Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **2**, March **2017**

A Multi-keyword Search Algorithm Based on Polynomial Function and Safety Inner-product Method in Secure Cloud Environment

Haiyun Li[1], Haifeng Li[2*];, Kaibin Wei[1], Shou-Lin Yin[3] and Chu Zhao[3]

[1]School of Electronic Information and Electrical Engineering
Tianshui Normal University
Tianshui, 741001, China
[2]School of Software
Dalian University of Technology
Dalian 116620, China
*Corresponding author:lihaifeng8848@ mail.dlut.edu.cn
[3]Software College
Shenyang Normal University
Shenyang 110034, China

Received August, 2016; revised December, 2017

Abstract. *As we all know, there are a lot of encrypted data in the cloud computing environment. Original search algorithms cannot effectively execute multi-keyword search. Therefore, a multi-keyword search algorithm based on polynomial function and safety inner-product method is proposed in this paper under secure cloud environment. This proposed method combines polynomial function with safety inner product method. The new scheme is divided into three steps. Firstly, polynomial function is adopted to hide the encrypted keyword. Secondly, we use inner-product method to protect the privacy of searched keyword, which can effectively conduct multi-keyword search and enhance the privacy of search. Thirdly, we conduct experiments from three aspects to evaluate the search ability and secrecy performance of our new method including index time cost , trapdoor generated time cost and query time cost. At last, the results indicate that multi-keyword search algorithm based on polynomial function and safety inner-product method is very effective under secure cloud environment.*
Keywords: Cloud computing environment, Multi-keyword search, Polynomial function, Safety inner-product method, Privacy.

1. **Introduction.** Currently, cloud computing[1-2] plays an increasingly important role in daily life. Cloud users can outsource their data to the cloud by remote service[3] and check the shared computing resources through the service on demand[4]. Cloud computing provides the user with a lot of convenience, such as reducing storage load etc,. With the popularity of cloud services, more and more sensitive information are outsourced to the cloud, these data needs to be encrypted, but how to make the encrypted data effectively utilized is a challenge task.

When effectively using the encrypted data, the most important thing is to authorize the keyword search. Though keywords retrieval in cloud data is difficult, it has been a hot research, many researchers have designed different strategies.

Coffman[5] presented the most extensive empirical performance evaluation of relational keyword search techniques to appear to date in the literature. Results indicated that many

existing search techniques did not provide acceptable performance for realistic retrieval tasks. In particular, memory consumption precluded many search techniques from scaling beyond small data sets with tens of thousands of vertices. He also explored the relationship between execution time and factors varied in previous evaluations. In conclusion, his work confirmed previous claims regarding the unacceptable performance of these search techniques and underscored the need for standardization in evaluations-standardization exemplified by the IR community. Fakas[6] proposed and investigated the effectiveness of size-l Object Summary(t)s and size-l Object Summary(a)s that consist of l tuple nodes and l attribute nodes respectively. Also an optimal dynamic programming algorithm, two greedy algorithms and preprocessing heuristics were proposed. By collecting feedback from real users, he assessed the relative usability of the two different types of snippets. Song[7] first made keywords retrieval for encrypted data. Liang[8] proposed an improved based on K-gram index fuzzy keywords retrieval algorithm, development language was cryptographic primitives, but it did not meet the high service requirements, such as retrieval experience or system adaptability. There are some single keyword retrieval algorithm such as [9-10].

Sun[11] presented a privacy protection multi-keyword text retrieval algorithm based on permutation method, it adopted an index structure based on tree model to improve the effectiveness of retrieval. But this structure could not effectively control data update, when adding or deleting a document, the data owners needed to update all retrieval index. And the algorithm needed an additional server to support the producing of index and trapdoor.

Cao[12] put forward a multi-keyword retrieval algorithm in cloud computing, the algorithm adopted coordinate matching to reduce the computation and communication overhead. In this strategy, users send search request to server, the server searches keyword index made by data owner. Then the server sends an encrypted document subset to users. In the process, the sever does not leak query conditions and keywords in the index. But in this strategy, it needs to build a binary vector for each document as index. Each bit in index indicates that whether the document contains the corresponding keyword. This means that the user must understand a keyword list and keyword position in a binary vector when sending query condition. The store and update of index may produce a large number of computational overhead, especially many keywords.

