ATLANTIS PRESS
Advances in Computer Science Research (ACSR), volume 76
7th International Conference on Education, Management, Information and Mechanical Engineering (EMIM 2017)

An Encryption Algorithm based on Matrix Supporting Fuzzy Retrieval in Cloud Computing

Ruwei Huang[1, a*], Zhikun Li[1, b], Enwei Jiang[1, c]

[1]School of Computer, Electronics and Information, GuangXi University, Nanning 530004, China

[a]ruweih@126.com, [b]2318950012@qq.com, [c]87913442@qq.com

*The Corresponding author

Keywords: Cloud computing; Privacy; Fuzzy retrieval; Encryption algorithm; Matrix operations

Abstract. With the extensive application of cloud computing, privacy has become the key problem. The traditional encryption technology can effectively guarantee the security of sensitive data, but it does not support the operations on ciphertext data directly, so that the security and computability of outsourcing data can't be taken into account together. Aiming at this problem, the paper proposes a retrievable encryption algorithm RESVMC (Retrievable Encryption Scheme based on Vector and Matrix Calculations), which realizes the prefix matching by the scalar product of vectors, and then realizes the fuzzy retrieval based on relevance ranking. The security analysis shows that RESVMC is IND-CCA (Indistinguishability under Chosen Ciphertext Attack) when the attacker only can visit the encryption Oracle and decryption Oracle of outsourced data. Compared with the existing scheme, RESVMC has smaller encryption and decryption computational loads, but the retrieval load is larger, and the storage/communication loads are slightly larger; the values of the performance indexes increase with the increase of vector dimension.

Introduction

Cloud computing provides on-demand, scalable and QoS guaranteed storage and computation resources which are delivered as services, and users can visit those services anytime and anywhere. Facing the powerful and appealing advantages of cloud computing, however, a lot of people and companies are hesitant to put their data in cloud. The main reason is that people and companies are afraid of loss of control on their data. Many famous consultants, including Gartner, have issued warnings on the privacy threats in cloud storage [1]. And the accidents of data leakage and loss happened in Google, MediaMax and Salesforce.com verify people's fears [2]. Therefore, to be sustainable, in-depth development, cloud computing must address the privacy concern.

In order to protect people's privacy, encryption is a commonly used method. Unfortunately, encryption makes effective data utilization become a very challenging task, namely, it doesn't support computations on encrypted data, for example, fuzzy string retrieval, which severely hampers the cloud service providers to provide further data management and computing services, and weakens the benefits of cloud computing. In this paper, we propose a retrieval encryption algorithm based on vector and matrix calculations (RESVMC) which supports fuzzy retrieval on encrypted strings based on relevance ranking.

The remainder of the paper is organized as follows. Firstly, introduces the related work. Secondly, gives the privacy-preserving cloud computing model and the threat model. Thirdly, provides the detailed description of RESVMC. Then evaluate the security and running overhead of RESVMC. Finally, concludes the paper and discusses future extensions.

Related Work

Liu et al. [3] proposed symmetric encryption-based ciphertext retrieval scheme. Bonech et al. [4-6] designed asymmetric encryption-based ciphertext retrieval schemes. Bellovin et al. [2,7-8] proposed Bloom Filter-based schemes. But the above schemes only support exact string retrieval. However, in many practical situations, minor typos and format inconsistencies are inevitable. So Li et al. [9]

Copyright © 2017, the Authors. Published by Atlantis Press.

This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

presented a fuzzy string retrieval scheme which exploited edit distance to quantify keywords similarity and developed an wildcard-based fuzzy keyword sets. It realized fuzzy string retrieval by several exact string matchings. Its shortcomings include that it couldn't sort the search results by similarity and it would generate too much running overheads. Besides, Wang et al. [10] proposed a secure ranked keyword search scheme based on OPSE [11], which can return the matching files in a ranked order regarding to certain relevance criteria. It demanded the owner to scan the whole file to find how many times a given keyword appears in the file, which was a burdensome work for owner. Hacigümüs et al. [12] explored techniques based on homomorphic encryption to support aggregation queries on encrypted data, which asked the owner to build the encrypted index table by himself before the data was outsourced. Hui et al. [13-14] proposed the encryption schemes base on Bloom Filter, which support fuzzy retrieval on encrypted data. But Bloom Filter has a high computational complexity and false detection rate, and the computational complexity is inversely proportional to the false detection rate, which make those methods have some limitations in the application.

