SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2020, Vol. 63 132101:1–132101:15
https://doi.org/10.1007/s11432-019-9900-9

Identifying the vulnerabilities of bitcoin anonymous mechanism based on address clustering

Baokun ZHENG[1,2], Liehuang ZHU[1*], Meng SHEN[1*], Xiaojiang DU[3] & Mohsen GUIZANI[4]

[1]*School of Computer Science and Technology, Beijing Institute of Technology, Beijing* 100081*, China;*
[2]*School of Information Management for Law, China University of Political Science and Law, Beijing* 102249*, China;*
[3]*Department of Computer and Information Sciences, Temple University, Philadelphia* 19122-6008*, USA;*
[4]*Department of Computer Science and Engineering, Qatar University, Doha* 2713*, Qatar*

Received 26 April 2019/Accepted 21 May 2019/Published online 11 February 2020

Abstract The anonymity mechanism of bitcoin is favored by the society, which promotes its usage and development. An adversary should not be able to discover the relation between bitcoin addresses and bitcoin users to ensure effective privacy. However, the relation among bitcoin transactions can be used to analyze the bitcoin privacy information, which seriously jeopardizes the bitcoin anonymity. Herein, we describe the vulnerabilities associated with the anonymity mechanism of bitcoin, including the relation among bitcoin addresses and the relation among bitcoin users. Further, we demonstrate that the existing methods do not guarantee the comprehensiveness, accuracy, and efficiency of the analysis results. We propose a heuristic clustering method to judge the relation among bitcoin addresses and employ the Louvain method to discover the relation among bitcoin users. Subsequently, we construct an address-associated database of historical transactions and implement real-time updates. Extensive experiments are used to demonstrate the comprehensiveness, accuracy, and efficiency of the proposed scheme. Specifically, the proposed scheme reveals the privacy vulnerability associated with the blockchain technology. We expect that our scheme can be applied to improve the blockchain technology.

Keywords privacy, blockchain, bitcoin, transaction, cluster

Citation Zheng B K, Zhu L H, Shen M, et al. Identifying the vulnerabilities of bitcoin anonymous mechanism based on address clustering. Sci China Inf Sci, 2020, 63(3): 132101, https://doi.org/10.1007/s11432-019-9900-9

1 Introduction

Typically, bitcoin addresses [1] are generated by users, and no third party can determine the corresponding relation among addresses and the user identity information. Furthermore, each user can generate countless addresses, and users can assign different addresses to different transactions, making it considerably difficult to infer identity information based on transaction rules. Thus, the bitcoin addresses exhibit stronger anonymity when compared with that exhibited by the traditional centralized systems. However, the relation among the bitcoin transactions can be used to analyze the bitcoin privacy information. The bitcoin transactions are stored in a public global ledger; therefore, an adversary can obtain the transaction information in a relatively easy manner. By analyzing the relation among transactions, an adversary can gradually reduce the anonymity of the blockchain addresses and can even find anonymous addresses corresponding to the user's true identity information.

* Corresponding author (email: liehuangz@bit.edu.cn, shenmeng@bit.edu.cn)

Bitcoin address clustering can aggregate the addresses of a single user by analyzing the transaction data, which can help to infer the identity information corresponding to the addresses. If the identity information of a single address in the clustering set is obtained, the identity information of the remaining addresses in the same clustering set can be inferred. Further, we can use the clustering addresses to obtain comprehensive transaction data with respect to specific users, allowing us to obtain considerably detailed transaction rules. As the amount of comprehensive clustering information increases, it becomes easier for anonymous users associated with the address set to describe the identity.

Bitcoin user clustering can be used to analyze the relation among bitcoin users, which can reflect the degree of closeness among bitcoin users. The bitcoin technology constantly changes the user ownership according to the user's transaction chain, which makes it difficult to determine the whereabouts of bitcoin. We can cluster close users to discover user relations and improve the accuracy of identity recognition.

Several recent studies have investigated the relation among bitcoin addresses [2–10]. In these studies, the authors have adopted heuristic methods to analyze the relation among bitcoin addresses. However, the clustering rules in these methods are observed to be insufficient. When compared to the studies that investigate the relation between bitcoin addresses, the relation among bitcoin users has received less attention.

This study attempts to understand the traceability of bitcoin transactions and to recognize anonymous users according to the specific user behaviors in bitcoin network. Most importantly, we believe that the results of this study reveal vulnerabilities in the anonymity mechanism of bitcoin such that the blockchain technology can be improved to some extent in the future.

The primary contributions of this study can be summarized as follows.

• We propose an address clustering heuristic algorithm that can aggregate different addresses of the same user. This algorithm investigates four rules, i.e., coinbase transactions, multi-input transactions, change address, and mining pool addresses. The clustering results are more comprehensive and accurate when compared with the existing algorithms.

• We employ the Louvain community discovery algorithm to analyze the relation among users. The algorithm can partition the community based on hierarchical clustering, which can partition the transaction addresses closely related to a community and identify the bitcoin user relations. Furthermore, we have improved the original algorithm to increase the efficiency. In addition, address clustering combined with user clustering can improve the accuracy of identity prediction.

• We propose an incremental clustering algorithm based on historical clustering that can realize address clustering for large datasets. We construct an address-associated database of historical transactions and implement real-time updates. The comprehensive clustering results can be obtained by combining historical clustering with incremental clustering. If the data queried by the user is already in the historical clustering database, it can be quickly fed back to the user.

• Address and user clustering are implemented in the bitcoin public network. Further, we have verified the comprehensiveness, accuracy, and efficiency of the proposed method by conducting a large number of experiments. The results reveal that the proposed algorithms outperform the existing clustering algorithms.

• We reveal the vulnerabilities of the anonymity mechanism of bitcoin using address clustering. In addition, we indicate the manner in which our proposed method can be applied to blockchain research such that the blockchain technology can be applied well in various fields.

When compared with our previous work [11], in this study, we implemented the algorithms of the theoretical model and provided extensive experimental evaluations to prove the comprehensiveness, accuracy, and efficiency of the proposed clustering method.

The remainder of this study is organized as follows. We present the preliminaries in Section 2. Further, we provide definitions and describe the proposed system model in Section 3. We discuss the measurement methodology and implementation in Section 4. The performance analysis is presented in Section 5, and the related work and conclusions are presented in Sections 6 and 7, respectively.