In this paper, our method is to eliminate the predefined binary vector which has been used in current multi-keyword retrieval strategy, and to ensure index effectively updating. It also can be extended to a large number of keywords retrieval. In order to prevent the leak of privacy in the process of multi-keyword retrieval, this paper first puts forward a multi-keyword retrieval strategy through the polynomial function to hide the encrypted keywords. In this algorithm, the query terms can be described as a polynomial function coefficient vector, so that it can prevent competitors seeing input keywords. To fight adversary equipped with strong computing resources, the new scheme in this paper combines the polynomial function method with the safety inner-product strategy. Because safety inner-product derives from K-Nearest-Neighborhood (KNN)[13-15], our new scheme does not need to establish a predefined binary index vector in the process of keywords retrieval, and it can effectively control data update.

The following are the structures of this paper. In section2, we give the data retrieval model. Section3 detailed introduces the multi-keyword retrieval algorithm. Section4 is the performance evaluation part, which is used for demonstrating the effectiveness of our method. There is a conclusion in section5.

2. **Data retrieval model.** In this section, we first describe the system model used for cloud computing. Then, some threaten models are given. Finally, we illustrate the design target for multi-keyword retrieval algorithm.

2.1. **System model.** Cloud data storage system is composed of three different entities, namely data owner, data user and cloud server as figure1. Data owner has n documents $F = (F_1, F_2, \cdots, F_n)$. After encrypting, documents will become ciphertext form $C = (C_1, C_2, \cdots, C_n)$ and they are stored in cloud. Data owner also builds a encrypted searchable index I, which makes the keywords research smoothly in C. Then data owner outsources I and C to cloud server.

FIGURE 1. Process of keywords retrieval in cloud computing

In this paper, we assuming that it has carried out appropriate data authority between data owner and user. To search the stored data document in server, user needs to set a keyword set W' formed by u keywords. Then W' will be encrypted and it generates trapdoor $T_{W'}$. A retrieval request will be sent to server. Once the server receives the $T_{W'}$, server will search the encrypted index I and return the corresponding encrypted document. In order to improve the system availability, the cloud should sort the retrieval results based on certain criteria, rather than just return disorderly results. In addition, in order to reduce the communication load, cloud server only sends the most relevant k documents to the user. Finally, the user decrypts received document through the access control mechanism.

2.2. **Threaten model.** In the cloud computing system, because system cannot maintain data unlike data owner, the system easily has security threaten. Attacker may intercept the network flow between data user and server. Specifically, in this paper we assume that attacker can deduce additional information from transformed data(i.e, encrypted data C, encrypted index I and trapdoor $T_{W'}$). According to the knowledge acquired by attacker, we take two threaten models into consideration to protect privacy information in retrieval process in cloud computing. The two threaten models are as follows.

1. Ciphertext only attack. Cloud attacker deduces the transformed encrypted data through intercepting the communication between data user and server. Under this condition, attacker collects some legal retrieval requests to produce new retrieval requests. Because this kind of attack has a deterministic property, hashing and encrypting can be used to prevent this attack.
2. Known background attack. In the robust model, the attacker further masters some background information in the database, such as the mian body of data set and statistics information of keywords. Therefore, attacker utilizes frequent captured encryption keywords and background information and can infer the right keywords. It is not safety for the protection of retrieval mode. Because the generated retrieval requests contain deterministic properties, the privacy leak problem exists in the retrieval encryption strategy.

2.3. **Our design target.** This paper designs an orderly multi-keyword retrieval strategy, this strategy can effectively protect privacy of keyword retrieval for encrypted data in the cloud computing system. Particularly, this paper design the system, which can achieve some following goals at the same time.

- Realize orderly multi-keyword retrieval. In cloud computing, this new method can conduct multi-keyword retrieval and similar data retrieval sorting at the same time.
- Realize privacy protection. The new method can prevent attacker intercepting data and getting additional information. That can protect the privacy of data.
- Realize low consumption. The new method takes a low computation and communication cost. What's more, when adding or deleting documents, it needs to update encrypted document C and retrieval index I.

3. **Multi-keyword Search Algorithm Based on Polynomial Function and Safety Inner-product Method.** In this paper, we use some symbols as following.

F: n data documents can be expressed as,

$$F = (F_i, i = 1, 2, \cdots, n). \tag{1}$$

C: n encrypted data documents in cloud can be expressed as,

$$C = (C_i, i = 1, 2, \cdots, n). \tag{2}$$

W: data owner sets n keywords set, corresponding data document is,

$$W = (W_i, i = 1, 2, \cdots, n). \tag{3}$$

Where each keyword set W_i has m_i keywords, so $W_i = (w_{i,j}, j = 1, 2, \cdots, m_i)$.

