEILI HAN+
Laboratory for Data Analytics and Security
Fudan University
Shanghai, 200433 P.R. China
E-mail: {18110240052; chenc}@fudan.edu.cn;
luohw21@m.fudan.edu.cn; {wangk20; wlhan+}@fudan.edu.cn

Cryptocurrencies, such as Bitcoin, have emerged as a popular means for illicit activities due to their decentralized and anonymous nature, allowing them to circumvent governmental oversight effectively. To combat crimes associated with Bitcoin, it is crucial to address the issue of deanonymizing Bitcoin transactions. Accordingly, this paper presents a novel methodology for Bitcoin transaction traceability, based on Bitcoin network traffic analysis. Specifically, we analyze network traffic data obtained at the physical convergence point of the local Bitcoin network to trace the input address of Bitcoin transactions. The proposed scheme is tested in a distributed Bitcoin network environment, yielding a promising recall rate of 45% and precision rate of 66.67%, with the exception of nodes linked through VPN, Tor, and similar tools. This traceability mechanism holds significant practical implications for regulatory and judicial investigation departments.

Keywords: bitcoin transactions, traceability method, traffic analysis, cryptocurrencies, blockchain

1. INTRODUCTION

The boom in cryptocurrencies, represented by Bitcoin [1], has had a huge and far-reaching impact on current domestic and international financial markets. Digital asset management firm CoinShares shows that inflows into bitcoin products and funds hit a record $6.4 billion as of November 2021. Bitcoin, for example, once surpassed silver as the eighth largest asset in the world by market value [2]; and has been an effective tool for financing wars and breaking international financial blockades in the recent conflict between Russia and Ukraine [3].

In recent years, cryptocurrencies, such as Bitcoin, have been widely used in investment, payment, and many other fields, thus attract the attention of users and researchers. However, 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.

Received September 30, 2022; revised January 29 & March 18, 2023; accepted April 26, 2023.
Communicated by Fu-Hau Hsu.
+ Corresponding author.
* The preliminary version has been appeared at the 2022 International Conference on Networking and Network Applications (NaNA), China. This paper is supported by the National Key R&D Program of China (No. 2019YFE0103800) and STCSM (No. 21511101600).

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 for malicious transaction tracking and special transaction discovery.

Transaction traceability based on the analysis of Bitcoin network data flow is one of the most important research directions [4], in the existing research on Bitcoin transaction regulation technology. 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. We propose a method, based on gateway network traffic analysis, to 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 45% recall rate and 66.67% 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 2, we provide the transmission approach of Bitcoin transactions and the related works. We describe the details of our traceability method in Section 3. The collection of datasets and the experiment's environment were showed in Section 4, the results of our experiment were evaluated in Section 5. In Section 6, we outline our conclusions and future work.

2. BACKGROUND AND RELATED WORK

2.1 Background

Bitcoin uses an Internet-based P2P network architecture which is decentralized [5]. All nodes in the network, while acting as resource users, also assume the role of resource providers. The nodes in the Bitcoin network are of equal status. However, depending on the differences in the functions provided by the nodes, they may have different divisions of labor. 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. Miners nodes can be divided into independent miners nodes, mining pool protocol servers, and mining nodes [6].

Bitcoin transactions can be created by both core clients and lightweight wallets in the Bitcoin system that support the Simplified Payment Verification Protocol (SPV) [7]. Bitcoin networks are composed of static IP nodes and dynamic IP nodes. The static IP

nodes are the backbone nodes of the Bitcoin network. Static IP nodes are also referred to as fixed nodes. They are able to be online consistently for a long time and provide external services such as information forwarding, transactions verification, *etc*. The static IP nodes maintain complete blockchain data files which can initiate transactions. Dynamic IP nodes are temporary nodes in the Bitcoin system that have been online for a short period of time and whose IP address may change at any time, and they are mainly used to initiate transactions.

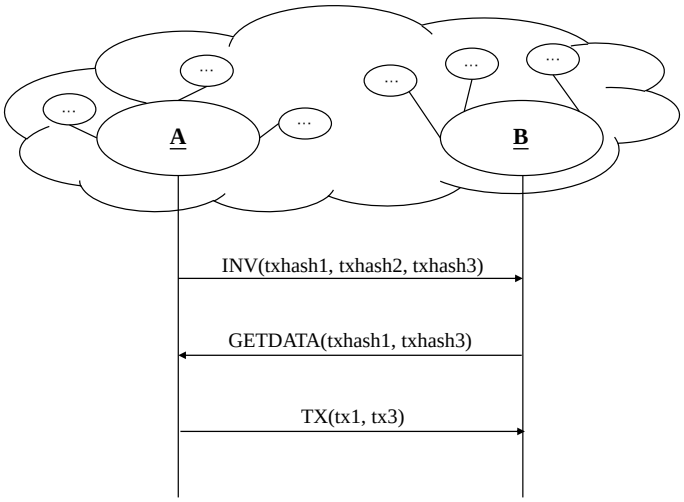


Fig. 1. Bitcoin transaction message transmission diagram.