2 Preliminaries

Bitcoin is a type of digital currency that relies on a global shared, distributed, and trusted ledger system. The ledger system comprises a series of complex computer-generated code. A bitcoin can be sent by anyone to any other person, regardless of the location. The bitcoin addresses are anonymous. In this section, we discuss the bitcoin addresses, bitcoin transactions, and the Louvain algorithm.

2.1 Bitcoin address

In a blockchain system, an address is a pseudonym used when a user is participating in a transaction [12]. Typically, addresses are generated using user-controlled public-key encryption algorithms such as elliptic curve cryptography. The generated public key will be used for generating the input or output address. The secret private key is used to sign a digital transaction. The blockchain address is user-generated and does not include the user identification information.

A bitcoin address comprises numbers and letters. Such addresses are essentially converted using a key pair (P_k, S_k) generated by a user using an asymmetric encryption algorithm. The public key P_k is formatted and converted to the address format Addr specified by the bitcoin protocol. The address is considered the user's account. The private key S_k, used to sign the transactions, is maintained secret by the user to ensure that only legitimate users can use a given address. The addresses in the bitcoin system are hexadecimal strings of 26–34 bits that can be processed using the hash160 algorithm and base58 encoding mechanism of the public key [13].

In addition, the address space that generates a blockchain address is considerably large, and the likelihood of duplicate addresses is low. Users can use different addresses for different transactions, enhancing the anonymity of user addresses. The private key space of the bitcoin address is 2^{256}, which is 10^{77} in decimal, and the visible universe is estimated to contain only 10^{80} atoms [13]. Thus, the bitcoin system contains sufficient address space to support a one-time address strategy.

The blockchain addresses exhibit better anonymity when compared with that exhibited by the addresses in a centralized system such as a bank card number. First, a user creates a blockchain address using the blockchain open-source program. The creation process does not require any third-party participation. The created address is a string not related with the user identity information. Therefore, no third-party organization can directly obtain the identity information corresponding to the address. Second, the blockchain address capacity is usually close to infinity; thus, a user can generate a different address for each transaction to hide the transaction rules and increase the difficulty of an attacker guessing the address identity information by analyzing the transaction data.

2.2 Bitcoin transaction

Typically, transactions represent the core business of a blockchain system, as depicted in Figure 1. A transaction involves reliable data transmission between addresses.

2.2.1 *Transaction types*

The blockchain transaction types can be categorized as coinbase, script hash, and pubkey hash transactions [14].

• Coinbase transaction. Each block corresponds to a coinbase transaction that does not contain input addresses, and the new coin is the source of all the remaining coins.

• Script hash transaction. Here, the receiving address is a synthetic address rather than the actual address. A synthetic address requires the generation of several pairs of public and private keys. In the generation process, the number of private and public key signatures that consume the coins in the address can be specified.

• Pubkey hash transaction. This transaction comprises N input addresses and M output addresses and is the most common type of transaction.

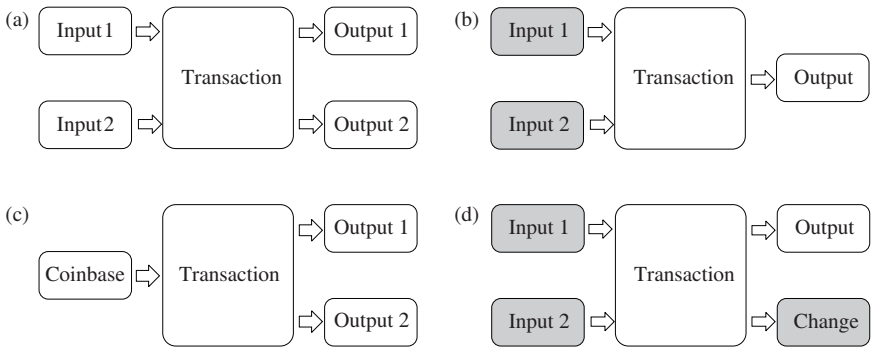

Figure 1 Bitcoin transaction. (a) Bitcoin transaction diagram; (b) multi-input transaction diagram; (c) coinbase transaction diagram; (d) change address diagram.

2.2.2 *Transactions*

Typically, a bitcoin transaction involves the transaction details, including the addresses of both the parties and the transaction amount. The sender's address can be referred to as the input address, and the recipient's address can be referred to as the output address. A transaction may contain one or more input and output addresses, and the sender's remaining coins are stored in a special address that can be referred to as the change address. The input address of a transaction is derived from the output of the previous transaction, which is used as the input of other transactions, as depicted in Figure 1(a).

However, there are some defects in the anonymity of blockchain addresses that allow the attackers to infer the relation between the addresses and the actual identity of a user [15–17]. There are three common defects.

• Address relation in a transaction. A blockchain transaction represents the data interaction between two or more users. We can infer that multiple addresses belonging to the same user or addresses of different users in a transaction exhibit an interactive relation by analyzing the transaction data. In case of the blockchain digital currency applications, multiple addresses in a transaction exhibit a close relation because of the particularity of the transaction format. For example, in a bitcoin system, each transaction must be provided with a signature for each input address involved in the transaction. Therefore, multiple input addresses created by the bitcoin wallet in a transaction must belong to the same user.

• Address relation among transactions. Although users can generate different addresses for each blockchain transaction, these addresses may be related because of the relation among transactions. For example, in a blockchain digital currency system, transaction fees must be paid while creating a transaction, indicating that the blockchain address usually requires pre-order transactions to obtain the necessary coins. The clusters comprising pre-order transactions will reveal the relation among multiple addresses.

• Relation among blockchain addresses and real identities. Although a blockchain address does not require third-party participation during creation or use, in actual practice, there are multiple touch points between the blockchain address and the user's true identity. For example, a user may reveal the identity information or a website that can leak information while accepting an off-chain service using a blockchain address.

2.3 Louvain algorithm

The Louvain [18] algorithm is a community discovery algorithm that can discover a hierarchical community structure. The advantages and disadvantages of network community partitioning are measured using the module degree of the Louvain algorithm. The advantages and disadvantages of network community division are measured by comparing the difference in connection density between the existing network and the benchmark network under similar community division. Here, the benchmark network is a random network exhibiting the same degree sequence as that exhibited by the original network. The Louvain algorithm initially makes each node a separate community. For each node i, its neighbor node j is considered. The node i is moved to the community with the largest revenue, where j is located.

