International Journal of Grid Distribution Computing
Vol.7, No.4 (2014), pp.21-32
http://dx.doi.org/10.14257/ijgdc.2014.7.4.03

A Novel Cloud Data Fragmentation Cluster-based Privacy Preserving Mechanism

Yali Shao, Yuliang Shi* and Hui Li

School of Computer Science and Technology, Shandong University, Jinan, China
shaoyali1411@163.com, shiyuliang@sdu.edu.cn, lih@sdu.edu.cn

Abstract

SaaS application is becoming more and more popular with the development of the cloud computing. In order to use the cloud service, the tenants should upload their data to the databases of cloud service provider, so how to protect the tenants' privacy information from snooping or leaking by DBA while keeping a good application performance is a big challenge. Therefore, we address this challenge by proposing a novel cloud data fragmentation cluster-based privacy preserving mechanism in this paper, the mechanism could give an optimal privacy preserving strategy by clustering relevancy matrix using Bond Energy Algorithm and partitioning the clustered matrix according to the privacy constraints proposed by the tenants.

Keywords: *cloud computing, SaaS, Bond Energy Algorithm, privacy preserving strategy*

1. Introduction

With the development of cloud computing, SaaS application is more and more popular. Tenants like enterprises use the service offered by SaaS provider to process their business, thus saving a lot of manpower and financial resources for software installation and maintenance. However, the tenants have to upload their data to the servers of SaaS provider. Therefore, how to guarantee the tenants' data not be leaked has become a research focus.

For the problem of privacy preserving, encryption and data confusion are the two main solutions. Encryption can effectively prevent the leakage of privacy information, but the process of decryption cost a lot of time which gave a poor user experience. Data confusion made data within the same group equivalence by the means of adding sand or generalization and so on, it can effectively hide the tenant's privacy data, but this method broke the original data information so that it cannot be reconstructed. In view of the above two shortcomings, we proposed a concept of data fragmentation in the study [1], which divided the tenants' sensitive information such as sickness or salaries and personal identity information such as names or ID numbers into different data fragmentations, then confusing their corresponding relationship. This method had a better performance than encryption and also can reconstruct the original data information.

Reconstructing need join the corresponding data fragmentations, and too much join will also affect the application performance, so this paper proposes a novel data fragmentation cluster-based privacy preserving mechanism, generating an optimal privacy preserving strategy by clustering the relevancy matrix using BEA, it can be proved that operations on the data fragmentations generated according to the privacy preserving strategy need the least number of join and have a best performance. In addition, we gave a cloud data fragmentation privacy preserving management

*Corresponding author. Email: shiyuliang@sdu.edu.cn

ISSN: 2005-4262 IJGDC

framework, so that the privacy preserving strategy could be generated use the same data fragmentation algorithm no matter what storage mode at the bottom layer.

The rest of the paper is organized as following. Section 2 is the related work about our study, Section 3 gives the whole cloud data fragmentation privacy preserving management framework, Section 4 introduces the specific realization of the cloud data fragmentation privacy preserving mechanism, Section 5 is the relevant experiments and analysis, finally, we give a briefly summarize.

2. Related Work

Protecting the tenants' data privacy which uploading to the cloud service provider effectively can promote the development of SaaS application, so how to protect data privacy from snooping or leaking is widely concerned. [2] proposed a safe business service management framework, it uses both symmetric and asymmetric encryption technology hiding tenants' privacy data. The method above need to decryption for operation, [3] proposed a method which can access the cipher text directly by a small index created before, but this method can only be used for querying.

Encryption method has great influence on the performance of data processing, the researchers put forward to prevent privacy disclosure by other means. So the method based on k- anonymous and partitioning appearing. [4] proposed a k-anonymous model, it requires each record in the group cannot be distinguished from other k-1 records. Based on this research, [5] gave a principle of l-diversity, it requires each sensitive attribute in the equivalent class have at least l different values. The methods above will cause information loss more or less, [6] put forward a method of data partition, and give an information loss calculation in detail, it proved that the method of data partition have a minimum information loss besides data leaking. However, these methods are mainly used for data publishing background, they could not reconstruct the processed data.

