ISBN 978-981-14-4787-7
Proceedings of 2020 the 10th International Workshop on Computer Science and Engineering (WCSE 2020)
Shanghai, 19-21 June, 2020, pp. 440-449
doi: 10.18178/wcse.2020.06.065

Clustering-Based Approach for Private Chain Data Anonymization

Quan Jiang, Bin Qu, Siyu Li, Li-e Wang+, Xianxian Li+

Guangxi Key Lab of Multi-source Information Mining & Security, College of computer science & information engineering, Guangxi Normal University, Guilin, China

Abstract. Blockchain is a new type of distributed data storage technology. With the rapid development of blockchain, vast amounts of data have been accumulated in these applications, which provide researchers with unprecedented opportunities to analyze blockchain data. However, if blockchain data is published openly, it may cause privacy leaks. Owing to the characteristics of blockchain data, the traditional anonymous method based on data publishing cannot be directly applied to blockchain data. This paper proposes an anonymous method based on clustering named clustering partition based on Bisecting k-medoids (CP-BK). We treat the transaction data of the blockchain as table data and use the k-anonymity model to protect the identity privacy of blockchain users. Finally, we evaluated the information loss and efficiency of this algorithm in experiments.

Keywords: Blockchain, Privacy Protection, Data Publish, *K*-anonymous, Clustering

1. Introduction

Blockchains have emerged as an exciting new paradigm for distributed systems; they enable many distrusting users to reach a consensus on certain transactions without requiring a trusted third party. According to actual application scenarios and requirements, blockchains can be partitioned into three categories: public chains, consortium chains, and private chains[1]. Each member on the public chain can join and leave the network freely, and there are no centralized server nodes in the network. In private chains, read and write permission is controlled by a particular company or institution. Consortium chains are blockchains managed by multiple institutions, and each institution runs one or more nodes. Essentially, a consortium chain is also a type of private chain. In this paper, the term private chains refer to both private chains and consortium chains. One cannot obtain the data in the private chain if he or she does not join it.

With the development of the blockchain and the increasing participation of users, a large amount of transaction data has been generated in various blockchains. Currently, only the transaction data in Bitcoin[2] have exceeded 200 Gb. The participation of a large number of users and active user transactions makes blockchain-based data analysis an essential and valuable research issue[3]. Recent works[4][5][6][7] analyzed blockchain's transaction networks in Bitcoin and Ethereum and found that transaction networks in blockchain consist of global financial transactions carried out by users hidden behind pseudonyms represented by public keys. By analyzing the transaction network, researchers can learn information such as user activity, transaction volume, and transaction mode.

The emergence of blockchain has brought much convenience to some applications but has also caused privacy issues. The transaction data in the blockchain are publicly visible to all participants, and any attacker can obtain all transaction information, causing privacy risks. Once a privacy leak occurs, it will cause immutable permanent loss. By analyzing transaction records related to blockchain addresses, the attacker can obtain the regular characteristics of these addresses and infer the identity information of the user based on

+ Corresponding author. Tel.: 15577308010&18677386909.
E-mail address:{ wanglie, lixx}@gxnu.edu.cn.

this[8][9]. An attacker can also obtain multiple blockchain addresses belonging to the same user by clustering the blockchain addresses[4][10]; then, they can obtain all transactions of that user.

For some blockchains with higher privacy requirements, there are currently three types of privacy protection methods: the mixing mechanism[11][12][13][14], data encryption mechanism[15], and restrict publishing technology. The mixing mechanism confuses the input and output of the transaction, making it difficult for the attacker to analyze the ledger. The data encryption mechanism encrypts sensitive data to ensure that only users with the key can read the data, thereby avoiding data leakage. These mechanisms reduce the performance of the blockchain, which is not allowed in many scenarios. Restrict publishing technology sets the blockchain as a private chain so that unauthorized nodes cannot obtain data on the chain. However, this method makes data collectors unable to obtain the data in it, which causes a massive waste of information. In this case the privacy-protected data publishing method can be used so that the external data collectors can obtain the data while protecting the privacy of the users in the private chain.