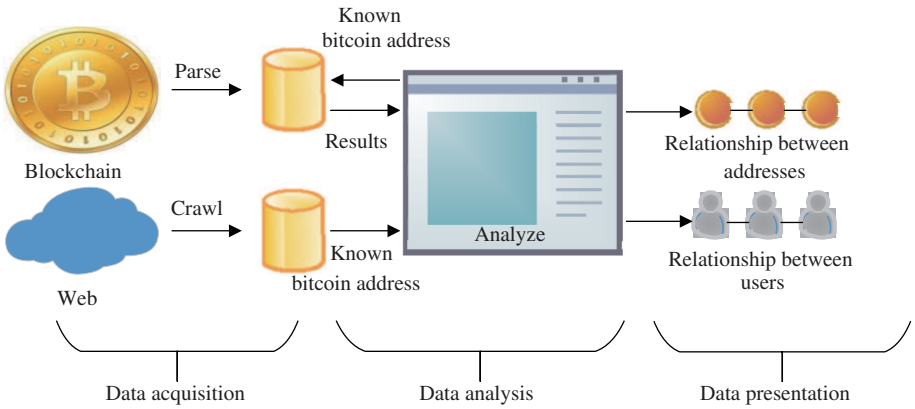

Figure 2 (Color online) System model of transaction analysis.

The process is iterated for each node, which may involve several iterations. When the modularity is the local maximum, i.e., no single node can move and improve the modularity, the first stage will stop. In the second stage, the nodes are continuously aggregated into new nodes, and the weights of the new connections are continuously recalculated.

In the Louvain algorithm, modularity (denoted as Q) and delta modularity (denoted as ΔQ) are the two main parameters. The modularity Q can describe the tightness of the internal nodes of the divided community. Here, the value range is [0, 1]. Q can be calculated as follows:

$$Q = \frac{1}{2m} \sum \left[A_{ij} - \frac{k_i k_j}{2m} \right] \sigma(c_i, c_j), \tag{1}$$

where A_{ij} represents the edge weight between nodes i and j, k_i and k_j represent the edge weights between all the nodes and nodes i and j, respectively, $\frac{k_i k_j}{2m}$ represents the average edge weight, c represents the community, c_i and c_j represent the communities in which any node i and j are located, respectively, and m represents the total number of edges in the network. σ represents the same community. When nodes i and j belong to the same community, $\sigma = 1$; otherwise, $\sigma = 0$.

While partitioning a community, the Louvain algorithm must recalculate the modularity of a community after adding a new node. After the new node joins the community, the change in community modularity ΔQ is expressed as follows:

$$\Delta Q = \left[\frac{\sum_{\text{in}} + k_{i,\text{in}}}{2m} - \left(\frac{\sum_{\text{tot}} + k_i}{2m} \right)^2 \right] - \left[\frac{\sum_{\text{in}}}{2m} - \left(\frac{\sum_{\text{tot}}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right], \tag{2}$$

where $\sum_{\text{in}}$ and $\sum_{\text{tot}}$ represent the sum of edge weights inside the community and all the edge weights connected to the community, respectively, and $k_{i,\text{in}}$ represents the sum of weights when any node i joins community c.

3 System model

3.1 Model

The transaction analysis denoted in Figure 2 denotes three processes, i.e., data acquisition, data analysis, and data representation. In the data acquisition process, we obtain all the bitcoin transaction data and the open bitcoin addresses from the public network. In the data analysis process, we analyze the bitcoin address relation and bitcoin user relation. In the data presentation process, we visualize the relation among bitcoin addresses and the relation among bitcoin users.

3.2 Definitions

Definition 1 (Address attribution). $U = \{u_1, u_2, \ldots, u_n\}$, $A = \{a_1, a_2, \ldots, a_n\}$, and $T = \{t_1, t_2, \ldots, t_n\}$ represent the user, address, and transaction sets, respectively. Input(t) and Output(t) represent the inputs and outputs of a transaction, respectively.

Definition 2 (Change address). In bitcoin transactions, there is sometimes a surplus after a user consumes. A new bitcoin address will be automatically created by the bitcoin client to maintain the balance, which can be referred to as the change mechanism of bitcoin. The user automatically spends bitcoin at the address as required.

Definition 3 (Transaction matrix). $G = (V, E)$ represents the user's transaction data matrix, used to present data. Here, V is a set of vertices transferred from the user addresses, and E is a set of edges transferred through the relation between transactions.

4 Measurement methodology and implementation

In this section, we discussed the implementation of the theoretical model. The experimental data use the addresses of the bitcoin blackmail virus CryptoLocker, which was observed in 2013.

4.1 Data acquisition

4.1.1 *Bitcoin transaction data*

Here, the transactions are confirmed transactions from the genesis block from January 3, 2009, to January 1, 2019. Note that the bitcoin transaction data (approximately 210 GB) are in the .dat format. In addition, the improved BitcoinDatabaseGenerator[1)] tool is used to parse the transaction data into an SQLserver database.

4.1.2 *Tracing data*

We use Scrapy[2)] to crawl the open bitcoin address on the web. The improved Scrapy algorithm is presented in Algorithm 1. We crawl 5 CryptoLocker blackmail addresses using Scrapy.

Algorithm 1 Scrapy web spider

Input: User URL seeds $S = \{s_1, s_2, \ldots, s_n\}$, $k \leqslant n$;
Output: Bitcoin addresses.
Initialize $S = \{s_1, s_2, \ldots, s_n\}$;
for $k \in [1, n]$ **do**
  Acquire s_k page data;
  if page data contains the addresses **then**
    Store the addresses;
  end if
  Find the associated URL from the user's URL;
  Save URL of page to S;
end for
return Addresses.

4.2 Data analysis

4.2.1 *Bitcoin address relation*

(1) Bitcoin address clustering rules. In the subsection, we propose a heuristic bitcoin address clustering algorithm containing four rules. Combining a variety of heuristic rules can effectively improve the comprehensiveness of clustering and provide better basic rules for inferring the correlation associated with transactions. The rules can be described as follows.

1) BitcoinDatabaseGenerator. https://github.com/ladimolnar/BitcoinDatabaseGenerator/releases.
2) Scrapy. https://scrapy.org.

• Multi-input transaction address clustering (MITAC). In a bitcoin transaction, the user u will select multiple bitcoin addresses from the wallet and match the value of their aggregation to achieve multi-input transactions, which avoids the transaction costs incurred by multiple transactions, if the balance in one user's bitcoin addresses is insufficient. The bitcoin in each address requires a separate signature; thus, all the input addresses in the transaction are controlled by the same user, as depicted in Figure 1(b).