Our study in [7] gives a policy-based customized privacy preserving mechanism. It creates an attribute relevancy tree according to the tenant's personalized privacy protecting demand and SaaS application performance demand, and searches the tree use PFA algorithm, then generating privacy preserving strategy. According to the strategy, the tenants' data is divided into different fragmentations so that the privacy can be protected without encryption and get a better application performance. However, the mechanism using greedy thoughts, it thought the fragmentation is bigger, the performance is better. This paper is focus on the relevancy between different attributes, generates the optimal privacy preserving strategy by clustering the relevancy matrix.

3. Cloud Data Fragmentation Privacy Preserving Management Framework

The cloud data fragmentation privacy preserving management framework this paper proposed is shown in Figure 1, it is divided into three layers: the underlying layer is a data storage layer, it provides different physical storage mode based on the tenants' customization; the middle layer is a data fragmentations management layer, this layer is an important part of the privacy preserving management framework; the top layer is an application layer.

Data fragmentation management layer is composed of three modules, including data management module, strategy generating module and application processing module. After the customization of application with service provider, the tenants should upload their data to the service provider for storing, at this moment, the data management module is awakened to extract the data attributes. Then, strategy generating module gives an optimal privacy

preserving strategy according to the attributes extracted. Finally, the data fragmentation module uses the strategy to divide the tenants' data into different fragmentations for storing.

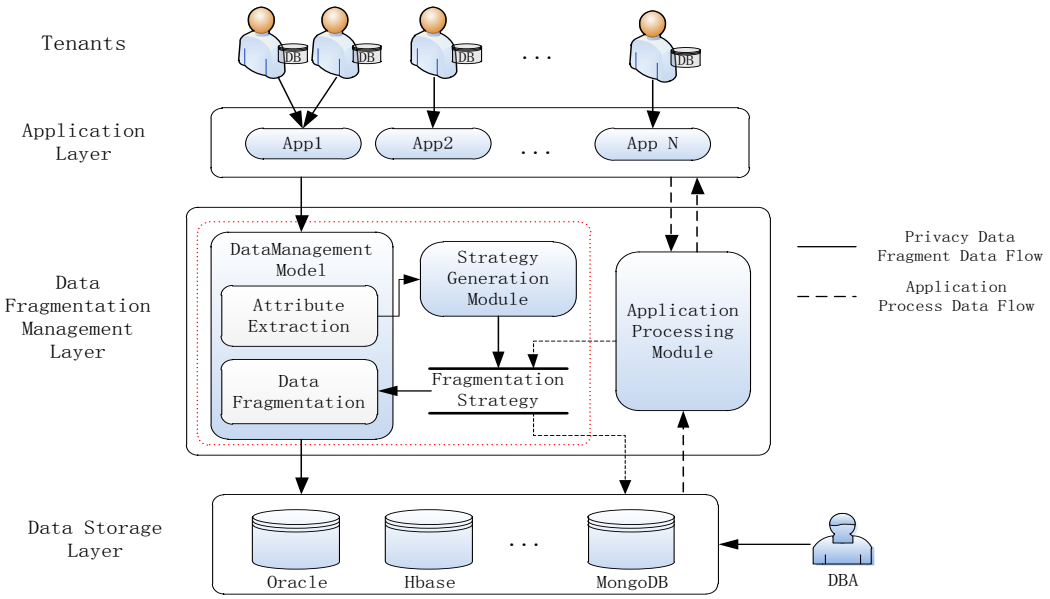


Figure 1. Cloud Data Fragmentation Privacy Preserving Management Framework

Sensitive information such as sickness or salaries and personal identity information such as names or ID numbers are stored in the different data fragmentations according to the privacy preserving strategy, DBA maintaining the underlying physical storage without knowing the strategy so that the privacy data could maintain safe.

4. Realization of Cloud Data Fragmentation Privacy Preserving Mechanism

In this section, we proposed a RDFA algorithm to generate an optimal privacy preserving strategy. For a better describe, we give some related concepts first.

4.1. Related Concept

Definition 1. Frequency of Occurrence for a given application database D = $\{T_1, T_2, \cdots, T_n\}$, T_i represents different operations. All attributes involved in D form the attribute set *Attrs*, *Attrs* = $\{a_1, a_2, \ldots, a_m\}$, where a_i is the attribute in D, also called ***Item***. A collection of items called ***ItemSet***, the ItemSet contains k items is called ***k-ItemSet***. Frequency of Occurrence is defined as the number of operations which involve the relevant ItemSet.