I *and* $\overline{I}$: we build retrieval index I and $\overline{I}$ respectively for each keyword of W in orderly multi-keyword retrieval strategy and privacy protection orderly multi-keyword retrieval strategy.

$$I = (I_i, i = 1, 2, \cdots, n); \qquad \overline{I} = (\overline{I}_i, i = 1, 2, \cdots, n). \tag{4}$$

Each sub-index in W_i can be expressed as: $I_i = (I_{i,j}, j = 1, 2, \cdots, m_i)$;
$\overline{I}_i = (\overline{I}_{i,j}, j = 1, 2, \cdots, m_i)$.

W': users set u keywords as,

$$W' = (w'_k, k = 1, 2, \cdots, u). \tag{5}$$

$T_{W'}$ *and* $\overline{T}_{W'}$: respectively denotes the trapdoor of W' in orderly multi-keyword retrieval strategy and privacy protection orderly multi-keyword retrieval strategy.

$C_{W'}$: top k encrypted data in list are returned to data user with trapdoor $T_{W'}$ or $\overline{T}_{W'}$.

3.1. **Main steps for new method.** The new scheme contains four main steps: setting, index creating, trapdoor and inquiry.

1. Setting. Data owner randomly generates a key SK and allocates SK to authorised user. This paper utilizes RSA public-key encryption algorithm to realize key allocation. Assuming that C delivers a public key to B and makes B trust this key belonging to A. Meanwhile, C can intercept the communication between A and B. So C can transfer his own public key to B, B believes that this key belongs to A. C can intercept all the information transformed from B to A and use his own key to decrypt this information. Then, this information will be encrypted by public key of A and transformed to A.
2. Index creating. According to keywords set W, data owner generates retrieval index I which will be encrypted by key SK. Then plaintext data set F will be encrypted as encrypted data set C. And it will publish index I and C for cloud server.

3. Trapdoor. Data user uses key SK to generate corresponding safety trapdoor $T_{W'}$ of input keywords set W'.
4. Inquiry. When cloud server receives $T_{W'}$, it will use $T_{W'}$ to search keywords in index I and return the top k document $C_{W'}$ to user.

3.2. **Orderly multi-keyword retrieval strategy based on polynomial function.**

1. Retrieval strategy based on polynomial function. In order to ensure no violation for the privacy and security request, it must hide the keywords set W' in this paper, therefore, it needs to encrypt these keywords. After generating trapdoor, it needs to build a polynomial function to hide these encrypted keywords. The detailed process of orderly multi-keyword retrieval strategy based on polynomial function is as follows.
   (a) Setting. Data owner generates a encryption function $E()$ and Hash function $H()$. It uses the two functions to construct a key $SK = (E(), H())$. Then data owner sends SK to authorized user.
   (b) Building index. Data owner extracts m_i keywords from document F_i and utilizes these keywords to construct index. To prevent attacker intercepting keywords information of index, data owner uses SK to encrypt each keyword. Keywords of encrypted document F_i can be expressed as $H(E(w_{i,j})), j = 1, 2, \cdots, m_i$. Then data owner will compute the energy consumption of encrypted keywords and build retrieval index: $I_{i,j} = (H(E(w_{i,j}))^0, \cdots, H(E(w_{i,j}))^d)^T$ (d is the maximum number of input keywords). So index information matrix of F_i is presented as:

$$I_i = (I_{i,1}, cdots, I_{i,m_i}) = \begin{bmatrix} (H(E(w_{i,1})))^0 & \cdots & (H(E(w_{i,m_i})))^0 \\ \vdots & \cdots & \vdots \\ (H(E(w_{i,1})))^d & \cdots & (H(E(w_{i,m_i})))^d \end{bmatrix} \tag{6}$$

   Data owner sends encrypted data C and index $I = I_i, i = 1, 2, \cdots, n$ of n documents to server.
   (c) Trapdoor. User sets a keyword set $W' = w'_1, \cdots, w'_u$. The u keywords are as retrieval input. To the total number of keywords is d, we add $d - u$ virtual keywords $w'_{u+1}, \cdots, w'_d$ into set W'.