Fig. 1 shows a brief flow of Bitcoin transaction data interaction. Peer A and B are the normal client node in the Bitcoin network. When peer A want to update the Bitcoin system information, it need to request or provide data related to transactions and blocks from other peers. In the first step, peer A sends the inventory message, which contains a list with a number of transaction hash data that peer A is going to broadcast to B. In this example, the INV message contains the hash list such as txhash1, txhash2 and txhash3.

The "INV" message (inventory message) transmits one or more inventories of objects known to the transmitting peer. It can be sent unsolicited to announce new transactions or blocks, or it can be sent in reply to a "getblocks" message or "mempool" message. The receiving peer B can compare the inventories from an "inv" message against the inventories it has already seen, and then use a follow-up message to request unseen objects.

After peer B receives the INV message, it checks its own information base and returns the GETDATA response message to peer A. The message contains the transaction hash list required by B. In the example, peer B only return the transaction hash list which contains the txhash1 and txhash3.

With the help of received the transaction hash list in the GETDATA message, peer A uses the TX message to send these transactions' content to peer B for validation. Peers that receive transactions, after validating those transactions, will also forward the transactions in the same way. This is the basic step in the dissemination of transaction data in the Bitcoin network.

Note that the transaction hash codes contained in INV messages and GETDATA messages need to be formatted before they can correspond to the transaction hash on the Bitcoin blockchain. This is because INV messages and GETDATA messages use different endian encoding method. For example, the transaction hash code contained in the GETDATA message is a4aea61c6a23ae8d13c16b7f629e53cd518674525a76cba45ec e2c66709426b7, and the corresponding transaction hash code on the blockchain is b7269 470662cce5ea4cb765a 52748651cd539e627f6bc1138dae236a1ca6aea4.

2.2 Related Work

Bitcoin static IP nodes are important for maintaining the proper and stable operation of the Bitcoin network. They serve as the backbone of the Bitcoin system and are usually required to provide some external services, such as helping other client nodes connect to the Bitcoin network, forwarding transaction information and validating 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.

[8,9] 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 [10–13]. In Bitcoin, Abu *et al*. leveraged the Bitcoin traffic to determine the nodes' states. Guo *et al*. [14]proposed an efficient Bitcoin client tracing mechanism to trace from Bitcoin server to the client through traffic analysis. Huang *et al*. [15] proposed a malicious node detection method based on behavior pattern clustering, which can quickly locate and eliminate malicious nodes. Imtiaz *et al*. [16] provided experimental evidence that the vast majority (97%) of Bitcoin nodes exhibit only intermittent network connectivity.Gervais *et al*. [17] 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.

At present, the research about the transaction traceability based on Bitcoin network traffic analysis includes two folders: traceability technology based on special propagation mode and traceability technology based on transaction propagation path [18].

Transaction broadcast mode: Some researchers have inferred the initiating node of a transaction by analyzing the broadcast pattern of transaction information in the network layer and using the special broadcast patterns generated in certain special cases. For example, Koshy *et al*. [19] 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 ori-

ginating node can be inferred through this special forwarding mode. However, the percentage of transactions with special broadcast patterns is small. In the experimental results of this paper, the percentage of transaction data with special broadcast patterns is less than 9% and the practicality of this traceability technique is poor.

Transaction broadcast path: It is a very effective way to analyze the propagation path of transaction data by collecting information about blockchain transactions transmitted at the network layer to trace the IP address of the server that created the transaction. Kaminsky [20] 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. the FirstReach [21] scheme proposed by Kaminsky: a transaction is considered to be initiated by a node in the Bitcoin network if and only if it reaches the probe first.This method only relies on the first node as a judgment feature, while the precision is low. Biryukov *et al*. [22] proposed a transaction tracing mechanism with the help of neighbor node information. The neighbour [23] scheme proposed by Biryukov *etal*.. A node is considered to have initiated a transaction when it has more than 2 neighbours in the first 8 nodes, the disadvantage of this scheme is that it needs to constantly send messages to all nodes, which may cause network congestion.The mechanism uses the transaction information broadcast by neighboring nodes as the basis for judgment in order to improve the traceability accuracy.However, this scheme requires nodes to continuously send a large amount of transaction information to all nodes in the Bitcoin network, which tends to cause serious interference to the Bitcoin network and thus is rejected by the Bitcoin network, resulting in its low practicality.