The rule should be very strict; however, mixing coins also exist in multi-input and multi-output transactions. This rule deals with transactions having only a single output. In other words, this rule does not consider multi-output transactions because several multi-output transactions are so-called shared send mixing coin transactions, which are performed by multiple users to cause confusion with respect to the transaction history.

• Coinbase transaction address clustering (CTAC). A coinbase transaction creates bitcoin tokens, and each block of a bitcoin blockchain corresponds to a coinbase transaction. This type of transaction only contains output addresses. At the source of the bitcoin transaction, the newly created token, i.e., the output addresses in the transaction, is sent to the miner as a reward. Therefore, all the output addresses in the transaction are controlled by the same user set, as depicted in Figure 1(c).

• Change address clustering (CAC). The core objective of the algorithm is to identify the change address in the output address. The characteristics of this algorithm include: it as the output address is usually only once, it does not appear at the same time in the input address and the output address, the output address cannot contain only change the address, as depicted in Figure 1(d). In this rule, it is important that the transaction t contains exactly two outputs but not two inputs because it may be a mixing coin transaction if there are two inputs and two outputs.

• Mining pool address clustering (MPAC). If the number of output addresses in a transaction exceeds 100 and one of the output addresses belongs to a known mining pool, we consider the transaction as an associated mining pool and mark all the output addresses as belonging to the miner. In other words, the output addresses belong to the same set. This rule can only be verified by the mining pool; however, this rule is relatively safe given the operational structure of the bitcoin mining pools [19].

(2) Clustering algorithm for specific addresses. Generally, we analyze the different addresses of a user based on specific addresses. The algorithm for specific address is given in Algorithm 2.

Algorithm 2 Confirmation of the bitcoin address incidence relation

Input: Inquiry addresses;
Output: Addresses in the same cluster.
Initialize address set $A = \{a_1, a_2, \ldots, a_n\}$, $k \leqslant n$;
Initialize transaction set $T = \{t_1, t_2, \ldots, t_n\}$;
Acquire data of the address transaction to save to A;
for $k \in [1, n]$ **do**
  Acquire s_k page data;
  if t is coinbase transaction **then**
    Addresses outputted to save to A;
  else if one of Output(t) belongs to a known mining pool and number of Output(t) $\geqslant 100$ **then**
    Addresses outputted to save to A;
  else
    Addresses inputted to save to A;
    Judge change address to save to A;
  end if
end for
return Addresses.

(3) Historical and incremental clustering. We can build a database of addresses and clustering relations such that we do not need to repeatedly analyze the transaction data. We can directly query the results from the clustering database when we want to query the clustering information of a specified address. The algorithm for clustering the historical transaction data is provided in Algorithm 3.

As the number of bitcoin transactions continues increasing, incremental clustering performs address clustering based on the existing clustering results. Therefore, there is no need for performing addi-

Algorithm 3 Clustering of the historical transaction data

Input: Historical transaction data;
Output: Addresses in the same cluster.
Initialize transaction data set txLsit;
Initialize a mapping set of address and cluster labels addr_label;
Initialize temporary address set tempList and temporary label tempLabel = 0;
Take the transaction TX_n out of txList;
for $k \in [1, n]$ and tempLabel = 0 **do**
  if TX_n is coinbase transaction **then**
    Extract all output addresses from TX_n and add them to the collection tempList;
  end if
  for $k \in [1, n]$ **do**
    Take an undetected address from tempList;
    if the same address can be found in tempList **then**
      The label corresponding to this address is written to tempLabel;
    else
      The address is written back to tempList;
    end if
  end for
  Write all the addresses in tempList to addr_label and assign the same label to those addresses;
  addr_label++;
end for
return Addresses.

tional analysis of the transaction information processed by historical clustering, which can significantly reduce the address clustering workload, achieve real-time address clustering, and satisfy the clustering performance requirements.

The historical transaction data are divided into several shards. Here, each shard uses Algorithm 3 to process the data, and the clustering results are fused and written to the clustering database. This process creates a cluster label for all the addresses in a given historical transaction. When incremental clustering is required, we only need to extract the address set with the same clustering label from the clustering database based on the specified address. Comprehensive clustering results can be obtained by combining the historical and incremental clustering techniques. Note that the clustering process for large quantities of historical data is performed in advance; thus, only a small number of the latest transactions need to be clustered based on address. This significantly reduces the clustering time.

The blockchain data disclosure characteristics allow the cluster analysis of blockchain transaction data to have sufficient data and time and can be used to analyze the correlation among blockchain addresses from different perspectives. Algorithm 2 analyzes the address information related to a specific address in the specified data, suitable for analyzing specific tasks. Algorithm 3 analyzes the relation between all the addresses in the blockchain data and can obtain a comprehensive association relation for the addresses within a specific transaction data range, conducive for performing further operations on this data.

4.2.2 *Bitcoin user relation*

In addition to the relation among bitcoin addresses, there is also a close relation among bitcoin users. We use the Louvain algorithm to find the relation among bitcoin users. Here, the address dataset comprises addresses of the bitcoin blackmail virus CryptoLocker found by Algorithms 1–3. Based on [20], we have improved the time complexity and optimal modulus.

• Node pretreatment. In a bitcoin transaction network, a node represents a bitcoin user address, and an edge represents a transaction between the bitcoin addresses. If there is only one link between addresses i and j, the addresses i and j should divide into the same community. If they are in the same community, they should belong to the same userset, as follows:

$$Q_{i \to C(j)} = \sum (A_{ij}) - a_j^2. \tag{3}$$

The formula for calculating the increment of corresponding modularity is

$$\Delta Q_{i\to C(j)} = \frac{\sum_{i.j}}{2m^2}(2m - k_j). \tag{4}$$

Conversely, if they are in two different communities, i.e., $Q_{i\to C(j)} \leqslant 0$, $2m < k_j$. Therefore, if node i is the one link to node j, they should be placed in the same community. We classify them prior to performing community partitioning, which can reduce the modularity Q calculation of a certain number of nodes, improving the community partitioning efficiency.

The node pretreatment algorithm is presented in Algorithm 4.