Blockchain transaction data are different from previous data types. From the perspective of data types, the blockchain transaction data can be regarded as table data, but there are some differences between them for three reasons: 1) identifiers in the blockchain cannot be removed directly; 2) the transaction record in blockchain can contain multiple input and output; 3) data do not have a clear distinction between quasi-identification and sensitive attributes. Therefore, traditional data anonymity methods cannot be directly applied to blockchain data. In addition, many clustering-based anonymous algorithms are mostly based on greedy algorithms at present, but these algorithms are usually difficult to obtain good solutions in the global range, and the results are volatile and unstable. In this paper, we propose an anonymous method based on clustering from the perspective of data publishing for blockchain transaction data. First, we use hash functions to handle identifiers; then, we treat the sum of multiple input amounts as a quasi-identifier (QI); and we do not distinguish between quasi-identification attributes and sensitive attributes, but generalize all attribute together. We use real blockchain transaction data to evaluate the degree of information loss and the efficiency of the algorithm, thereby, proving the practicability and reliability of the method.

The contributions of this paper are summarized as follows:

a) As far as we know, this is the first attempt at solving privacy issues in the blockchain from the perspective of data publishing.
b) Based on the characteristics of the blockchain data structure, we propose an anonymous scheme based on clustering.
c) We used transaction data in Bitcoin as the object of experiments and evaluated the effectiveness of the algorithm and the loss of information.

The following chapters are arranged as follows: In section 2, we introduce the background knowledge. In section 3, we introduce the model and definition of the problem. In section 4 we introduce an anonymity scheme based on clustering and analyze the time complexity of the algorithm. In section 5, we provide the experimental results.

2. Background

2.1. Transaction Data in Blockchain

In the blockchain, transactions are the basic events in the record. Blockchain address is usually used to represent the account, it is controlled by a specific entity, which can be a person or an institution. There are two mainstream accounting models in the blockchain, one being the unspent transaction output (UTXO) model represented by Bitcoin, and the other is the account/balance model represented by Ethereum. In this paper, we assume that the blockchain uses the UTXO model.

In the UTXO model, a user owns multiple blockchain addresses. A transaction records the source of income and expenditure of the transaction, which are called transaction input and transaction output, respectively. The input of each transaction comes from the output of the previous transaction until the initial mining income, which is called a coinbase transaction. During each transaction, the sender selects a subset of the address set as the transaction input and the receiver’s address as the transaction output. In a legitimate

transaction, the input of a transaction is always greater than the output of the transaction, and the difference is paid as a transaction fee to the validator of the transaction, which is the packer of the block.

Fig. 1 depicts a typical UTXO transaction network that contains 7 transactions, where TX2 contains 1 input and 2 outputs, input in0 is derived from out1 of the previous transaction TX0; the two outputs point to the addresses corresponding to in0 of tx4 and in0 of tx5, respectively.

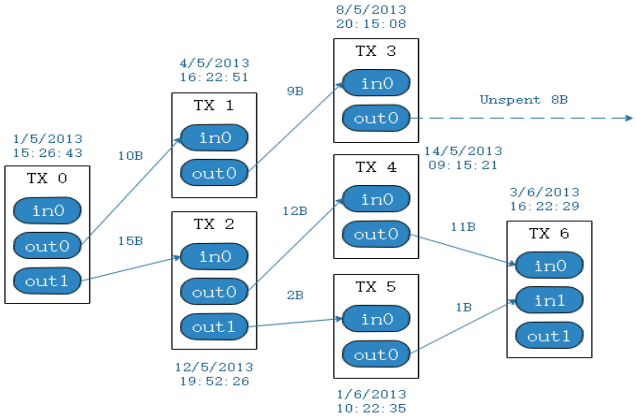


Fig. 1: UTXO model

Table 1: Transactions data in the blockchain

Txid	Input	Output	Time	goods
TX1	TX0 out0 (10B)	TX3 in0 (9B)	4/5/2013,16:22:51	apple
TX2	TX0 out1 (15B)	TX4 in0 (12B) TX5 in0 (2B)	12/5/2013,19:52:26	banana
TX3	TX1 out0 (9B)	Unspent (8B)	8/5/2013,20:15:08	oil
TX4	TX2 out0 (12B)	TX6 in0 (11B)	14/5/2013,09:15:21	salt
TX5	TX2 out1 (2B)	TX6 in1 (1B)	1/6/2013,10:22:35	milk

The transaction records in the blockchain can be viewed as a table, where each transaction record contains several transaction-related attributes. Table 1 corresponds to the transaction data of Fig. 1. There are two parts to the input and output attributes. The first part (outside the bracket) is the address corresponding to the transaction, and the second part (inside the bracket) is the amount of the transaction. The blockchain does not have a uniform data format, but in most cases, it contains the transaction id, transaction input, transaction output, and time.