3. A TRACEABILITY METHOD FOR BITCOIN TRANSACTIONS BASED ON GATEWAY NETWORK TRAFFIC ANALYSIS

3.1 The Architecture of Traceability System

The Bitcoin system works on a P2P network that operates on the Internet. The current Internet architecture is star-shaped and consists of many local area networks as subnetworks. These sub-networks are connected through gateway devices such as routers and switches. 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 the network traffic data and record the valued results to a Bitcoin log file which will be stored in the log server as shown in the figure. By carefully laying out the deployment of monitoring equipment, we can collect Bitcoin transaction data in the target area effectively as shown in Fig. 2.

Many large enterprises and organizations operate their own Intranets, and at the same time, they connect their Intranets to the Internet as subnetworks through gateway devices at key nodes. As a result, their gateway devices often become a key pathway for sub-

networks to access the Internet. By mirroring the traffic of these gateway devices, it is possible to get a stream of Bitcoin network transaction data.

The high throughput of Bitcoin transactions really puts a heavy burden on the traceability system. In order to track the Bitcoin transactions effectively, it is necessary to design a traceability system on the physical link of the Bitcoin network based on network log storage and querying: 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 the key information but not all network traffic data to reduce the requirement of hard drive.

For the convenience of expression, this paper refers to the network covered by the traceability system as 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. In the range of network jurisdiction, we can draw the propagation map of transaction message and record the time as soon as the traffic parse server get it.

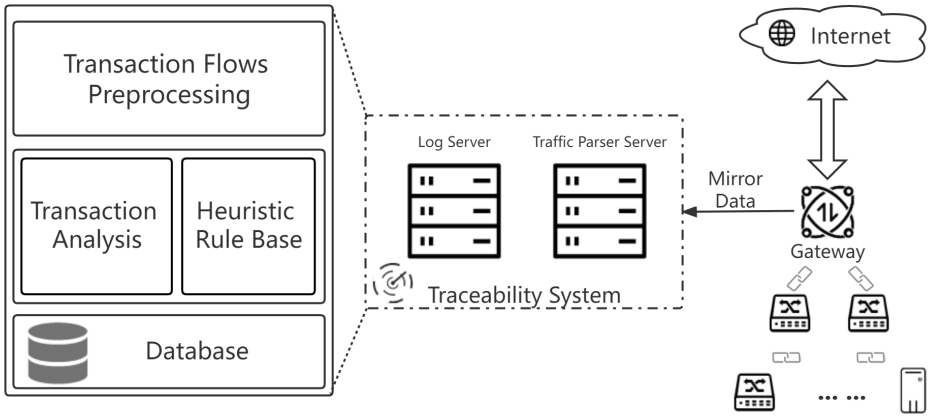


Fig. 2. The architecture of traceability system.

3.2 The Deployment of Traceability System

The deployment of traceability system should be carefully designed and it will affect the efficiency of the system. In the hierarchical network, the mirror server should be placed at the port of jurisdiction area. It will help our system to cover and monitor all the targeted network traffic flows with a set of traceability system equipment. For the Bitcoin network, all peers need to communicate with each other. Therefore, the network traffic flowing through the gateway port in the jurisdiction will contain the Bitcoin transaction information that we need to collect.

As shown in Fig. 3, a node in the jurisdiction initiates a bitcoin transaction, that node broadcasts the transaction information to other peer nodes, which outside the jurisdiction, on the bitcoin network. The traceability system placed on the gateway then detects the transaction data and thus traces the information back to the node. However, the situation will be completely changed when a full node appears in the jurisdiction at the same time.

Full nodes such as miners have the static IP and will forward transactions to the other Bitcoin nodes as a peer in P2P network. If any client initiates a transaction, the transaction data may be sent to the full node in the jurisdiction at first and broadcast to the whole Bitcoin network by full node as shown in Fig. 3. With the help of P2P network, the monitoring server which be placed at the port of our jurisdiction area can only obtain transaction information from the full node and record the IP address of full node. The information of transaction initiation node will be hidden.