Algorithm 4 $G = (V, E)$ pretreatment

Input: $G = (V, E)$ using adjacency list;
Output: The result of division of $G(V, E)$.
Initialize G;
Calculate the module value $Q_1, Q_0 = Q_1$;
$Q_2 = Q_1$;
for $k \in [1, N]$ **do**
  if $V_i == 1$ **then**
    Put vertex V_i to the community connected with it;
  else
    Take vertex V_i out of original community;
    Put V_i to community with largest ΔQ;
  end if
end for
Calculate the module value Q_1;
if $Q_2 > Q_1$ **then**
  Go to step 2;
end if
Combine the various communities into one hyperdot.

• Optimized modularity. In the coarsening inverse operation optimization process, the properties of the community nodes connected to the outside world are reconfirmed by the K-medoids algorithm [21] based on [20]. The K-medoids algorithm can improve the efficiency because the number of V_i and V_j is limited.

The algorithm is presented in Algorithm 5.

Algorithm 5 $G = (V, E)$ coarseness inverse operation optimization

Input: $G = (V, E)$;
Output: The result of division of $G(V, E)$.
Initialize G;
Check node V_i of every community connecting outside;
Check node V_j with highest degree in every community;
Adopt K-medoids algorithm;
Calculate distance value D_{ij} of every V_i and every V_j;
Calculate minimum value of D_{ij};
V_i save to community of V_j.

4.3 Data presentation

We use Gephi[3)] to visualize the relation among bitcoin addresses and users.

4.3.1 *Visualizing the address relation*

Consider a CryptoLocker blackmail address (1AEoiHY23fbBn8QiJ5y6oAjrhRY1Fb85uc) as an example (The same below, and Algorithms 2 and 4 only iterate twice; the test results are used for presentation).

3) Gephi. https://gephi.org.

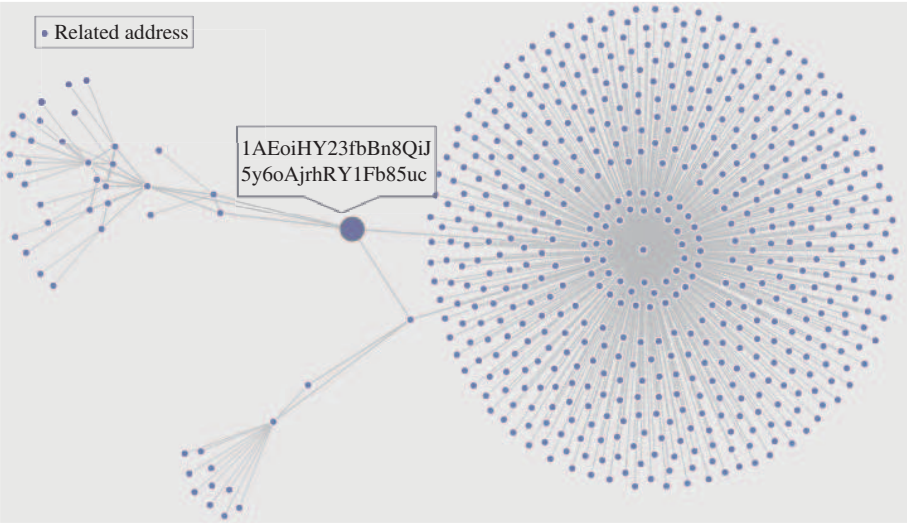

Figure 3 (Color online) Address transaction relationship diagram.

The address incidence relation diagram is acquired as depicted in Figure 3. Here, the maximum vertex is the bitcoin blackmail address that has been previously mentioned. The addresses acquired by Algorithm 2 belong to the same user.

4.3.2 *Presentation of the user relation*

We convert the CryptoLocker blackmail address set into matrices, where the addresses represent the vertices and the transactions between the addresses represent the edges. The improved Louvain algorithm identifies the addresses of different users using different colors, and 17 different subcommunities can be observed in the CryptoLocker network. We observe that their coins are gathered at the central node. We adopt Algorithms 4 and 5 to recluster subcommunities based on the shared addresses and virus active time, as depicted in Figure 4.

5 Performance analysis

In our experiment, the transaction data of the bitcoin public network are selected as the data source, and these data can be used to cluster specific bitcoin addresses and output the clustering results. First, BitcoinDatabase Generator, an open-source bitcoin data parsing tool, is used to import approximately 210 GB of the bitcoin transaction data in the .dat format into the SQLserver database. Further, we use the Java and SQL languages to develop address clustering and transaction clustering programs. The server configuration used by the clustering program is Intel® Xeon® CPU (2.60 GHZ) with 256 G RAM running Linux release 8.0.

5.1 Comprehensiveness and accuracy

The effect of address clustering is evaluated relative to the recall and accuracy rates. Here, recall refers to the ratio of the number of accurate addresses found by the clustering scheme to the total number of real addresses, as obtained using (5). Accuracy represents the ratio of the number of accurate addresses found by the clustering scheme to the total number of real clustering addresses (Eq. (6)).

$$\text{Recall rate} = \frac{\text{CN}}{\text{TN}_1}, \tag{5}$$

$$\text{Accuracy rate} = \frac{\text{CN}}{\text{TN}_2}. \tag{6}$$

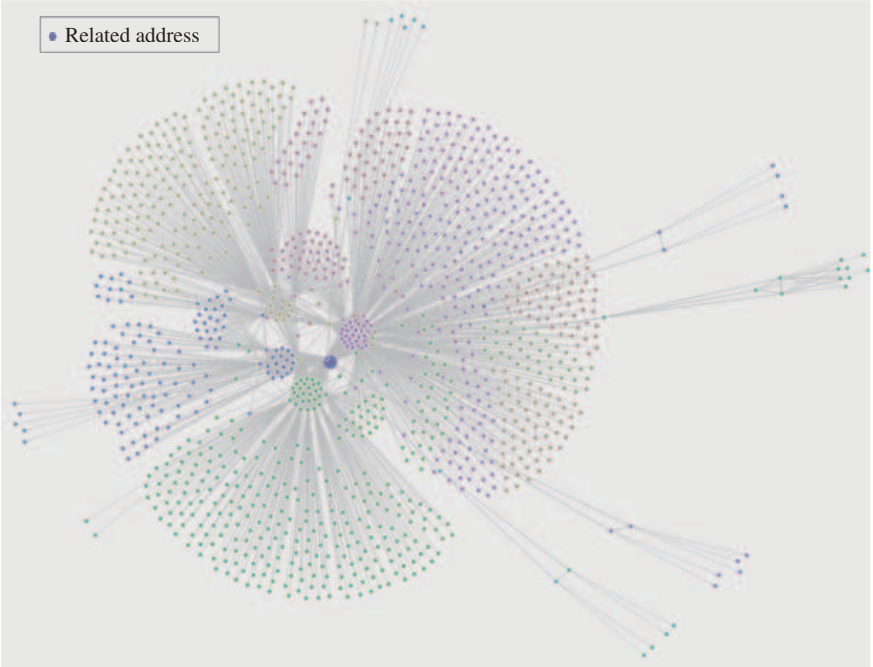

Figure 4 (Color online) The user incidence relation diagram (different colored addresses belonging to different subcommunities).