2.2. Data Anonymity

Scholars have conducted extensive research on anonymous methods in data publishing[16][17][18][19]. Among them, popular anonymous technologies include *k*-anonymity[16], *l*-diversity[17], *t*-closeness[18], and slicing[19]. In these anonymous models, attributes are generally partitioned into three types: identifier, QI, and sensitive attribute. Identifiers can uniquely identify an individual identity, and QI is an attribute group consisting of several attributes. By linking QI and other information, the individual identity may be inferred with a high probability; Sensitive attributes are attributes containing private data.

We use the *k*-anonymity model to protect privacy in the blockchain. In *k*-anonymity, for each record, there are at least *k*-1 identical records in terms of QI, so the attacker cannot identify the specific individual to which the private information belongs, thereby protecting personal privacy.

3. Problem Definition

3.1. Privacy Model

The data of the private chain L is published by the private chain administrator. Let D be the original transaction data in L and D*'* be the published anonymization data of D. Only people inside the private chain

and authorized people can access and use D, which is an internal resource and is not available to the public. D' is public and can be used by data collectors for research.

For convenience of discussion, it is assumed that the data to be published contains n transaction records. Each transaction record is called a tuple, and each tuple contains d attributes. Let $T=(t_1,t_2,\ldots,t_n)$ be the collection of all tuples, and $A=(A_1,A_2,\ldots,A_d)$ be the attributes of each tuple. The attributes can be partitioned into numerical attributes and classified attributes. We use A_x to represent numerical attributes and A_y to represent classified attributes.

Attackers use background knowledge combined with published ledger data to infer user privacy information. Here background knowledge is a subset of QI. By correlating the transaction information with the records in the ledger, the attacker can know the blockchain address information of both parties in the transaction, and then he or she can query all the transaction records of these addresses in the blockchain.

In this study, we use the clustering method to implement *k*-anonymity, ensuring that the probability of each transaction being associated through background knowledge is less than 1/*k*. In this manner, the probability that the two parties of each transaction being identified does not exceed 1/*k*, thereby to achieving privacy protection.

3.2. Concepts in Cluster Anonymity

In the *k*-anonymous model, the basic concept of clustering is to divide a data set into several clusters such that each cluster contains at least *k* tuples and then generalize the attributes of all tuples in the same cluster to the same value so that the data set satisfies the *k*-anonymous model. We measure the similarity of attribute values by defining distance formulas: if the distance between two tuples is smaller, the two tuples are assumed to be closer. During clustering, according to the defined distance, all tuples in T are divided into several clusters by CP-BK.

Definition1 (Attribute generalization) Suppose there are c clusters after clustering T, then for each cluster $C_i(i=1,2,\ldots,c)$, replace QI value of all tuples in C_i with a wider range of values, this process is called attribute generalization.

Definition2 (Equivalent class) In the transaction set T to be published, the QI value of similar tuples are generalized to the same value, and these tuples are assumed to belong to the same equivalent class.

Definition3 (Equivalent tuple) In each equivalent class, all tuples have the same QI value, which are collectively called equivalence tuples.

Definition4 (Identity anonymity) In L', The probability of an attacker using background knowledge to associate the identity of a blockchain address does not exceed 1/*k*.

3.3. Data Generalization and Information Loss

The basic concept of data generalization is to replace the original attribute value with a wider range of values, multiple different attribute values have the same value after being enlarged. For a cluster C containing m tuples $t_i(i=1,2,\ldots,m)$, we generalize the QI value of all tuples in the cluster to equal values. Specifically, for the purpose of discussion, the attribute can be partitioned into two cases of numerical data and classified data.