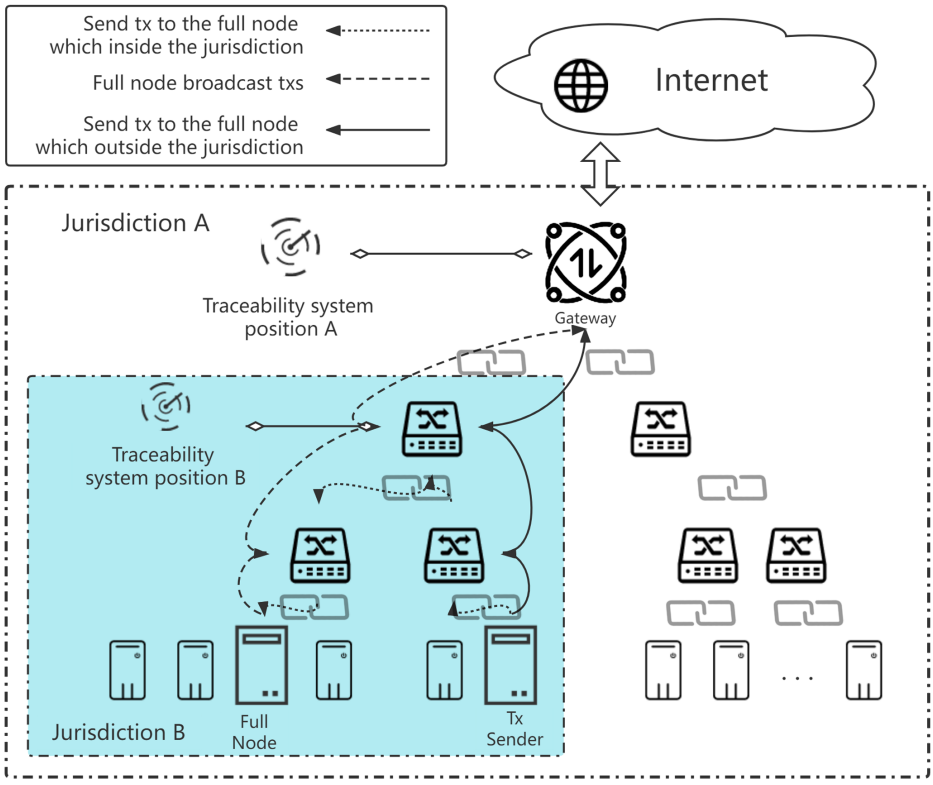


Fig. 3. The coverage of traceability system.

There are two ways to overcome this difficulty. The one way is to force the jurisdiction area to be free of full nodes by turning them off. This way is simple and efficient. The downside is that in some cases the full node cannot be removed from the system directly.

The second approach is to place a traceability system at the port of network zone which have the full node. Reducing the scope of monitoring can help us locate the initiator of the transaction and get the better accuracy of system. The disadvantage of this approach is that it requires a lot of investment and traceability system. At the same time, there will be multiple traceability system in the jurisdiction. The data between these traceability systems needs to be synchronized.

3.3 The Propagation of Bitcoin Transaction in the System

In the Bitcoin network, the protocol requires each node to have no more than 125 neighbor nodes. All nodes maintain the Bitcoin system operation by exchanging information with their neighbors. A $2_{nd} - level$ node refers to the neighbor node of the neighbor node. Through analogy, starting from the originating node, the entire Bitcoin network can be divided into neighbor nodes, $2_{nd} - level$ neighbor nodes, ..., $n_{th} - level$ neighbor nodes.

When a Bitcoin node initiates a transaction, it will immediately broadcast the transaction to its neighbor nodes. After the neighboring nodes validate the transactions, the validated transactions are broadcast to the next level of neighboring nodes according to the Bitcoin system's Trickling or Diffusion forwarding policy [24]. The transaction is then packed into a blockchain file by the miner node. The transaction will continue to be broadcasted until it reaches every node in the Bitcoin network. Even though the Bitcoin system uses a delayed forwarding strategy, all Bitcoin transactions must be initiated earlier than the time the transaction is written to the blockchain file, which is the transaction confirmation time.

Before a transaction can be written to a block file, it needs to be continuously broadcast on the Bitcoin network for a period of time. We name the transactions, which are in such period of broadcast time, as unconfirmed transactions. The traceability system can record the unconfirmed transactions, and the log record 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.For the P2P network architecture, we cannot deploy the monitor equipment for the whole the network or obtain all of the Bitcoin transactions to get their initiation time. However, with the help of the traceability system, we can log the propagation path and the timestamp of transaction which is initiated within the jurisdiction network.

Without considering interference factors such as network latency and packet leakage, we can analyze the broadcast time and broadcast path of transactions in jurisdiction of the traceability system based on these logs, to infer the origin node and its neighbor nodes. The transaction hash, as well as the relevant Bitcoin input address and the IP address, port and the time of the originating node are composed into the structured data as shown in Table 1.

Table 1. Bitcoin transaction network traffic log.

logtime	**IP$_{src}$: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

3.4 Trace the Bitcoin Transactions Which Are Originated Within the Jurisdiction Network

There are about 12,512 nodes with fixed IP addresses in the Bitcoin network, generating about 388,000 transaction records per day (data in Sept 2022 https://bitnodes.io/). The analysis of the daily flow of tens of thousands of Bitcoin transactions demands a substantial amount of computing and storage resources. To address this issue, this paper proposes three methods aimed at reducing the consumption of computational resources.