Table 1. Access Mode Statistic Result

TID	Attributes in Operation	Times
T_1	A B C	10
T_2	A B D	10
T_3	A B E	50
T_4	C D	50

Supposed that there is a tenant attribute set Attrs={A, B, C, D, E}, the statistic result about the application operations is shown in Table 1. Then, Frequency of 2-ItemSet {A,B} is the number of operations which involve both A and B, Frequency{A,B} = 10 + 10 + 50 = 70.

Definition 2. Attribute Relevancy the association degree between two attributes, it is defined as the number of the operations which involves the two attributes a_i, a_j, that is the frequency of occurrence of the 2-ItemSet, it can be defined as follows,

$$Relevancy(a_i, a_j) = Frequency(a_i, a_j) = \sum T(a_i, a_j) \ (a_i, a_j \in Attrs) \quad (1)$$

Among which, *Attrs* is the tenants' data attribute sets, a_i, a_j is the arbitrary two different attributes in *Attrs*, $T(a_i, a_j)$ is the operation which involves a_i, a_j in tenant's access mode, $\sum T(a_i, a_j)$ is the total number of operations which involves a_i, a_j.

Table 2. Attribute Relevancy

2-Item Sets	Attribute Relevancy	2-Item Sets	Attribute Relevancy
{A,B}	70	{B,C}	10
{A,C}	10	{B,D}	10
{A,D}	10	{B,E}	50
{A,E}	50	{C,D}	50

According to the statistic results about the tenants access mode in Table 1, calculating the attribute relevancy as shown in Table 2. Then, Relevancy (A,B) = 10 + 10 + 50 = 70.

Definition 3. Privacy Constraint the privacy preserving demand proposed by tenants, $PC\{(a_1,a_2),(a_3,a_4,a_5),\ldots\ldots\}$, also called Incompatible attribute set. The attribute in the privacy constraint (a_1,a_2), (a_3,a_4,a_5), …... cannot appear in the same data fragmentation, or tenants' privacy will be leaked.

Definition 4. Privacy Preserving Strategy divide the tenants' attribute sets into different subsets, Attrs = $\cup$ partSet(i) (i=1,2,…,k), non-overlap between any two attribute subset, partSet(i) $\cap$ partSet(j) = $\emptyset$, make sure that each attribute subsets could not break the privacy constraints, *partSet* $\not\supseteq$ *PC*.

4.2. Attribute Relevancy Cluster based Data Fragmentation Algorithm

A. Creation of Relevancy Matrix

Creating a relevancy matrix which is a N × N non negative symmetric matrix, and the elements is the attributes relevancy. N above is the number of different attributes in the tenants' data, the algorithm as follows:

Algorithm1: Relevancy Matrix Creation
Input: Attributes, Attribute Relevancy
Output: Relevancy Matrix
Procedure
1: **foreach** Transaction
2: **foreach** i∈Transaction
3: **foreach** j∈Transaction
4: RM[i,j] ++
end

The first line in algorithm 1 visits every operation, and the value of the unit, which mapped arbitrary two attributes in every operation, in the matrix will plus 1. According to the data of attributes relevancy in Table 2, creating relevancy matrix, the result shown in Figure 3a.

B. Cluster based on BEA

Bond Energy Algorithm [10], short for BEA, proposed by McCormick *et al.*, is a rearrangement clustering technique which is widely used in vertical partitioning big tables in distributed database system. BEA made the relevancy matrix as input, and keep replacing arbitrary two rows or columns so that the elements which has high relevancy could get together, and having a maximum ME. ME is short for Measure of Effectiveness, represents the similarity between each element with the up and down and left and right the four neighbors, defined as follows:

$$ME(AA) = 1/2\sum_{i=1}^{i=M}\sum_{j=1}^{j=N} a_{i,j}[a_{i,j+1} + a_{i,j-1} + a_{i+1,j} + a_{i-1,j}] \quad (2)$$

Among them, AA represents a M×N non negative matrix.

$$a_{0,j} = a_{M+1,j} = a_{i,0} = a_{i,N+1} = 0 \quad (3)$$

Equation 3 defines the boundary conditions, when one element in the matrix on the left side of the left or the right of the right matrix (upper and lower boundary similarly), the element value is 0, namely boundary elements and the elements outside the matrix have no relevancy.