From the above analysis, we find it is necessary to study an algorithm which can support fuzzy retrieval on ciphertext and has high security, and the retrieval results reflect the relevance between the plaintext and the retrieved words.

The Retrieval Encryption Scheme—RESVMC

To better realize fuzzy retrieval, we should observe people's retrieval habits. According to the construction of English words, people always consider that the meaning of "cloudy" is more similar with that of "cloud" than "clout" because "cloudy" has more same characters with "cloud", which is called prefix-matching. Second, to most of people, a word in a sentence or a phrase is more meaningful than a word as a part of another word, for example, the word "alone" in "be alone" is more meaningful than in "abalone", which is called keyword-matching. So encrypting a string is to construct its prefix ciphertext and keyword ciphertexts, which can be used to realize fuzzy retrieval by prefix matching. And the prefix matching will return the results sorted according to the following rules: (1) the strings which are equal to the query string are considered as the most matching results; (2) the strings who begin with the query string are considered as the second matching results; (3) the strings whose last character is different with that of query string are considered as the third matching results, and so on; (4) in the same matching level, the nearer the distance between the first different characters of them in ASCII table, the more similar they are.

Transformation of String. Given an outsourced string *np*, the data owner gets the substrings of *np* firstly. The basic method is splitting *np* with a space, for example, if *np* is "cloud computing", the substrings are "cloud" and "computing". Then, he conducts the following operations on the string *np* and its substrings: let's assume the computation element set ***cs*** is made up of ***n*** elements, and the length of the longest string which is ruled in the scheme is *len*=(*n*-1)*6, namely every six characters make up an element; the string *np* whose length is *len'* will generate $\lceil len'/6 \rceil$ elements and the remaining $\lceil n-len'/6-1 \rceil$ elements are zero. The *i*th element is transformed into numeric value v_i by connecting the numeric value of every character, which is computed by subtracting 23 from ASCII value of that character so as to assure that every value is double-digit. Next, calculate $v_i^{'}=v_i*10^m$, where $m=12*(n-1-i)$. So the owner gets a (*n*-1)-dimensional vector $p=(v_1^{'},v_2^{'},...,v_{n-1}^{'})$, which is the result of transformation. If *len'*>*len*, the owner will get $\lceil len'/len \rceil$ (*n*-1)-dimensional vectors. Because the following operations on the vectors are same, we assume that *np* is transformed to a vector. For the users, the way to transform query string is the same as that of outsourced string.

Encryption of Outsourced String. Suppose p_i is the (*n*-1)-dimensional vector corresponding to the outsourced string np_i. The data owner creates a *d*-dimensional vector $p_i'=(p_i,-0.5*\| p_i \|^2,r_1,w_2,...r_{k-3},w_{k-2},-(\sum_{j=1}^{k/2-1} r_{2*j-1} * \ w_{2*j-1}),1)^T$, where $\|p_i\|^2$ is the scalar product of p_i, r_j is random number and $r_j \in R$. That is to say, the computation element set is $ce=\{p_i,-0.5*\| p_i \|^2\}$, the random

element set is $re=(r_1,w_2,...r_{k-3},w_{k-2},-(\sum_{j=1}^{k/2-1}r_{2*j-1}*w_{2*j-1}),1)$. Then he splits p_i' into two vector p_{i1}' and p_{i2}' according to splitting string S: if $S[z]$=0, $p_i'[z]$ is randomly splitted into $p_{i1}'[z]$ and $p_{i2}'[z]$; if $S[z]$=1, $p_{i1}'[z]=p_{i2}'[z]=p_i'[z]$. Finally, he encrypts the vectors p_{i1}' and p_{i2}' to get the encrypted vectors $P_{i1}=M*p_{i1}'$, $P_{i2}=M*p_{i2}'$, and sends $P_i=(P_{i1},P_{i2})$ to service provider. The decryption of outsourced string P_i is the inverse process of encryption, so we won't discuss it in detail.