In (5) and (6), CN denotes the number of accurate addresses. TN_1 denotes the total number of real addresses, and TN_2 denotes the total number of real clustering addresses.

Our experimental data are the public bitcoin network transaction data; we use Walletexplorer[4)], which provides a clustering service for the public bitcoin network data, as a reference for obtaining the clustering results.

We extract three sets of clustering results from this website.

- Data group 1. Number of addresses in the address clustering set is less than 10.
- Data group 2. Number of addresses in the address clustering set is greater than 10 and less than 100.
- Data group 3. Number of addresses in the address clustering set is greater than 100 and less than 500.

Data are extracted from the three datasets and clustered by using the clustering algorithm. Figure 5 compares the clustering results obtained using the three datasets. The recall rates of the three datasets are 100%, indicating that the scheme can obtain comprehensive clustering information using three clustering rules. Note that an increased number of addresses results in reduced accuracy, primarily because the rules based on the change address may cause false positives. In addition, the probability of occurrence of false positives increases with increasing number of addresses.

We found 2118 CryptoLocker blackmail addresses, 795 transactions, and 1128.40 bitcoin values using Algorithms 1–3 with the virus outbreak time and bitcoin transaction volume.

To take a CryptoLocker blackmail address (1AEoiHY23fbBn8QiJ5y6oAjrhRY1Fb85uc) as an example to carry out test, and our heuristic method iterates tenth. The test results include 232 addresses, as depicted in Figure 6. The proposed heuristic method is iterated once to obtain a certain number of addresses associated with the target address. The original target address and the newly obtained address participate in the next iteration to search for the address associated with them again. Note that the number of addresses in the search increases as the number of iterations increases, which results in highly associated addresses and comprehensive results.

We use the following method to compare the effect of clustering user addresses.

4) Walletexplorer. https://www.walletexplorer.com.

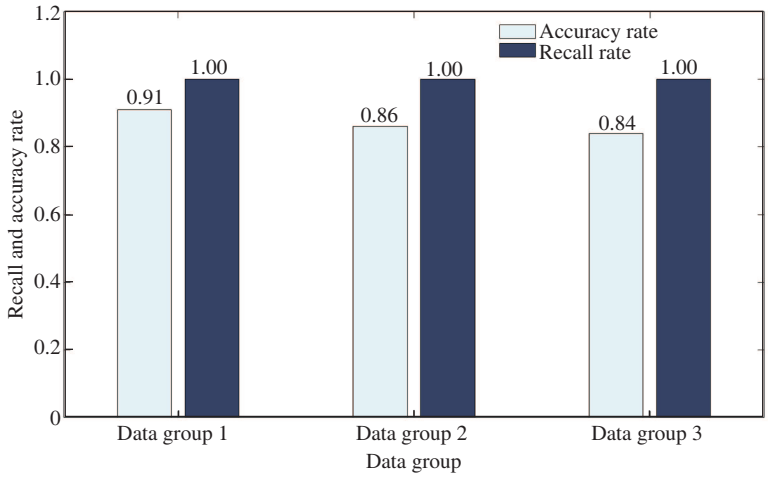

Figure 5 (Color online) Different data group results.

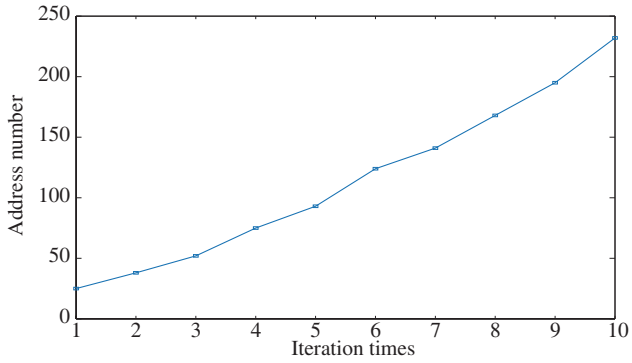

Figure 6 (Color online) Different iteration results.

- Method 1 (MITAC). The clustering rule only adopts multi-input transaction address clustering.
- Method 2 (MITAC + CTAC). The clustering rules adopt multi-input transaction address clustering and coinbase transaction address clustering.
- Method 3 (MITAC + CTAC + CAC). The clustering rules adopt multi-input transaction address clustering, coinbase transaction address clustering, and change address clustering.
- Method 4 (MITAC + CTAC + CAC + MPAC). The clustering rules adopt multi-input transaction address clustering, coinbase transaction address clustering, change address clustering, and mining pool address clustering.

Using these four methods, we obtain results for 123, 181, 224, and 232 addresses, respectively, as depicted in Figure 7. Note that each method involves different heuristic conditions and that these heuristic conditions are independent of each other; thus, with more number of heuristic conditions, more results can be obtained. The result is very close to the data provided by Walletexplorer. We conclude that more comprehensive results can be obtained as more factors are considered. During the test, as the number of iterations increased, the test results also increased.

5.2 Efficiency

In this subsection, we verify the efficiency of the improved Louvain algorithm. We uses the 2118 CryptoLocker blackmail addresses and 795 transactions that have been estimated as the dataset. We visualize these as graphs, where the vertices represent the addresses and the edges represent the transactions. In Figure 8, the average runtimes for different amounts of data are presented, and Figure 9 denotes the change in ΔQ. The results demonstrate that the runtime is reduced by approximately 4.533% and that

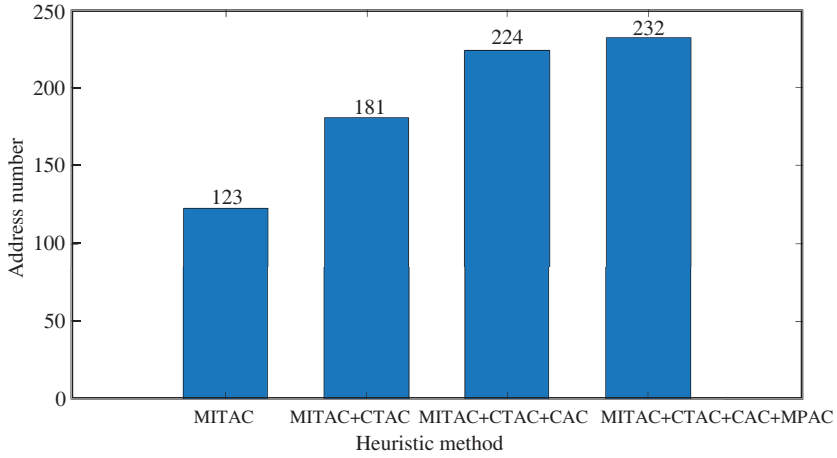

Figure 7 (Color online) Different heuristic method results.