For numerical data, suppose the value range of the numerical attribute A_x in C is $[a_c, b_c]$, where a_c is the minimum and b_c is the maximum; then, the value of $t_i(i=1,2,\ldots,m)$ on A_x is generalized to $[a_c,b_c]$. In particular, for the amount attribute, we suppose that the transaction amount is calculated based on the sum of all inputs, and each input amount is generalized separately during generalization. If transaction t contains i inputs, each input amount is $(i_1,i_2,\ldots,i_i)$; then, each input amount can be generalized as

$$(i_1^*, i_2^*, ..., i_i^*) = \left(\left[\frac{a_c \times i_1}{\sum i}, \frac{b_c \times i_i}{\sum i} \right], \left[\frac{a_c \times i_2}{\sum i}, \frac{b_c \times i_2}{\sum i} \right], \ldots, \left[\frac{a_c \times i_i}{\sum i}, \frac{b_c \times i_i}{\sum i} \right] \right) \quad (1)$$

After generalization, the sum of the inputs of each tuple in C has the same value.

For classified attributes, generalization is performed according to a predefined generalization tree, where each attribute value is generalized to the smallest type that can summarize a wider range of original attribute

values in cluster C. From the perspective of generalization tree, this value is the minimum upper bound node of all values of A_y in C. Fig. 2 depicts a generalization tree. The leaf node is the original value of the attribute, and the parent node is the generalized value of the child node. If tuple t_a has a classified attributes Apple, and t_b has a classified attribute Milk; then they are both generalized to Food Ingredient.

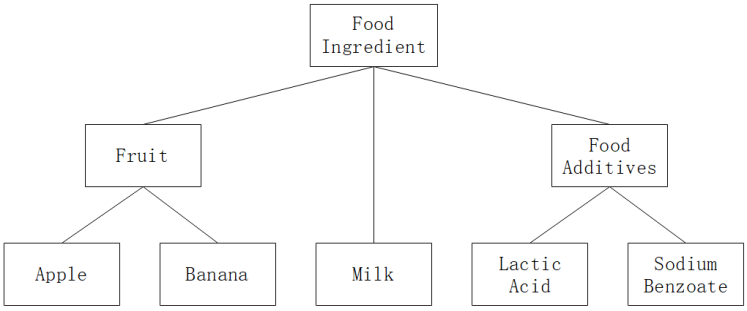


Fig. 2: Classified attributes generalization tree

Data generalization reduces the accuracy of quasi-identification attribute values, which will induce some information loss. This study uses different methods to evaluate the generalized information loss for numerical and classified attributes.

Definition5 (Information loss of generalized numerical data). Let $MaxT(A_x)$ and $MinT(A_x)$ respectively represent the maximum and minimum values of the numerical attribute A_x in T. Suppose t is generalized as $[a_c, b_c]$ on attribute A_x. Then, the information loss of t on the attribute A_x is

$$Loss(t[A_x]) = \frac{b_c - a_c}{MaxT(A_x) - MinT(A_x)} \tag{2}$$

Definition6 (Information loss of generalized classified data). Assuming that the value of the tuple *t* generalized on A_y is $t^*[A_y]$, then the information loss of t on the attribute A_y is

$$Loss(t[A_y]) = \frac{Path(t[A_y], t^*[A_y])}{Path(t[A_y])} \tag{3}$$

Here, Path ($t[A_y]$) represents the path length from leaf node $t[A_y]$ to the root node on generalization tree, and Path($t[A_y]$, $t^*[A_y]$) represents the distance from node $t[A_y]$ to node $t^*[A_y]$.

Definition7 (Information loss of generalized tuple). The information loss of tuple *t* is defined as the sum of the information loss of all attributes on A. Assuming that for all the attributes in A, the number of numeric attributes is d_1 and the number of classified attributes is d_2, we have $d=d_1+d_2$, then

$$Loss(t) = \sum_{i=1}^{d_1} Loss(t[A_{d_i}]) + \sum_{j=1}^{d_2} Loss(t[A_{d_j}]) \tag{4}$$

Definition8 (Average information loss of generalized data set). The average information loss on T is defined as

$$\text{Loss}(T) = \frac{1}{n}\sum_{i=1}^{n} Loss(t_i) \tag{5}$$

4. Anonymous Publishing Method

The purpose of anonymous publishing is to protect the identity privacy of individuals before clustering; first, we have a data pre-processing stage. In the data pre-processing stage, the identifiers are processed to prevent privacy disclosure. In the generalization stage, we divide T into several equivalent groups, each of which contains at least *k* tuples. For the privacy model, we choose the *k*-anonymity model instead of *l*-diversity because there is no apparent sensitive attribute in the blockchain data.