According to the definition of BEA, clustering the relevancy matrix, making attributes which have higher relevancy get together so that ME have a maximum value. the objective function is defined as follows:

$$Max\{ME(AA) = 1/2\sum_{i=1}^{i=N}\sum_{j=1}^{j=N} Relevancy(a_{i,j}[a_{i,j+1} + a_{i,j-1}])\} \quad (4)$$

Due to the relevancy matrix in algorithm 1 is symmetric, the order of rows is same as columns, so the objective function could be simplified as follows:

$$Max\{ME(AA) = 1/2\sum_{i=1}^{i=N}\sum_{j=1}^{j=N} Relevancy(a_{i,j}[a_{i,j+1} + a_{i,j-1}])\} \quad (5)$$

The clustering process has two steps:

1) Reorganize the init relevancy matrix by first placing one of the columns arbitrarily, i = 1.
2) Try to place individually each of the remaining N - i columns in each of the i + 1 possible positions (to the left and right of the i columns already placed), and compute each column's contribution to the ME. Place the column in the position that gives the largest incremental ME. Increment i by 1 and repeat this step until i = N.

Algorithm2: Attribute Relevancy based Clustering

Input: Relevancy Matrix
Output: Clusterd Matrix
Procedure

```
1: ME = 0;
2: for (i = 1; i < N; i++)
3:  for (j = 0; j <= i; j++)
4:    Insert(F[i], TempMatrix, j, i);
5:    if (CompME(i) > MaxME) {
6:      MaxME = CompME(i);
7:      pos = j;
          }
end
```

The first line gives ME an initial value 0, fourth line insert the i^{th} column F[i] to the j^{th} column in middle matrix TempMatrix. CompME is computed according to the formula (5), if ME is bigger, record the position and replace the value of MaxME, otherwise nothing changed.

According to the relevancy matrix constructed in Figure 2a, clustering use algorithm 2, the process as Figure 2b to the Figure 2d.

	A	B	C	D	E
A	∞	70	10	10	50
B	70	∞	10	10	50
C	10	10	∞	50	0
D	10	10	50	∞	0
E	50	50	0	0	∞

Figure 2a

	C	D	A	B	E
A	10	10	∞	70	50
B	10	10	70	∞	50
D	50	∞	10	10	0
C	∞	50	10	10	0
E	0	0	50	50	∞

Figure 2b

	A	B	C	D	E
D	10	10	50	∞	0
C	10	10	∞	50	0
A	∞	70	10	10	50
B	70	∞	10	10	50
E	50	50	0	0	∞

Figure 2c

	C	D	A	B	E
C	∞	50	10	10	0
D	50	∞	10	10	0
A	10	10	∞	70	50
B	10	10	70	∞	50
E	0	0	50	50	∞

Figure 2d

C. Clustered Relevancy Matrix Partition

According to the privacy constraints proposed by tenants, partitioned the clustered matrix, make sure the subsets after the partition not breaking the privacy constraints and have the highest relevancy. The algorithm as follows:

Algorithm3: CM Partition
Input: Clustered Matrix
Output: Privacy Preserving Strategy
Procedure
1: **foreach** Privacy Constraints
2: if the attribute sets was divided
3: **if**(Y) **then**
4: continue
5: **else**
6: [partSet(count),partSet(++count)] = partSet(i)
7: if the attribute sets can be combined with other sets
8: **if**(Y) **then**
9: merge
10: count--
end

The first line traverse each privacy constraints. The 2-6 lines check attributes between each other is partitioned or not, if the attributes in the privacy constraints appear in different partSets, the attributes has been partitioned, and move on to the next cycle; otherwise, divided partSet(i) into two parts, make sure PM (Measure of PartSets) have a maximum value:

$$PM = \sum_{i=1}^{k}\sum_{j=1}^{k} CM_{ij} * \sum_{i=k+1}^{n}\sum_{j=k+1}^{n} CM_{ij} - (\sum_{i=1}^{k}\sum_{j=k+1}^{n} CM_{ij})^2 \quad (6)$$

Line 7 to 9 is to determine whether the data fragmentations can be combined, if yes, then merge the data fragmentations, and the number of it minus 1.

4.3. Privacy Preserving Strategy Optimal Proof

Property 1. Downward Closure If there is no true superset A' of A, so that each operation contains A also contains A', then we say the attribute set A is closed. So it's obvious that the occurrence frequency of superset is no higher than the subset.