1. Record the transactions sent out from the jurisdiction (Fig. 2).
2. Analyze the transactions that are earlier than the confirmed time on the chain.
3. This paper primarily focuses on the transaction hashes conveyed by the Getdata message, which constitutes a response to the INV message and includes the requested transaction hash list. Compared to the INV message and the data volume of the TX message, the number of transaction hashes present in the Getdata message is relatively smaller. However, it is worth noting that the transmission direction of the Getdata message is reversed.

The following describes the operation steps in detail as shown in Fig. 4,

1. **Find out the net output transaction set of our jurisdiction.** Only the transactions which initiated from our jurisdiction area is meaningful to our system. The Bitcoin transaction set which entering the jurisdiction area is denoted by TX_{in}; and the Bitcoin transaction set leaving the jurisdiction is denoted by TX_{out}; 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 denoted by $(TX_{\mathrm{out}} \cap TX_{\mathrm{in}})'$; the transaction set outgoing from this jurisdiction area is denoted by TX_{netout}, that is shown in Eq. (1),

$$TX_{\mathrm{netout}} = (TX_{\mathrm{out}} - TX_{\mathrm{in}}) \cup (TX_{\mathrm{out}} \cap TX_{\mathrm{in}})'. \tag{1}$$

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 TX_{netout}, find the corresponding transaction record on the blockchain to extract the confirmation time of the transaction. The net outgoing transaction set TX_{netout} excludes the transaction set $\mathrm{TX}_{>=blocktime}$ that is later than the confirmation time stamp and forms the pre-confirmation time. The net outgoing transaction set TX_{early}, that is shown in Eq. (2),

$$TX_{\mathrm{early}} = TX_{\mathrm{netout}} - TX_{>=\mathrm{blocktime}}. \tag{2}$$

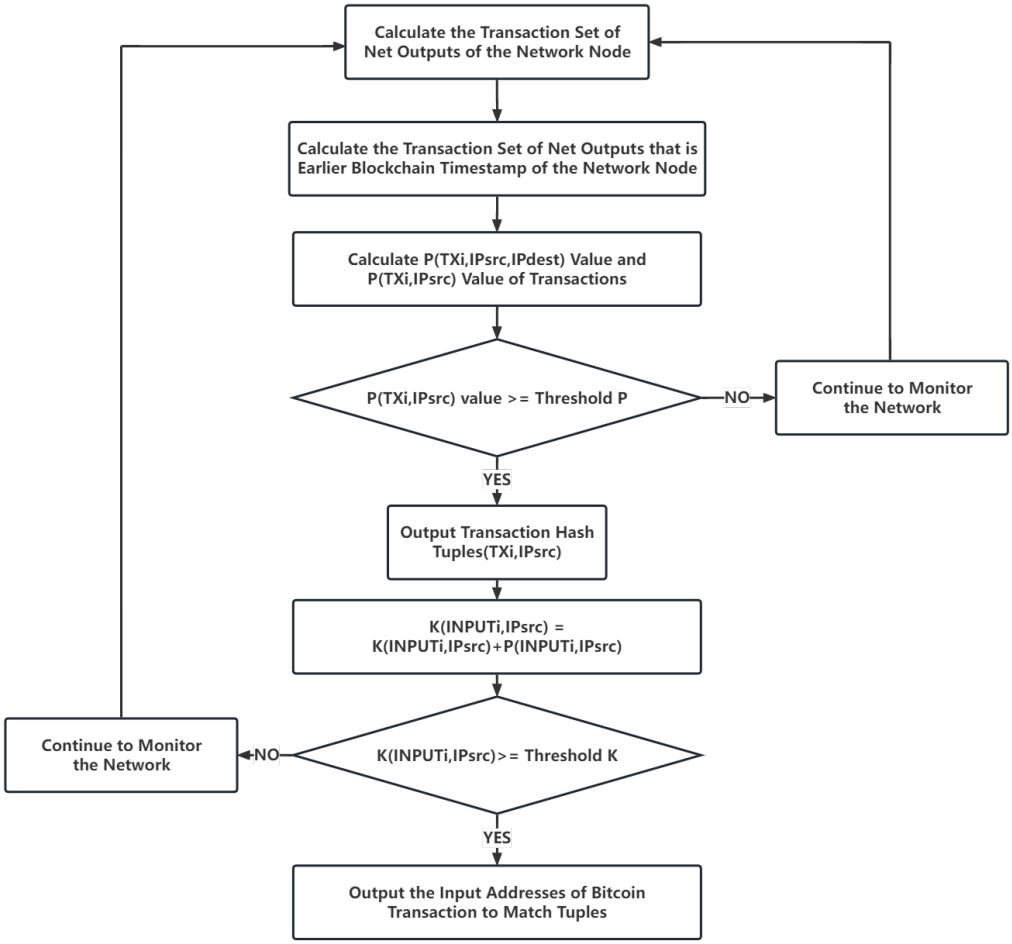


Fig. 4. Flow chart of Bitcoin transaction traceability mechanism based on network log analysis.