4.1. Preprocessing

The original blockchain data is recorded in the block, and each block contains several transactions. During the data pre-processing process, these transaction data need to be extracted from the block for recombination. In the following algorithm, the transaction is used as the basic unit for processing. In addition, there are two unique attributes in a transaction: one is the transaction ID that uniquely identifies a transaction,

and the other is the blockchain address. These two attributes are both identifiers. In previous anonymity mechanisms, identifiers are removed directly from the published data. However, in the data type of the blockchain, if the transaction id and the blockchain address are removed, then UTXO cannot link to the previous transaction and the system will fail.

To anonymize the transaction id and the blockchain address, a hash function can be used. In the hashing process, a random value salt needs to be added. Let Tid and Addr be the original transaction id and blockchain address in transactions, and Tid' and Addr' be the hash values; then,

$$\text{Tid}' = \text{Hash}(\text{Tid} + \text{salt}) \tag{6}$$

$$\text{Addr}' = \text{Hash}(\text{Addr} + \text{salt}) \tag{7}$$

After hashing, the attacker will not be able to directly correspond to the identity attributes while maintaining a one-to-one correspondence between before and after the hash, so as not to destroy the data structure of UTXO and ensure anonymity.

4.2. Definition of Distance

Definition9 (Distance between tuples). Assume there are two tuple t_p and t_q. The distance between t_p and t_q is defined as the sum of the differences in the values of each of their attributes.

In particular, the distance between t_p and t_q on the numerical attribute A_x is defined as

$$Diff(t_p[A_x], t_q[A_x]) = \frac{\left| t_p[A_x] - t_q[A_x] \right|}{MaxT[A_x] - MinT[A_x]} \tag{8}$$

For the classification attribute A_y, the distance between t_p and t_q on A_x is defined as

$$Diff(t_p[A_y], t_q[A_y]) = \frac{Path(t_p[A_y],\ t^*[A_y]) + Path(t_q[A_y],\ t^*[A_y])}{2 \times Path(A_y)} \tag{9}$$

Definition10 (Distance between classes). The distance between two Equivalence classes is defined as the distance between the Equivalent tuple of them.

Definition11 (Distance between tuple and class). The distance from tuple t to equivalent class C is defined as the distance between t and C's equivalent tuple t_c.

The distance between tuples reflects their similarity on QI, the closer the distance between two tuples, the more similar they are. In addition, by comparing the definition of attribute information loss and the definition of tuple distance, it can be seen that the distance between tuples is proportional to the generalized information loss. Therefore, when the equivalence class is constructed according to the method of minimizing the distance between tuples, we can get the minimum loss of information.

4.3. CP-BK Algorithm

Next, the tuples are clustered based on the distance between them, which are achieved by a *k*-medoids based clustering method in this study. The basic idea is to divide the data set into two clusters each time starting from the original data set T. If these two clusters satisfy *k*-anonymity, then continue to divide the clusters, otherwise, the division is stopped.

Algorithm 1 Binary k-medoids algorithm

```
input: C
Output:C1, C2
function kmedoids_parse()
1.    select_init()
2.  C1,C2 = Assign the points in C to the nearest
center
3.    while(Center point change or i<MAXITER)
4.      C1,C2 = Assign the points in C to the nearest
    center
5.      Reselect new cluster centers
6.    return C1, C2 distribution
```

Algorithm 2 CP-BK algorithm

```
input: T
Output: T*
function cluster()
1.  Q.put(T) V=∅ T*=∅
2.      while(not Q.empty())
3.          C = Q.get()
4.          C1, C2 = kmedoid_parse(C)
5.          if len(C1)<k and len(C2)<k
6.                  T*.push(C)
7.          else if len(C1(C2))>=k
8.                  Q.put(C1(C2))
9.          else
10.     V.push(C1(C2) if C1(C2)<k)
11.  for t in V
12. Assign t to the nearest cluster in the T*
```

The *k*-medoids algorithm is an improved version of *k*-means clustering. Unlike the *k*-means algorithm, each time the cluster center is updated, *k*-medoids selects a median value from the sample points as the new cluster center, whereas the *k*-means algorithm selects the average of the coordinates as the new center. Another reason why we choose *k*-medoids algorithm is that due to the existence of classification attributes, the "centroid" can not be calculated between tuples, which is a necessary step in the *k*-means algorithm.