Proof. Suppose that there is a subset $Q_{sub}\{i_1, i_2, \ldots, i_m\}$, its superset is $Q_{su}\{i_1, i_2, \ldots, i_m, \ldots, i_n\}$. Due to the attributes of subset will also appeared in the super set, then the operation involves the attributes in superset will also involves the attributes in subset, that is $Frequency\{Q_{su}\} <= Frequency\{Q_{sub}\}$.

According to the data in Table 1, {A,B} is a subset of {A,B,C}, Frequency{A,B,C} = $|T_1|$ = 10, Frequency{A,B} = $|T_1| + |T_2| + |T_3|$ = 10 + 10 + 50 = 70, visibly, Frequency{A,B,C} < Frequency{A,B}, namely the occurrence frequency of superset is less than subset.

Property 2. Completeness The occurrence frequency of the attribute set which have a high frequency is higher than the superset of the attribute set which have a lower frequency.

Proof. Suppose that there are two subsets $Q_{sub1}\{i_1, i_2\}$, $Q_{sub2}\{i_3, i_4\}$, $Frequency\{Q_{sub1}\} > Frequency\{Q_{sub2}\}$. $Q_{su2}\{i_3, i_4, \ldots, i_n\}$ *is* superset of $Q_{sub2,}$ According to the property 1, $Frequency\{Q_{sub2}\} > Frequency\{Q_{su2}\}$, so $Frequency\{Q_{sub1}\} > Frequency\{Q_{sub2}\} > Frequency\{Q_{su2}\}$.

Theorem 1. Partitioning the tenants' data according to the privacy preserving strategy given by the RDFA, then the application operation on the data fragmentations need the least join and have a best performance.

Proof. The total number of join that application operations needed is

$$\sum_{i=1}^{i=N}\sum_{j=1}^{j=N} Frequency(a_i, a_j) = 1/2 \sum_{i=1}^{i=N}\sum_{j=1}^{j=N} Frequency(a_{i,j}[a_{i,j+1} + a_{i,j-1}]) = ME(AA)$$

RDFA construct relevancy matrix based on the relevancy between attributes, namely the elements of the relevancy matrix is the occurrence frequency of 2-itemset, through the rearrangement of the order of rows or columns, ME has a maximum value, the objective function is $Max\{ME(AA) = 1/2 \sum_{i=1}^{i=N}\sum_{j=1}^{j=N} Relevancy(a_{i,j}[a_{i,j+1} + a_{i,j-1}])\}$,

that is, $Max\{ME(AA) = 1/2 \sum_{i=1}^{i=N}\sum_{j=1}^{j=N} Frequency(a_{i,j}[a_{i,j+1} + a_{i,j-1}])$. Supposed that the objective function is defined as making the occurrence frequency of k-itemset is the highest (k>2), that means solving the function

$$Max\{ME'(AA) = 1/2 \sum_{i=1}^{i=M}\sum_{j=1}^{j=N} Frequency(a_{i,j}, a_{i,j+1}, \ldots, a_{i,j+(k-1)} + a_{i,j}, a_{i,j-1}, \ldots, a_{i,j-(k-1)}])\}$$

According to the property 1 and property 2,

$$Frequency(a_{i,j}, a_{i,j+1}, \ldots, a_{i,j+(k-1)}) < Frequency(a_{i,j}, a_{i,j+1}),$$
$$Frequency(a_{i,j}, a_{i,j-1}, \ldots, a_{i,j-(k-1)}) < Frequency(a_{i,j}, a_{i,j-1}),$$

then $Max\{ME'(AA) < Max\{ME(AA)\}$.

RDFA set the relevancy between attributes as the objective function to generating the privacy preserving strategy, so the number of join needed by application operations is the least.