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 $T(tx_i)$, and the time when tx_i is sent from IP_{src} to IP_{dst} is recorded by the traceability system as $TR(tx_i,IP_{src},IP_{dst})$. $TR(tx_i)$ indicates the earliest time the transaction tx_i was recorded by the traceability system. $T(tx_i)$-$TR(tx_i,IP_{src},IP_{dst})$ is expressed as the difference between the two.

As shown in Eq. (3), the value of P(tx_i,IP_{src},IP_{dst}) is less than or equal to 1, and the earliest recorded transaction has P(tx_i, IP_{src}, IP_{dst})) = 1.

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

The transaction tx_i will be sent to different IP_{dst} addresses from the same IP_{src}, so P(tx_i, IP_{src}) represents the synthesis of all the propagated P values sent by tx_i from IP_{src}, as shown in Eq. (4).

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

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_{src}). When P(tx_i, IP_{src}) 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 K($input_i$, IP_{src}) 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 K($input_i$, IP_{src}) is 0, and the corresponding P($input_i$, IP_{src}) of all inputs in tx_i are superimposed to K($input_i$, IP_{src}) as Eq. (5), once K($input_i$, IP_{src}) 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_{src}).

$$K(input_i, IP_{src}) = K(input_i, IP_{src}) + P(input_i, IP_{src}) \tag{5}$$

4. DATA COLLECTION AND EXPRIMENT

4.1 Acquisition Time Settings

To optimize 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. 5.

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. 5 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.

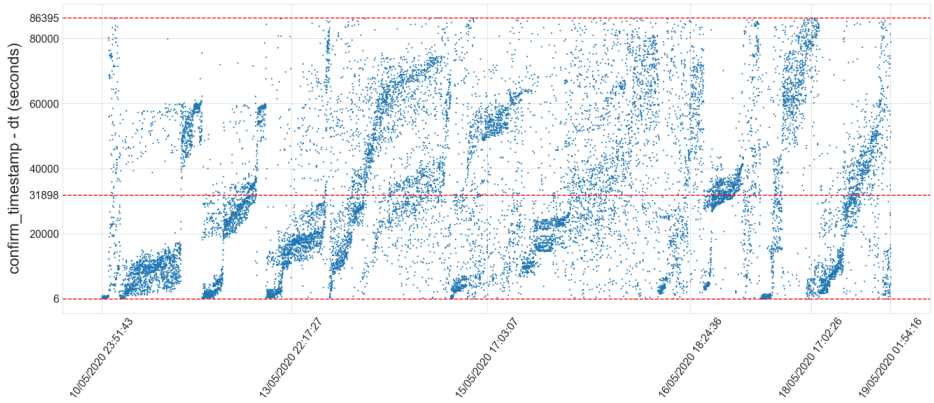


Fig. 5. The delay between transaction occurrence and confirmation.

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 Fig. 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.

4.2 Experimental Data

The threshold P-Value is used to determine whether a transaction is initiated by a node. When the $P(tx_i, IP_{src})$ 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 2.

Table 2. 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 2 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 Eq. (3). Table 3 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 3, 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 issued by the node, which was recorded earlier than the blockchain timestamp.The 10th and 15th transactions were the transactions forwarded by the neighbor nodes of the known node. They reached the traceability system firstly. The remaining 13 transactions were the node that initiates the transactions, and they reach the traceability system firstly. The log time of the traceability system was 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 Eq. (4), the two-tuple(tx_i, IP_{src}) of Bitcoin transactions with P(tx_i,IP_{src}) greater than 1 is set. Based on the data of the two-tuple (tx_i, IP_{src}), 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.Considering the reuse of the same input address in different transactions, the K values of the same input address and IP_{src} in the hash of transactions larger than the threshold value are to be superimposed, and in the experimental results we select 32 records with sumK values larger than 1 for analysis.the data (Table 4) 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_{src}), the 'IP_{src}' 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.

5. PERFORMANCE EVALUATION

5.1 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.

The traceability precision and recall described in Table 5 vary with the change of the threshold value. When the threshold value is greater than or equal to 2.5, 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 2, the precision rate is 75%, but the recall rate is 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 75%, 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

Table 3. Bitcoin transaction transfer log and P-value.