$$W' = (w'_1, \cdots, w'_u, w'_{u+1}, \cdots, w'_d). \tag{7}$$

   Because each virtual key is composed of a hybrid sequence generated by random characters and numbers. And the word in virtual keyword is different from that in real dictionary, so the virtual keywords do not affect the retrieval result. Then user will utilize the key $SK = E(), H()$ to encrypt d keywords.

$$H(E(W')) = H(E(w'_k)), k = 1, \cdots, d. \tag{8}$$

   As we all know, calculation of polynomial factorization process is very complicated, especially when the polynomial order is very large. Therefore, we use polynomial function to hide the encrypted keywords. Specifically, a $d - order$ polynomial function is built.

$$f(x) = (x - H(E(w'_1))) * \cdots * (x - H(E(w'_d))) \tag{9}$$

$$= b_0 + b_1 x + \cdots + b_d x^d. \tag{10}$$

   Only when $w \in W'$, polynomial function satisfies the requirement of $f(H(E(w))) = 0$. So user can use coefficient of polynomial function to form a retrieval request and send it to cloud server.

$$T_{W'} = b_0, \cdots, {b_d}^T. \tag{11}$$

(d) Inquiry. It will create index I_i for each document F_i under trapdoor $T_{W'}$. Matrix χ_i is calculated as:

$$\chi_i = (T_{W'})^T I_i \tag{12}$$

$$= (T_{W'})^T (I_{i,1}, \cdots, I_{i,m_i}) \tag{13}$$

$$= (b_0, \cdots, b_d) \begin{bmatrix} (H(E(w_{i,1})))^0 & \cdots & (H(E(w_{i,m_i})))^0 \\ \vdots & \cdots & \vdots \\ (H(E(w_{i,1})))^d & \cdots & (H(E(w_{i,m_i})))^d \end{bmatrix} \tag{14}$$

$$= (\chi_i(1), \cdots, \chi_i(m_i)). \tag{15}$$

So when $w_{i,j} \in W'$, $\chi_i(j) = (T_{W'})^T | I_{i,j} = 0$. In this paper, we define the similarity between $T_{W'}$ and index I_i. And the similarity is denoted the number of successfully matched words. Therefore, the number of zero in χ_i can be shown as similarity. Then the server will compute zero number of corresponding matrix χ_i to get the similarity between inquiry word and each $I_i (i = 1, \cdots, n)$. These similarities will be ordered, then cloud server sends top k ID in orderly list $C_{W'}$ to user.

2. Analysis. According to the above description, the final similarity is defined as the number of inquiry keywords in data document index I_i. Hence, it should keep all the similarity in data document. When the top k ID are returned to user, the document most similar to inquiry request is obvious in the list. In addition, this new strategy does not set up predefined binary index vector for keywords. When adding a new keyword in retrieval index, data owner only computes the energy consumption of this encrypted keyword and adds a new column in retrieval index as formula (7). So the new strategy not only can effectively control the dynamic data updating, but also satisfy the privacy protection requirement of retrieval encryption strategy.
   - Ciphertext only attack. In this threaten model, attacker can intercept secret document set C, index I and trapdoor $T_{W'}$. Under this situation, it is difficult for attacker to make factorization for polynomial function used by keywords in $T_{W'}$ and $H(E(W'))$. Therefore, attacker cannot generate new retrieval request by collecting legal retrieval request. In addition, keywords in $T_{W'}$ and index are encrypted. So our new multi-keyword retrieval strategy is secure for this threaten model.
   - Known background attack. In this threaten model, attacker wants to utilize the background information of data set and frequent retrieved keywords to deduce the right keywords. The new scheme has a advantage of generating two different inquiry data for the same keywords set W', in that it can randomly produce virtual keyword. Therefore, inquiry condition cannot be generated by a determining method, attacker cannot distinguish the retrieval frequency of any keyword. And attacker cannot deduce right keywords. So our new multi-keyword retrieval strategy is also secure for this threaten model.

3.3. **Multi-keyword Search Algorithm Based on Safety Inner-product.** In this paper, the proposed retrieval strategy based on polynomial function hides keywords, which can protect the privacy information from threatening by the above two threaten models. But the method still appears privacy leak situation: attacker can deduce the encrypted keywords with factorization of polynomial function when inquiring $T_{W'}$. In addition, attacker can acquire encrypted keywords information through index retrieval. Therefore, attacker can generate a new legal trapdoor by the deduced encrypted keywords. In this

section, we propose a multi-keyword search algorithm based on safety inner-product to protect the privacy and resist the strong attacker.