5. Experiments

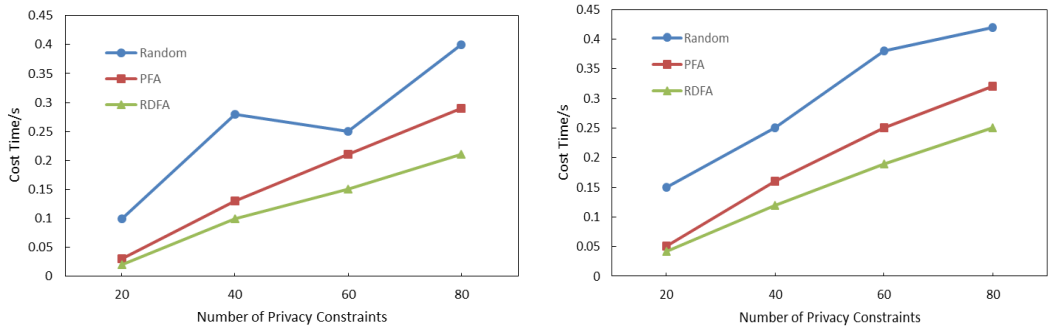
In order to verify the validity and performance of the cloud data fragmentation cluster-based privacy preserving mechanism proposed by this paper, and the generality of the cloud data fragmentation management framework, we build 3 servers with the same configuration. That is CPU Inter (R) Core(TM) i5-2400 3.10GHz, memory 8GB, hard disk 1TB, Red Hat Enterprise Linux6.2 System, test databases are Oracle Database 11g Release 2, MongoDB 2.2, Hbase-0.94.13, programming environment uses Eclipse-SDK-3.4.3-win32, and Java6 is the programming language.

We are ready to several groups of different experimental data sets, including different size of tenant data with different number of attributes, different size of privacy constraints, different types of access patterns, as shown in table 3.

Table 3. Experiment Data Sets

Size of tenant attibutes	Number of privacy constraints	Access mode
50	20	RAS=100
100	40	RAS=200
150	60	RAS=300
200	80	RAS=400

In the experiment, take the group whose attributes number is 100 and 200 as our test data, partitioning them under the privacy constrains which number is 20, 40, 60 and 80 with Random, PFA, RDFA algorithm respectively, the result is shown in Figure 3.

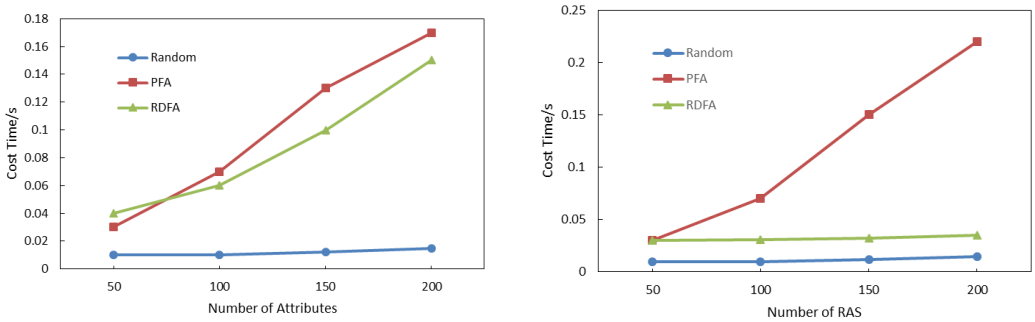


a) When Attributes is 100 **b) When Attributes is 200**

Figure 3. Cost Time of the Application on the Data Block generated by Random Algorithm, PFA, RDFA

As can be seen from the graph, when the number of privacy constraints increases, the time cost of operations also increases, this is because the number of data fragmentations increases with the complexity of privacy constraints increases, so the application operations on the data fragmentations requires more join. It is obviously that operations cost more time with the privacy preserving strategy generated by Random algorithm, because Random algorithm divided the attributes into different data fragmentations randomly without taking the access pattern into account. Also because of the

randomness, the time cost changed frequently, comparatively, the curve of PFA and RDFA are smooth and steady. RDFA precedes PFA from the overall perspective, because PFA is designed from the data fragmentations perspective, the data fragmentations involves higher relevancy attributes as bigger as better, however, RDFA is designed from the relevancy perspective, clustering the attributes according to the relevancy, avoiding the problem that dividing the attributes with higher relevancy but in the small data fragmentations into the big one.



a) With different number of attributes b) With different number of RAS

Figure 4. Cost Time of the Generation for Privacy Fragmentation Strategy with the Varying Number of Attributes or RAS

Figure 4 is the cost time of privacy preserving strategy generating with the Random, PFA and RDFA algorithm on different size of tenants' data and different access pattern. Specifically, in accordance with the experimental data given in Table 3, choose the data whose size were 50, 100, 150, 200, and choose the access pattern whose RAS were 100, 200, 300, 400, the number of privacy constraints fixed to 60. As can be seen in Figure 4a, along with the increase of number of attributes, the three curves show a increase tendency. Random algorithm has a highest efficiency because it only needs to check whether the data fragmentations is in violation of privacy constraints. PFA and RDFA algorithm increased linearly, for PFA, this is because the attribute correlation model is a tree form, attribute set is greater, exploration branches is more, for RDFA, the relevancy matrix takes the number of attributes as its dimension, attribute set is greater, the size of relevancy matrix is larger, and the convergence time of clustering is longer.