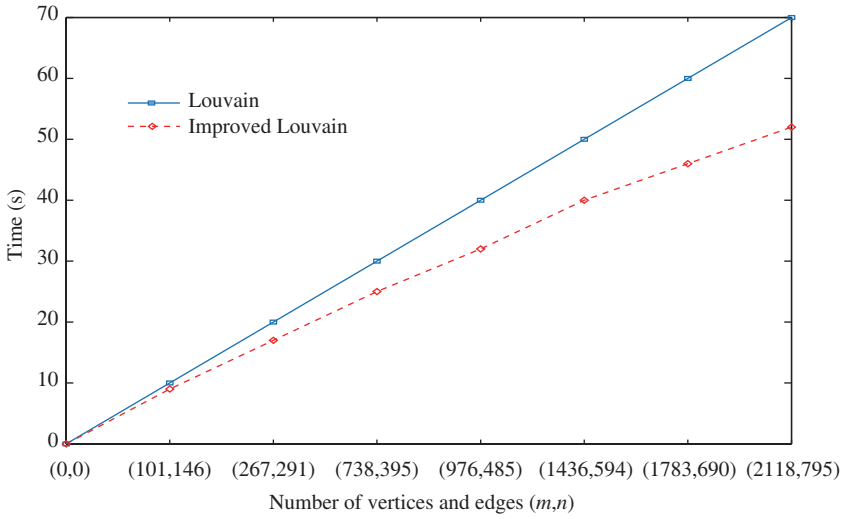

Figure 8 (Color online) Runtime comparison between the Louvain and improved Louvain algorithms.

ΔQ increases from 0.003 to 0.013 when compared with the original algorithm. Therefore, the improved Louvain algorithm exhibits obvious advantages when compared with the original Louvain algorithm.

6 Related work

Many researches have studied security and privacy in various fields [22–31], and bitcoin privacy issues have attracted significant attention. Here, a study related to the bitcoin address relation and Louvain algorithm is introduced.

6.1 Bitcoin address relation

Many studies have denoted the privacy vulnerability of the bitcoin system. For example, Reid et al. [6] set up transaction and user networks to analyze the user balance, source of coins, and capital flow of the bitcoin address published by WikiLeaks. Meiklejohn et al. [4] used a heuristic clustering method to analyze the bitcoin transaction data for identifying different addresses belonging to the same user. They analyzed the public addresses of the Silk Road that is a black market and addresses associated with some thefts, and they observed many associated addresses. Ron et al. [7] examined the statistics of bitcoin transactions. They observed that large-value transactions use several methods to spread coins across different addresses, including long-chain trading models, forked merge and loop modes, storage account

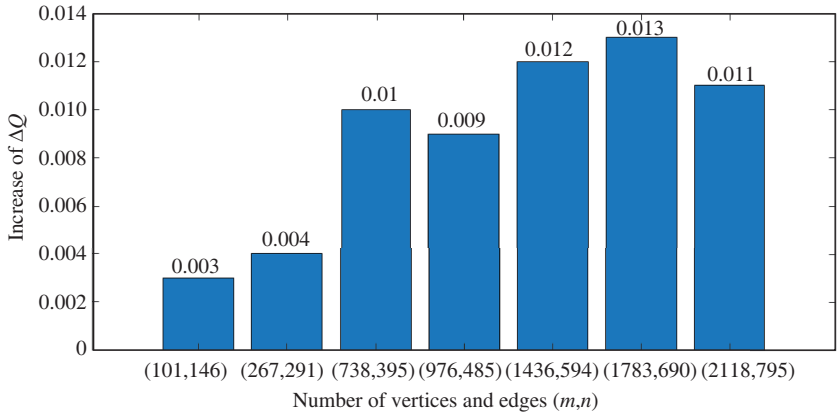

Figure 9 (Color online) Increase of $\triangle Q$ with the improved Louvain algorithm.

models, and binary tree models. Androulaki et al. [2] designed simulation experiments in schools where students used bitcoin as the daily trading currency and analysts used behavior-based clustering techniques to analyze the related bitcoin transaction data. They observed that 40% of the student identity and blockchain address can be matched even if the user adopts the bitcoin-recommended privacy protection method (i.e., the one-time address method). Zhao et al. [9] and Spagnuolo et al. [8] implemented input address and change address clustering. Monaco [5] abstracted the user transaction rules and proposed 12 parameters to identify the users. The methods could find the user's real identity corresponding to the anonymous bitcoin addresses with a recognition accuracy of 62% and error rate of less than 10.1%. In addition, Liao et al. [3] studied the process of ransomware for extortion, classified addresses received by ransom using clustering methods, and tracked the ransom addresses.

6.2 Louvain algorithm

The Louvain algorithm [18] is a community discovery algorithm that can discover a hierarchical community structure. It is primarily used for conducting social network research. We employ this algorithm to classify bitcoin addresses closely related to the community. Gach et al. [20] used a multi-level approach to maximize modularity for improving the Louvain algorithm. Meo et al. [32] optimized the modularity of the Louvain algorithm to improve the work efficiency. The optimized modularity helps to achieve the maximum network modularity by calculating the path from the central point to its connected nodes.

6.3 Difference from our previous work

Compared to our previous work [11], we have implemented the theoretical model and enriched the algorithm and have conducted experimental verification of each part of the model. In particular, we have described the algorithm for address clustering and user clustering in detail, and we have established a database that can be queried for address relation. To obtain better performance evaluation, we used different datasets to test our methods, compared the results obtained using different methods, and provided extensive experimental evaluation.