1. Retrieval strategy based on safety inner-product. In this strategy, we compute the inner-product of each $I_{i,j}$ in trapdoor $T_{W'}$ and index I_i. The zero number in result Y_i is retrieval keywords number in corresponding document. In the above analysis, attacker can deduce the encrypted keywords with factorization of polynomial function. Therefore, it needs to hide trapdoor and each inner-product of index to protect privacy information. In this paper, we improve the secure KNN method and regard the new KNN method as inner-product method. The detailed introduction for the KNN inner-product method is as follows.
   - Setting. In order to improve the security for this new strategy, this paper utilizes some randomly generated matrix and vector to encrypt trapdoor and index. So it not only generates encryption function $E()$ and hash function $H()$, but produces two random $d \times d-$dimension invertible matrixes M_1, M_2 and one d bit random binary string S. Assuming $S(k)$ denotes $d-th$ bit in S. Then data owner builds key $SK = (E(), H(), M_1, M_2, S)$ and sends it to authorized user.
   - Index creating. To prevent the privacy information leak, we use the parameter M_1, M_2, S to further encrypt index. Considering that it creates the sub-index $I_{i,j} = (H(E(w_{i,j}))^0, \cdots, H(E(w_{i,j}))^d)^T : a)$ for keyword $w_{i,j}$ in retrieval process, data owner divides $I_{i,j}$ into two random vectors $I^a_{i,j}$ and $I^b_{i,j}$. For $1 \leq k \leq d$, if $S(k) = 1$, then data owner randomly divides $I_{i,j}(k)$ into $I^a_{i,j}(k)$ and $I^b_{i,j}(k)$, so $I^a_{i,j}(k) + I^b_{i,j}(k) = I_{i,j}(k)$. If $S(k) = 0$, then data owner uses $I^a_{i,j}$ and $I^b_{i,j}$ to construct $I_{i,j}(k); b$. And $I_{i,j}$ will be encrypted as $M_1^T I^a_{i,j}$ and $M_2^T I^b_{i,j}$. So in this new scheme, we construct a index for each keywords expressed as: $\overline{I}_{i,j} = (M_1^T I^a_{i,j}, M_2^T I^b_{i,j})$.
   - Trapdoor. To protect privacy information, it needs to eliminate the relationship between $T_{W'}$. So we use the parameter M_1, M_2, S to further encrypt trapdoor $T_{W'}$.
   - Building $T_{W'} = b_0, \cdots, b_d{}^T$. User adopts similar process to divide $T_{W'}$ into two random vector $T^a_{W'}$ and $T^b_{W'}$. The different is that for $1 \leq k \leq d$, if $S(k) = 0$, then data owner randomly divides $T_{W'}(k)$ into $T^a_{W'}(k)$ and $T^b_{W'}(k)$, so $T^a_{W'}(k) + T^b_{W'}(k) = T_{W'}(k)$. If $S(k) = 1$, then $T_{W'}$ will be encrypted as $M_1^{-1} T^a_{W'}$ and $M_2^{-1} T^b_{W'}$. So in this new scheme, $\overline{T}_{W'} = (M_1^{-1} T^a_{W'}, M_2^{-1} T^b_{W'})$.
   - Inquiry. Cloud server uses trapdoor $T_{W'}$ to search each sub-index $I_{i,j}$ in keyword $w_{i,j}$ and utilizes $\overline{I}_{i,j}$ to calculate $\overline{Y}_i(j)$ and judge whether this trapdoor contains the sub-keyword.

$$\begin{aligned}
\overline{Y}_i(j) &= (\overline{T}_{W'})^T \cdot \overline{I}_{i,j} && (16)\\
&= M_1^{-1} T^a_{W'}, M_2^{-1} T^b_{W'}{}^T \times M_1^T I^a_{i,j}, M_2^T I^b_{i,j} && (17)\\
&= (T^a_{W'})^T (M_1^{-1})^T M_1^T I^a_{i,j} + (T^b_{W'})^T (M_2^{-1})^T M_2^T I^b_{i,j} && (18)\\
&= (T^a_{W'})^T I^a_{i,j} + (T^b_{W'})^T I^b_{i,j} && (19)\\
&= (T_{W'})^T \times I_{i,j} && (20)\\
&= Y_i(j). && (21)
\end{aligned}$$

   Therefore, $\overline{Y}_i(j) = 0$. If $w_{i,j} \in W'$, then the zero number in $\overline{Y}_i = (\overline{Y}_i(1), \cdots, \overline{Y}_i(m_i))$ can indicates the successful match words number between $\overline{T}_{W'}$ and $\overline{I}_i$. Similarly, we use corresponding $\overline{Y}_i(i = 1, \cdots, n)$ to compute the similarity degree between

$\overline{T}_{W'}$ and $\overline{I}_i$. Finally, cloud server will send the top $k - ID$ in $C_{W'}$ to user after ranking similarity degree.