#	txhash	IP_{src}: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.*.*.161:1227	151.*.*.235:8333	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	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.*.*.161:4705	195.*.*.147:8333	1
	753e...1bbd	115.*.*.161:5028	73.219.130.254:8333	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.*.*.130:1971	50.*.*.27:8333	0.968944
	e4fd...d414	115.*.*.7:8333	47.*.*.169:13905	1

Table 4. Bitcoin input address transfer log and K-value.

input	**IP**$_{src}$	**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

captured by the traceability system, and its log time is earlier than its confirmed time in the blockchain. The precision rate is 66.67%, the recall rate is 45%, and the F value obtains the highest value of 53.73%.

Table 5. Precision and recall of transaction hash traceability with P-value.

P	samples	outputs	correct	precision	recall	F
1	40	27	18	**66.67%**	**45%**	**53.73%**
1.5	40	8	6	**75%**	**15%**	**25%**
2	40	4	3	**75%**	**7.5%**	**13.64%**
2.5	40	3	2	**100%**	**7.5%**	**13.95%**
3	40	2	2	**100%**	**5%**	**9.52%**

According to different traceability requirements, we need to 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.

5.2 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_{src}), the input address information in the transaction is analyzed experimentally, and the effect of different threshold K-Values on the precision and the recall of input address traceability is investigated. The optimal threshold value for the system is found experimentally.

Table 6 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 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 66.67%, but the recall is still 12.5%; when the threshold is 1.5, its precision is 75%, the recall is 18.75%.Under the condition that the threshold P-Value is 1, for input address traceability, the threshold K-Value is 1 as the optimal solution. When the threshold value is 1, it has an accuracy of 66.67%, a recall of 45% and the highest value of 53.73% obtained for the F-value.

Among the existing studies on the traceability of Bitcoin transactions through Bitcoin network data analysis. The neighbour [23]scheme proposed by Biryukov et al: A node is considered to have initiated a transaction when it has more than 2 neighbours in the first 8 nodes, the disadvantage of this scheme is that it needs to constantly send messages to all nodes, which may cause network congestion.the FirstReach [21] scheme proposed by Kaminsky: a transaction is considered to be initiated by a node in the Bitcoin network if and only if it reaches the probe first.

Table 6. 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%**

Table 7. Comparison of different network traceability schemes tested.

Scheme	samples	outputs	correct	precision	recall
FirstReach	20	264	4	1.5%	20%
Neighbour	20	74	6	8.1%	30%
This scheme	20	14	10	71%	50%

Table 7 describes the results of the above three Bitcoin transaction tracking techniques compared to this paper's scheme. The experiment tests all three scenarios simultaneously. The experiments use 20 transactions initiated by nodes within the jurisdiction, as well as probing nodes outside the jurisdiction. Each scheme analysed the acquired Bitcoin network data according to its own transaction tracing method and output the identified transaction initiation information. The FirstReach scheme had 264 outputs and 4 correct results.The Neighbour scheme had 74 outputs and 6 correct results.Through in-depth analysis of network logs, the tracking system uses log data to distinguish transactions that are earlier than the blockchain timestamp and net output transactions that are earlier than the jurisdictional network timestamp.this paper's scheme can filter out most of the forwarded transaction data that belong to interference noise, which has the advantage of reducing the interference to the system caused by factors such as network latency. When the threshold P is set to 1, the precision is 71% and the recall is 50%, both higher than the existing FirstReach and Neighbor schemes.

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 precision. 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 get better usage results.it is worth mentioned that the receiving node of FirstReach is very unlikely to receive the transaction forwarded by the experimental node directly if it is a node with a short online time of ordinary network speed.

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

action 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.

By observing the experimental data results, it can be found that in the default setting, when a node with a non-fixed IP initiates a transaction, the node sends the transaction directly to a neighboring node outside its jurisdiction instead of looking for a neighboring node 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.

6. CONCLUSION AND FUTURE WORK

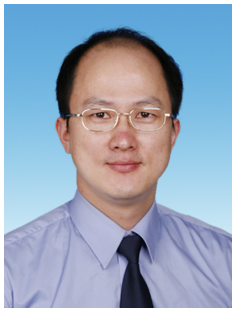
Cryptocurrency supervision techniques have attracted a lot of attention from researchers in various industries. The transaction data tracking method given in this paper uses a passive collection of transaction traffic data from the Bitcoin network, which minimizes disruption to the Bitcoin network. The method achieved high traceability precision that transactions initiated within jurisdiction. We also discuss the impact on the system when full nodes are present in the jurisdiction. It is important to strike a balance between reducing the number of network data collection nodes and obtaining highly accurate tracking results while satisfying the traceability system performance. There is still much further work to be done on our tracking method. We also need to apply our mechanism to more complex network environments to validate its effectiveness, as well as to improve its performance.