Algorithm 1 is a binary *k*-medoids algorithm, which divides the cluster into 2 subsets each time. The algorithm first chooses 2 tuples as the initial cluster center, then continuously updates the cluster center until the cluster center does not change or reaches the maximum number of iterations (MAXITER). Finally, the algorithm returns 2 clusters.

Algorithm 2 maintains three data structures: Q is a queue and stores the clusters to be divided, T* stores the set that satisfies *k*-anonymity after partitioning, and V stores the cluster that with less than *k* tuples.

Initially, Q contains only one cluster, which includes all tuples, V and T* are empty. In each iteration, the algorithm removes a cluster C from Q and splits the cluster into two parts C_1 and C_2 by using Algorithm 1. If the number of tuples in C_1 and C_2 is less than *k*, we cannot split C anymore, and the algorithm puts C into T*. Otherwise, if the number of tuples in C_1 or C_2 is greater than *k*, it is added to Q for further division, and if it is less than *k*, it is placed in V. At the end of the algorithm, clusters in V are merged into the nearest cluster in T*. Then, T* is the final result.

4.4. Time Complexity Analysis

Algorithm 1 is the *k*-medoids algorithm. When the cluster center point is updated each time, the first step is to find the sum of the distances from each point in the cluster to the remaining points, in which the time complexity is $O(n^2)$. Then, the minimum distance is found, for which the time complexity is O(n). Thus, the time complexity of Algorithm 1 is $O(tn^2)$, where *t* is the number of iterations. Algorithm 2 continuously divides the data set using Algorithm 1, and the height of the algorithm tree is O(log*n*). Finally, the tuples in V are combined into T*, and the time complexity is O(*nc*log*c*), where *c* is the number of clusters in T*. Overall, the time complexity is $O(tn^2\log n)$, where *t* can be seen as a constant.

5. Experiment

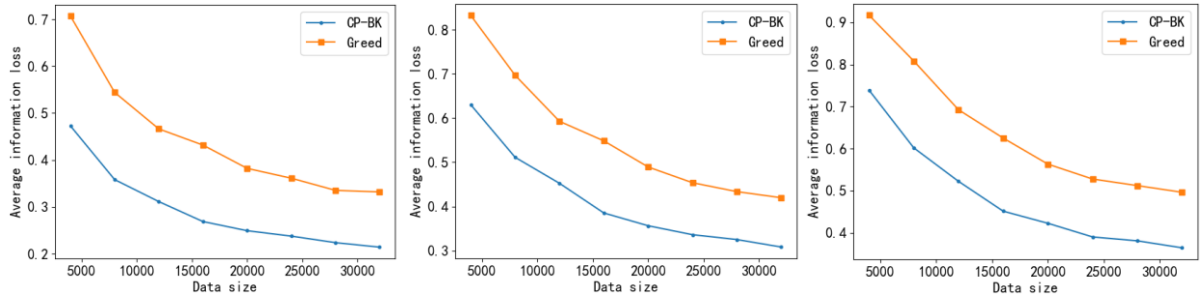
We conducted two experiments that included generalized information loss and the efficiency of the algorithm. We choose the greedy algorithm used in literature[20] as the comparison object. Due to the difference between the blockchain data and the general table data, we have made some changes to the algorithm so that the algorithm can be applied in the blockchain.

The experimental data set is from the real transaction records of Bitcoin. We selected records from 200 consecutive blocks starting on January 1, 2013, containing 32000 transaction records. Each transaction holds 7 attributes, which are *txid*, *size*, *weight*, *locktime*, *vin*, *vout*, and *time*. We downloaded the source code of bitcoin from GitHub and synchronized some of the existing bitcoin transactions. We used the method in bitcoind to extract the transaction from the block and parse it into json format. The experimental

environment was Intel(R) Core(TM)i7-4790 CPU, 8G memory, Windows 10 Professional Edition. All algorithms were implemented in Python.