2. Analysis. Privacy protection orderly multi-keyword retrieval strategy can be used to solve privacy protection problems in orderly multi-keyword retrieval strategy. First, index vector and inquiry vector are randomly divided into two vectors. What's more, the vectors will be encrypted by M_1 and M_2 generated from the process of building index and trapdoor. Hence, attacker cannot deduce the encrypted keywords in trapdoor and retrieval index. We just encrypt parameter (M_1, M_2, S), then safety inner-product can be used to protect the privacy information of trapdoor and index. In addition, due to the random virtual keywords, it can generate two different trapdoors for the similar W' by using the paper's new method, so the new scheme also can protect retrieval model.

4. **Experience and analysis.** The experience data is composed of 150 emails and short messages. For the random input keywords set, cloud server will search all the data and find the required keywords set. We repeat 80 times for this experiment. And we use three aspects to evaluate the performance of our method including index time cost, trapdoor generated time cost and query time cost.

- Creating index time cost. Time cost on data owner building retrieval index includes time of extracting and encrypting each keyword in data document.
- Trapdoor generated time cost. Time cost on data user building trapdoor includes the time of encrypting time and the time of generating trapdoor.
- Query time cost. Time cost on cloud server completing a retrieval request includes time of computing document similarity degree and ranking time.

We set the document number range from 100 to 600 and select 40 keywords in each document. Table1 is the index time cost with different documents. And a comparison to [] and [] is shown in table1. From table1, we can know that the time cost of creating index will increase with the adding of documents. In addition, because the time of building sub-index for each document is unchanged, the relation between time cost and document number is quasi-linear.

Table 1. Index time cost with different documents

Document number	100	200	300	400	500	600
This paper's method	0.25s	0.49s	0.74s	0.93s	1.07s	1.61s
	0.35s	0.52s	8.79s	1.13s	1.22s	1.71s
	0.34s	0.51s	0.81s	1.11s	1.22s	1.77s

Then we change the keywords number in each document. The total number of document is 600 under the different keywords. So the index time with different keywords is as shown in table2. Table2 shows the effectiveness of our new method. The more keywords are, the retrieval time is higher. However, new multi-keyword search algorithm based on polynomial function and safety inner-product method can cost less time.

Furthermore, we make experience for generating index time with different keyword number as table3. In table3, when the maximum number of keyword is 30 (i.e. $d = 30$), the trapdoor generating time is the optimal with our method. From table3 we can know that the keyword number cannot affect the time cost and all the time cost is less than 0.002s.

Table 2. Index time cost with different keywords

Keyword number	10	20	30	40	50
This paper's method	0.47ms	1.12ms	1.35ms	1.57ms	1.74ms
	0.57ms	1.33ms	1.53ms	1.69ms	1.92ms
	0.58ms	1.30ms	1.54ms	1.75ms	1.89ms

Table 3. Generating trapdoor time with different keywords number

Keyword number	5	10	15	20	25	30
This paper's method	1.14ms	1.14ms	1.17ms	1.17ms	1.17ms	1.16ms
	1.40ms	1.44ms	1.40ms	1.40ms	1.43ms	1.44ms
	1.42ms	1.43ms	1.38ms	1.40ms	1.44ms	1.43ms

Table 4. Generating trapdoor time with different maximum keywords number

Maximum Keyword number	10	20	30	40
This paper's method	0.45s	0.94s	1.14s	1.23s
	0.79s	1.08s	1.46s	1.57s
	0.78s	1.11s	1.47s	1.58s

Table 5. Inquiry time with different documents

Document number	100	200	300	400	500	600
This paper's method	0.11s	0.21s	0.28s	0.34s	0.42s	0.62s
	0.12s	0.21s	0.34s	0.44s	0.54s	0.71s
	0.13s	0.21s	0.33s	0.43s	0.53s	0.69s

5. **Conclusions.** In this paper, we adopt density-based clustering method for K-anonymity privacy protection. The data will be aggregated as K clusters. Data are similar to each other in one cluster, however, they are greatly different from each other in different clusters. The data with similar sensitive attributes are aggregated together. Then it makes a clustering for the whole data according to aggregation results to achieve K-anonymous. Finally, we make a comparison from data quality and running time. The results show that when the K value increases, this new algorithm can obtain high quality data. In the future, we will study how to realize the efficient protection for huge amounts of data.