Encryption of Query String. We suppose the authorization between the data owner and the users is appropriately done. When a user wants to query a string nq, he generates a random number r ($r\in R^+$) and creates a vector q corresponding to nq according to the above transformation method. Then he expands q to a d-dimensional vector $q'=r*(q,1,w_1,r_2,...,w_{k-3},r_{k-2},1,-(\sum_{j=1}^{k/2-1}r_{2*j}*w_{2*j}))$, where r_j is random number and $r_j\in R$. In other words, $ce'=\{q,1\}$ and $re'=(w_1,r_2,...,w_{k-3},r_{k-2},1,-(\sum_{j=1}^{k/2-1}r_{2*j}*w_{2*j}))$. Then he splits q_i' into two vector q_{i1}' and q_{i2}' according to splitting string S: if S[z]=1, q_i'[z] is randomly splitted into q_{i1}'[z] and q_{i2}'[z]; if S[z]=0, q_{i1}'[z]= q_{i2}'[z]= q_i'[z]. Finally, he encrypts the vectors q_{i1}' and q_{i2}' to get the encrypted vectors $Q_{i1}=q_{i1}'*(M^{-1})$, $Q_{i2}=q_{i2}'*(M^{-1})$, and sends $Q_i=(Q_{i1},Q_{i2})$ to service provider for query.

Retrieval on Encrypted Strings. When the service provider receives the retrieval request, he will calculate the scalar product of Q and P_i which is in the authorized scope:

$$\begin{aligned}
&Q*P_i=((q_1',q_2')*M^{-1})*(M*(p_{i1}',p_{i2}'))\\
&=(q_1',q_2')*M^{-1}*M*(p_{i1}',p_{i2}')\\
&=(q_1',q_2')*(p_{i1}',p_{i2}')=q_1'*p_{i1}'+q_2'*p_{i2}'\\
&=r*(q,1,w_1,r_2,...,w_{t-3},r_{t-2},1,-\sum_{j=2}^{2*t}r_j*w_j)*\\
&(p_i,-0.5\|p_i\|^2,r_1,w_2,...,r_{k-3},w_{k-2},-(\sum_{j=1}^{2*t-1}r_j*w_j),1)^T\\
&=r*(p_i*q-0.5\|p_i\|^2+\sum_{j=1}^{2k}r_j*w_j-\sum_{j=1}^{2k}r_j*w_j)\\
&=r*(p_i*q-0.5\|p_i\|^2).
\end{aligned}\tag{1}$$

Then, the service provider compares the values of scalar product. The larger the value is, the more similar the np_i is with nq. The reason is described as following: suppose P_1 and P_2 are two encrypted vectors corresponding to the outsourced strings np_1 and np_2 respectively, and Q is the encrypted vector of query string nq:

$$\begin{aligned}
&Q*P_1-Q*P_2=r*(p_1*q-0.5*\|p_1\|^2-p_2*q+0.5*\|p_2\|^2)\\
&=r*(p_1*q-0.5*\|p_1\|^2-p_2*q+0.5*\|p_2\|^2-0.5*\|q\|^2+0.5*\|q\|^2)\\
&=-0.5*r*(\|p_1\|^2-2p_1q+\|q\|^2)+0.5*r*(\|p_2\|^2-2p_2q+\|q\|^2)\\
&=0.5*r*[d^2(p_2,q)-d^2(p_1,q)]
\end{aligned}\tag{2}$$

Where $d(p,q)$ denotes Euclidean distance between vector p and q. According to Eq.2,

$$\because\ d(p_i,q)\geq 0 \text{ 且 } r\in R^+,$$

$$\therefore\ Q*P_1-Q*P_2=0.5*r*(d^2(p_2,q)-d^2(p_1,q))>0\Leftrightarrow d(p_2,q)>d(p_1,q).\tag{3}$$

So the service provider can realized fuzzy retrieval by sorting the strings according to Eq.3.