5.1. Information Loss Analysis

This section mainly discusses the trend of information loss changing with the change of data size and *k* value, where *k* represents the anonymous parameter of *k*-anonymous, and information loss is defined by definition 8. Fig. 3 shows the result of the CP-BK and Greed algorithms, where the abscissa is the data scale, and Ordinates represent loss of information. The values of *k* in 3a, 3b and 3c are 5, 8 and 10 respectively. In these three figures, the change in information loss is discussed below.



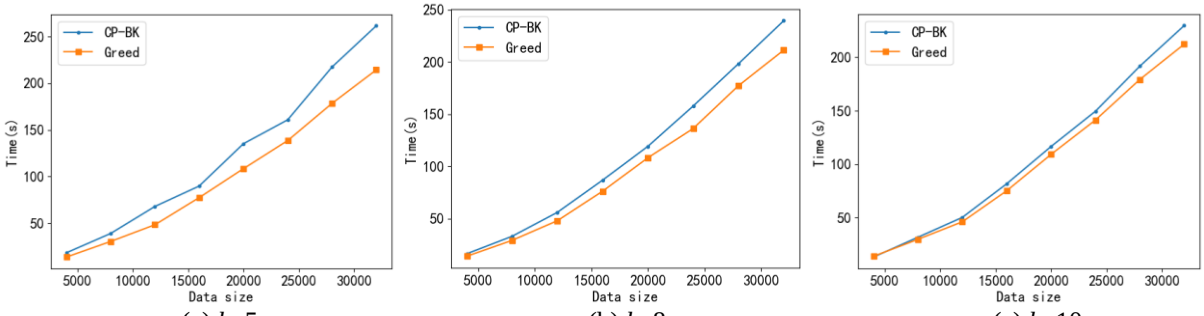
(a) *k*=5 (b) *k*=8 (c) *k*=10

Fig. 3: Information loss

It can be seen from the Fig 3 that as the data size increases, the average information loss decreases; this is because as the sample size increases, the accuracy of clustering will be higher, making the tuples in the cluster more similar. Thus, the average information loss decreases. Under the same circumstances, the information loss will increase with *k*. This is because a larger value of *k* means that there are more tuples in each equivalent class, and a larger range of generalizations is required to meet the anonymity requirement. Moreover, the larger the value of *k*, the higher the degree of anonymity. Thus, a balance between availability and anonymity according to the actual situation needs to be found.

Under the same condition of *k* value and data scale, the information loss of CP-BK is lower than that of Greedy algorithm. This shows that the CP-BK is more accurate in grouping, and the similarity within the group is higher, so the information loss is lower. In addition, the greedy algorithm does not consider the global situation when clustering, resulting in a relatively large difference in the partial grouping during clustering.

5.2. Operational efficiency analysis



(a) *k*=5 (b) *k*=8 (c) *k*=10

Fig. 4: Execution time

Fig. 4 shows that as the data size increases, the execution time of the algorithm has increased significantly. Under the same circumstances, the running time of CP-BK is slightly higher than that of Greedy algorithm in both three figures. In CP-BK, the larger the value of *k* means the shorter the running time. This correlation is because when the value of *k* is larger, the size of each clusters increases, and the number of times the partition algorithm is executed decreases; then, Algorithm 1 will be executed fewer times, so the execution time will decrease. The running time of the greedy algorithm is less affected by the change in *k* value, because as the value of *k* increases, the equivalence class will become larger, so the time to construct a single equivalence class will become longer, but at the same time because the number of total

tuples is fixed, the number of equivalence classes divided will decrease, so the total running time of greedy algorithm changes little.

6. Summary

The issue of privacy protection in data publishing is a long-discussed topic. In view of the characteristics of blockchain data, this paper proposed a privacy protection method for blockchain data publishing. We treated blockchain transactions as table data and used *k*-anonymity technology based on a clustering algorithm to protect privacy, so that the probability of each transaction record being identified does not exceed 1/*k*, thereby protecting the identity of the sender and receiver associated with the transaction record. In a specific solution, we first used a hash method to encrypt the identity attribute to prevent the identity attribute from causing privacy leakage. During clustering, we treated the QI attributes of each transaction as a multi-dimensional array. By calculating the distance between tuples, the tuples are classified into several clusters, and each cluster contains at least *k* tuples to ensure anonymity. We evaluated the information loss and time in the experimental results and proved the reliability of the method.

7. Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (Nos. 61662008, 61672176, 61502111 and 61941201), the Guangxi "Bagui Scholar" Teams for Innovation and Research Project, the Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing, the Guangxi Talent Highland Project of Big Data Intelligence and Application, the Guangxi Natural Science Foundation (Nos. 2018JJA170082) and Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security (No. 19-A-02-02).

8. References

[1] Yuan Y, Wang F-Y. Blockchain: The State of the Art and Future Trends. ACTA AUTOMATICA SINICA, 2016, 42(4): 481-494. doi: 10.16383/j.aas.2016.c160158

[2] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[R]. Manubot, 2019.

[3] Chen W, Zheng Z. Blockchain Data Analysis: A Review of Status, Trends and Challenges. Journal of Computer Research and Development, 2018, 55(9): 1853-1870.

[4] Meiklejohn S, Pomarole M, Jordan G, et al. A fistful of bitcoins: characterizing payments among men with no names[C]// Proceedings of the 2013 conference on Internet measurement conference. ACM, 2013.

[5] Alqassem I, Rahwan I, Svetinovic D. The Anti-Social System Properties: Bitcoin Network Data Analysis[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018:1-11.

[6] Chen T, Zhu Y, Li Z, et al. Understanding Ethereum via Graph Analysis[C]// IEEE INFOCOM 2018 - IEEE Conference on Computer Communications. IEEE, 2018.

[7] Maesa D D F, Marino A, Ricci L. Uncovering the bitcoin blockchain: an analysis of the full users graph[C]//2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA). IEEE, 2016: 537-546.

[8] Monaco J V. Identifying bitcoin users by transaction behavior[C]//Biometric and Surveillance Technology for Human and Activity Identification XII. International Society for Optics and Photonics, 2015, 9457: 945704.

[9] Androulaki E, Karame G O, Roeschlin M, et al. Evaluating user privacy in bitcoin[C]//International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2013: 34-51.

[10] Zhao C, Guan Y. A graph-based investigation of Bitcoin transactions[C]//IFIP International Conference on Digital Forensics. Springer, Cham, 2015: 79-95.

[11] Bonneau J, Narayanan A, Miller A, et al. Mixcoin: Anonymity for Bitcoin with accountable mixes[C]//International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2014: 486-504.

[12] Valenta L, Rowan B. Blindcoin: Blinded, accountable mixes for bitcoin[C]//International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2015: 112-126.

[13] Ruffing T, Moreno-Sanchez P, Kate A. Coinshuffle: Practical decentralized coin mixing for bitcoin[C]//European

Symposium on Research in Computer Security. Springer, Cham, 2014: 345-364.

[14] Bissias G, Ozisik A P, Levine B N, et al. Sybil-resistant mixing for bitcoin[C]//Proceedings of the 13th Workshop on Privacy in the Electronic Society. ACM, 2014: 149-158.

[15] Sasson E B, Chiesa A, Garman C, et al. Zerocash: Decentralized anonymous payments from bitcoin[C]//2014 IEEE Symposium on Security and Privacy. IEEE, 2014: 459-474.

[16] Sweeney L. k-anonymity: A model for protecting privacy[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002, 10(05): 557-570.

[17] Machanavajjhala A, Gehrke J, Kifer D, et al. l-diversity: Privacy beyond k-anonymity[C]//22nd International Conference on Data Engineering (ICDE'06). IEEE, 2006: 24-24.

[18] Li N, Li T, Venkatasubramanian S. t-closeness: Privacy beyond k-anonymity and l-diversity[C]//2007 IEEE 23rd International Conference on Data Engineering. IEEE, 2007: 106-115.

[19] Li T, Li N, Zhang J, et al. Slicing: A new approach for privacy preserving data publishing[J]. IEEE transactions on knowledge and data engineering, 2010, 24(3): 561-574.

[20] Jiang HW, Zeng GS, Ma HY. Greedy clustering-anonymity method for privacy preservation of table data publishing. Journal of Software, 2017,28(2):341−351