REFERENCES

1. G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, Vol. 151, 2014, pp. 1-32.
2. bitcoinnew, "8thasset," https://news.bitcoin.com/bitcoin-is-now-worlds-8th-most-valuable-asset-btc-now-targets-silvers-1-31t-market-cap/, 2021.

3. csis.org, "Cryptocurrency's role in the Russia-Ukraine crisis," https://www.csis.org/analysis/cryptocurrencys-role-russia-ukraine-crisis, 2022.
4. P. Treleaven, R. G. Brown, and D. Yang, "Blockchain technology in finance," *Computer*, Vol. 50, 2017, pp. 14-17.
5. 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, 2022, pp. 408-425.
6. A. M. Antonopoulos, *Mastering Bitcoin: Programming the Open Blockchain*, O'Reilly Media, Inc., 2017.
7. 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, 2020, pp. 1-15.
8. S. Park, S. Im, Y. Seol, and J. Paek, "Nodes in the bitcoin network: Comparative measurement study and survey," *IEEE Access*, Vol. 7, 2019, pp. 57009-57022.
9. 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, 2020, pp. 1931-1942.
10. J. Holland, P. Schmitt, N. Feamster, and P. Mittal, "New directions in automated traffic analysis," in *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 3366-3383.
11. E. Papadogiannaki and S. Ioannidis, "A survey on encrypted network traffic analysis applications, techniques, and countermeasures," *ACM Computing Surveys*, Vol. 54, 2021, pp. 1-35.
12. M. Abbasi, A. Shahraki, and A. Taherkordi, "Deep learning for network traffic monitoring and analysis (NTMA): a survey," *Computer Communications*, Vol. 170, 2021, pp. 19-41.
13. 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, 2021, p. 102940.
14. W. Guo and J. Zhang, "Towards tracing bitcoin client using network traffic analysis," in *Proceedings of IEEE International Conference on Signal, Information and Data Processing*, 2019, pp. 1-5.
15. 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, 2017, pp. 20099-20110.
16. M. A. Imtiaz, D. Starobinski, A. Trachtenberg, and N. Younis, "Churn in the bitcoin network," *IEEE Transactions on Network and Service Management*, Vol. 18, 2021, pp. 1598-1615.
17. 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.
18. Z. Liehuang, G. Feng, S. Meng, L. Yandong, Z. Baokun, M. Hongliang, and W. Zhen, "Review of blockchain privacy protection research," *Computer Research and Development*, Vol. 54, 2017, p. 17.

19. P. Koshy, D. Koshy, and P. McDaniel, "An analysis of anonymity in bitcoin using p2p network traffic," in *Proceedings of International Conference on Financial Cryptography and Data Security*, 2014, pp. 469-485.
20. D. Kaminsky, "Black ops of tcp/ip 2011," *Black Hat USA*, 2011, p. 44.
21. Dankaminsky, "Black ops of tcp/ip," https://dankaminsky.com/2011/08/05/bo2k11/, 2017.
22. I. Pustogarov, "Deanonymisation techniques for tor and bitcoin," Ph.D. dissertation, Department of Computer Science, University of Luxembourg, 2015.
23. A. Biryukov, D. Khovratovich, and I. Pustogarov, "Deanonymisation of clients in bitcoin p2p network," in *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 15-29.
24. A. Biryukov and S. Tikhomirov, "Transaction clustering using network traffic analysis for bitcoin and derived blockchains," in *Proceedings of IEEE Conference on Computer Communications Workshops*, 2019, pp. 204-209.



Dapeng Huang is a Ph.D. student at Fudan University. He is currently a member of the Laboratory of Data Analytics and Security. His research interest mainly includes blockchain data analytics and system security.

Chen Chen received the Ph.D. degree from Fudan University, Shanghai, China, in 2012. He is currently a Senior Engineer with the Software School, Fudan University. His research interests are mainly in the fields of data systems security.

Haowei Luo is a graduate student at Fudan University. He received his BS degree from Southeast University in 2021. He is currently a member of the Laboratory of Data Analytics and Security. His research interests are mainly includes blockchain data analytics and system security.

Kai Wang is a Ph.D. student at Fudan University. He received his BS degree from Fudan University in 2020. He is currently a member of the Laboratory of Data Analytics and Security. His research interest mainly includes blockchain data analytics and system security.

Weili Han is a Full Professor at Software School, Fudan University. He received his Ph.D. at Zhejiang University in 2003. Then, he joined the faculty of Software School at Fudan University. From 2008 to 2009, he visited Purdue University as a Visiting Professor funded by China Scholarship Council and Purdue University. His research interests are mainly in the fields of data systems security, access control, and password security. He is now the distinguished member of CCF and the members of the IEEE, ACM, SIGSAC. He serves in several leading conferences and journals as PC members, reviewers, and an associate editor.