Figure 4b presents the variation tendency of the three algorithms with the different access patterns. As can be seen, Random and RDFA curve have little change, this is because the Random algorithm does not consider the tenants' data access patterns, and the RDFA algorithm only use access pattern to calculate attributes relevancy, and the variation of access pattern has little effect on the calculation cost. For PFA, the correlation attribute set is generated according to access pattern, and the number of correlation attribute set has a direct effect on the number of branches of correlation tree. When the number of correlation attribute set increases, the time cost of PFA increases linearly.

To sum up, the time cost of RDFA increases linearly with the increasing size of tenants' data attribute set within the acceptable range; at the same time, access pattern has little effect on the RDFA, that means RDFA is more stable than PFA.

6. Summarize

For the popularity of SaaS application, the problem of privacy preserving has attracted a lot of tenants' attention. How to protect data privacy from snooping or leaking and improve the application efficiency is a challenge. To address the challenge, this paper proposes a novel cloud data fragmentation cluster-based privacy preserving mechanism, it uses BEA to cluster the relevancy matrix, then partitioned the matrix according to the privacy constraints given by the users, generating an optimal privacy preserving strategy at last. Next we will focus on the privacy preserving strategy generating algorithm under the condition that the frequently alteration requirement of the tenants' privacy constraints.

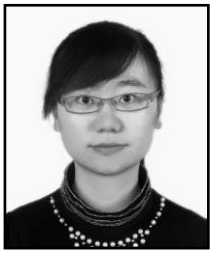
Acknowledgements

This research was supported by the National Natural Science Foundation of China (No. 61272241), Independent Innovation Foundation of Shandong University (No. 2012TS074).

References

[1] K. Zhang, Q. Li and Y. Shi, "Research on Data Combination Privacy Preservation Mechanism for SaaS", Chinese Journal of Computers, vol. 33, no. 11, **(2010),** pp. 2044-2054.
[2] Y. Karabulut and I. Nassi, "Secure Enterprise Services Consumption for SaaS Technology Platforms", IEEE International Conference on Data Engineering, **(2009)**, pp.1749-1756.
[3] C. Liu, L. Zhu and L. Li, "Fuzzy Keyword Search on Encrypted Cloud Storage Data with Small Index", 2011 IEEE Internaional Conference on Cloud Computing and Intelligence Systems, **(2011),** pp. 269-273.
[4] L. Sweeney, "k-anonymity: A model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems, **(2002)**, vol. 10, no. 5, pp. 557-570.
[5] A. Machanavajjhala, J. Gehrke, D. Kifer and M. Venkitasubramaniam, "l-diversity:Privacy beyond k-anonymity", Proceedings of the 22nd International Conference on Data Engineering, Atlanta, Georgia, USA, **(2006)**, pp. 24-35.
[6] H. Wang, "Privacy-Preserving Data Sharing in Cloud Computing", Journal of Computer Science and Technology, vol. 25, no. 3, **(2010),** pp. 401-414.
[7] Y. Shi, Z. Jiang and K. Zhang, "Policy-Based Customized Privacy Preserving Mechanism for SaaS Applications", GPC: Grid and Pervasive Computing, vol. 7861, **(2013)**, pp. 491-500.
[8] Apache Hbase. http://hbase.apache.org/.
[9] MongoDB. http://www.mongodb.org/.
[10] W. T. McCormick Jr., P. J. Schweitzer and T. W. White, "Problem Decomposition and Data Reorganization by a Clustering Technique", Operations Research, **(1970)**, pp. 993-1009.

Authors



Yali Shao, she was born in 1987. M. S. candidate. Her research interests include cloud computing and privacy preserving.

Yuliang Shi, he was born in 1978. Ph. D. and associate professor. His research interests include service computing, cloud computing and database. He is a membership of China Computer Federation (E200011607M).

Hui Li, she was born in 1967. Ph. D. and associate professor. Her research interests include business process management, cloud computing and database.