7 Conclusion

This study has proposed a heuristic method to cluster the incidence relation among bitcoin addresses. In addition, we have confirmed the incidence relation among users using the Louvain algorithm. However, the proposed heuristic method may generate certain errors while evaluating a change address with considerably large test datasets. In future, the Louvain algorithm must be further improved for ensuring more efficient implementation. The different iteration frequencies of the methods may lead to different quantities. However, a large iteration frequency results in low efficiency. Thus, studies should be conducted in the future to examine such problems.

Acknowledgements This work was supported by National Cryptography Development Fund (Grant No. MMJJ20180412).

References

1 Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. Technical Report, 2008. https://bitcoin.org/bitcoin.pdf
2 Androulaki E, Karame G O, Roeschlin M, et al. Evaluating user privacy in bitcoin. In: Proceedings of International Conference on Financial Cryptography and Data Security, Berlin, 2013. 34–51
3 Liao K, Zhao Z M, Doupe A, et al. Behind closed doors: measurement and analysis of cryptolocker ransoms in bitcoin. In: Proceedings of Symposium on Electronic Crime Research, Toronto, 2016
4 Meiklejohn S, Pomarole M, Jordan G, et al. A fistful of bitcoins: characterizing payments among men with no names. In: Proceedings of Conference on Internet Measurement Conference, New York, 2013. 127–140
5 Monaco J V. Identifying bitcoin users by transaction behavior. In: Proceedings of Biometric and Surveillance Technology for Human and Activity Identification XII, Baltimore, 2015. 945704
6 Reid F, Harrigan M. An analysis of anonymity in the bitcoin system. In: Proceedings of IEEE International Conference on Social Computing (SocialCom), Boston, 2011. 1318–1326
7 Ron D, Shamir A. Quantitative analysis of the full bitcoin transaction graph. In: Proceedings of Financial Cryptography and Data Security, Okinawa, 2013. 6–24
8 Spagnuolo M, Maggi F, Zanero S. BitIodine: extracting intelligence from the bitcoin network. In: Proceedings of International Conference on Financial Cryptography & Data Security, Christ Church, 2014. 457–468
9 Zhao C, Guan Y. A graph-based investigation of bitcoin transactions. In: Proceedings of IFIP International Conference on Digital Forensics, Orlando, 2015. 79–95
10 Zheng B K, Zhu L H, Shen M, et al. Scalable and privacy-preserving data sharing based on blockchain. J Comput Sci Technol, 2018, 33: 557–567
11 Zheng B K, Zhu L H, Shen M, et al. Malicious bitcoin transaction tracing using incidence relation clustering. In: Proceedings of International Conference on Mobile Networks & Management, Melbourne, 2017. 313–323
12 Gao F, Zhu L H, Shen M, et al. A blockchain-based privacy-preserving payment mechanism for vehicle-to-grid networks. IEEE Netw, 2018, 32: 184–192
13 Antonopoulos A M. Mastering Bitcoin: Unlocking Digital Crypto-Currencies. Sebastopol: O'Reilly Media, 2014
14 Swan M. Blockchain: Blueprint for a New Economy. Sebastopol: O'Reilly Media, 2015
15 Saxena A, Misra J, Dhar A. Increasing anonymity in bitcoin. In: Proceedings of International Conference on Financial Cryptography & Data Security, Christ Church, 2014. 122–139
16 Neilson D, Hara S, Mitchell I. Bitcoin forensics: a tutorial. In: Proceedings of International Conference on Global Security, London, 2017. 12–26
17 Li H Y, Zhu L H, Shen M, et al. Blockchain-based data preservation system for medical data. J Med Syst, 2018, 42: 141
18 Blondel V D, Guillaume J L, Lambiotte R, et al. Fast unfolding of communities in large networks. J Stat Mech, 2008, 2008: 10008
19 Lewenberg Y, Bachrach Y, Sompolinsky Y, et al. Bitcoin mining pools: a cooperative game theoretic analysis. In: Proceedings of International Conference on Autonomous Agents and Multiagent Systems, Istanbul, 2015. 919–927
20 Gach O, Hao J K. Improving the Louvain algorithm for community detection with modularity maximization. In: Proceedings of Artificial Evolution, Bordeaux, 2013. 145–156
21 Park H S, Jun C H. A simple and fast algorithm for K-medoids clustering. Expert Syst Appl, 2009, 36: 3336–3341
22 Du X J, Chen H H. Security in wireless sensor networks. IEEE Wirel Commun, 2008, 15: 60–66
23 Zhang C, Zhu L H, Xu C. PTBI: an efficient privacy-preserving biometric identification based on perturbed term in the cloud. Inf Sci, 2017, 409: 56–67
24 Du X J, Xiao Y, Guizani M, et al. An effective key management scheme for heterogeneous sensor networks. Ad Hoc Netw, 2007, 5: 24–34
25 Fahad A, Alshatri N, Tari Z, et al. A survey of clustering algorithms for big data: taxonomy and empirical analysis. IEEE Trans Emerg Top Comput, 2014, 2: 267–279
26 Zhang C, Zhu L H, Xu C, et al. PPDP: an efficient and privacy-preserving disease prediction scheme in cloud-based e-Healthcare system. Future Generation Comput Syst, 2018, 79: 16–25
27 Guan Z T, Si G L, Zhang X S, et al. Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities. IEEE Commun Mag, 2018, 56: 82–88
28 Guan Z T, Zhang Y, Zhu L H, et al. EFFECT: an efficient flexible privacy-preserving data aggregation scheme with authentication in smart grid. Sci China Inf Sci, 2019, 62: 032103
29 Shen M, Ma B L, Zhu L H, et al. Secure phrase search for intelligent processing of encrypted data in cloud-based IoT. IEEE Int Thing J, 2019, 6: 1998–2008
30 Shen M, Ma B L, Zhu L H, et al. Cloud-based approximate constrained shortest distance queries over encrypted graphs with privacy protection. IEEE Trans Inform Forensic Secur, 2018, 13: 940–953
31 Shen M, Tang X Y, Zhu L H, et al. Privacy-preserving support vector machine training over blockchain-based encrypted iot data in smart cities. IEEE Int Thing J, 2019. doi: 10.1109/JIOT.2019.2901840
32 de Meo P, Ferrara E, Fiumara G, et al. Generalized Louvain method for community detection in large networks. In: Proceedings of International Conference on Intelligent Systems Design and Applications, Córdoba, 2011. 88–93