Acknowledgment. This work is supported by the Natural Science Foundation of China No.61602080 and Education Department Research Fund of Gansu Province, China (No.2013B-078). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] M. U. Shankarwar, A. V.Pawar, Security and privacy in cloud computing: A survey, *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*. Springer International Publishing, pp. 105-112, 2015.

[2] L. Wei, H. Zhu, Z. Cao, et al., Security and privacy for storage and computation in cloud computing[J]. *Information Sciences An International Journal*, vol. 258, no. 3, pp. 371-386, 2014.

[3] A. Ibrahim, H. Jin, A. A. Yassin, et al., Secure Rank-Ordered Search of Multi-keyword Trapdoor over Encrypted Cloud Data, *[C]// Apscc*, pp. 263-270, 2012.

[4] H. Flores, S. N. Srirama, C. Paniagua, A generic middleware framework for handling process intensive hybrid cloud services from mobiles*[C]// MoMM'2011-The Nineth International Conference on*

TABLE 6. Inquiry time with different maximum keywords number

Maximum Keyword number	10	20	30	40
This paper's method	0.57s	0.57s	0.61s	0.78s
	0.64s	0.66s	0.68s	0.79s
	0.61s	0.62s	0.65s	0.83s

Advances in Mobile Computing and Multimedia, 5-7 December 2011, Ho Chi Minh City, Vietnam. pp.87-94, 2011.

[5] J. Coffman, A. C. Weaver, An Empirical Performance Evaluation of Relational Keyword Search Techniques*[J]. IEEE Transactions on Knowledge & Data Engineering,* vol. 26, no. 1, pp. 30-42, 2014.

[6] G. J. Fakas, Z. Cai, N. Mamoulis, Versatile Size-, Object Summaries for Relational Keyword Search[J]. IEEE Transactions on Knowledge & Data Engineering, vol. 26, no. 4, pp.1026-1038, 2014.

[7] D. X. Song, D. Wagner, A. Perrig, Practical Techniques for Searches on Encrypted Data, *[C]// IEEE Symposium on Security and Privacy. IEEE Computer Society*, pp.44-55, 2000.

[8] Y. Liang, J. Lu, L. X. Liu, C. Zhang, Vague-keyword Search under Encrypted Environment in Cloud Computing, *[J]. Computer Science.*, vol.38, no.10.pp.99-101. 2011.

[9] C. Wang, N. Cao, J. Li, et al., Secure Ranked Keyword Search over Encrypted Cloud Data*[C]// IEEE, International Conference on Distributed Computing Systems. IEEE Computer Society*, pp.253-262, 2010.

[10] Y. C. Chang, M. Mitzenmacher, Privacy Preserving Keyword Searches on Remote Encrypted Data, *[M]// Applied Cryptography and Network Security. Springer Berlin Heidelberg*, pp.442-455, 2015.

[11] W. Sun, B. Wang, N. Cao, et al., Verifiable Privacy-Preserving Multi-Keyword Text Search in the Cloud Supporting Similarity-Based Ranking*[J]. IEEE Transactions on Parallel & Distributed Systems*, vol. 25, no. 11, pp.71-82, 2013.

[12] N. Cao, C. Wang, M. Li, et al., Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data, *[J]. IEEE Transactions on Parallel & Distributed Systems*, vol. 25, no. 1, pp.829-837, 2011.

[13] S. Zhou, Y. Zhao, J. Guan, et al., A Neighborhood-Based Clustering Algorithm, *[J]. Lecture Notes in Computer Science*, vol. 3518, pp.361-371, 2005.

[14] X. Luo, Y. Xia, Q. Zhu, et al., Boosting the K-Nearest-Neighborhood based incremental collaborative filtering, *[J]. Knowledge-Based Systems,* vol. 53, no. 9, pp.90-99, 2013.

[15] D. Tarlow, K. Swersky, I. Sutskever, Stochastic k-neighborhood selection for supervised and unsupervised learning, *International Conference on Machine Learning*, pp. 199-207, 2013.