Security Analysis

Theorem 1. RESVMC is not distance-recoverable.

Proof. If RESVMC is distance-recoverable, there is a computational function f such that $\forall\ m_1,m_2\in D$ and any encryption key M, $f(Enc(m_1,M),Enc(m_2,M))=d(m_1,m_2)$. Choose two different points $x_1,x_2\in D$ and two different keys M_1 and M_2, RESVMC satisfies:

(1) $a_1=Enc(m_1,M_1)=Enc(x_1,M_2)$;

(2) $a_2=Enc(m_2,M_1)=Enc(x_2,M_2)$;

(3) $d(m_1,m_2)\neq d(x_1,x_2)$.

If RESVMC is distance-recoverable, we have:

$f(a_1,a_2)=f(Enc(m_1,M_1),Enc(m_2,M_1))=d(m_1,m_2)$,

$f(a_1,a_2)=f(Enc(x_1,M_2),Enc(x_2,M_2))=d(x_1,x_2)$;

$\therefore d(m_1,m_2)= d(x_1,x_2)$

It leads to a contradiction with (3). So RESVMC is not distance-recoverable. □

Definition 4.[Unsolvable equation] Suppose P is a set of *d-dimensional* vectors and M is a set of $d*d$ matrixes, the equation $f(P_i,M_j)=P_k$ is unsolvable if $\forall P_i,P_k \in P$ and $\forall M_j \in M$, there are s_1 unknown numbers in the left of equation and s_2 unknown numbers in the right of equation and $s_1>s_2$.

Theorem 2. When RESVMC is used to encrypt strings, (1) RESVMC is IND-CCA if an adversary can just encrypt or decrypt strings adaptively by the encryption and decryption Oracles of outsourced strings; (2) RESVMC is unsafe if the adversary can encrypt strings adaptively by encryption Oracle of query strings.

Proof. (1) An adversary A can encrypt or decrypt strings to get $t(t\in N)$ pairs of plaintexts/ciphertexts by the encryption and decryption Oracles of outsourced strings. Owing to existing of splitting string S, the elements of the splitted vectors become unknown numbers when A doesn't know the value of S. Suppose d-dimensional vectors and $d*d$ matrixes are used in RESVMC, and s_1 and s_2 indicate the number of unknown numbers in the left and right of equation respectively, we can get $s_1=d*d+t*2*d$ and $s_2=t*2*d$. So the result is $s_1>s_2$ and RESVMC is unsolvable. When A guesses the value of S by exhaust algorithm, the time complexity is $O(2^d)$. According to [19], RESVMC is actually safe when complexity grows exponentially.

We define a game G: (i) A may perform any number of encryption or decryption operations. (ii) A submits two distinct chosen strings m_0, m_1 ($|m_0|=|m_1|$) to CESCMC. CESCMC selects a bit $b\in\{0,1\}$ uniformly at random, and sends the ciphertext c_b of m_b to A; (iii)A is free to perform any number of additional encryption or decryption operations except decrypting c_b; (iv) A returns a guess for the value of b.

From the ciphertext constructions of outsourced and query strings and the calculation rules of vector and matrix, all the elements of result vector are affected by the elements of operand vector and operand matrix and random numbers, which makes the ciphertexts become different even the corresponding plaintexts are same. According Theorem 1, RESVMC is not distance-recoverable. And the ciphertext-space V is a set of d-dimensional vectors, and all elements in those vectors belong to the real number field R, the different of advantage of A in G is:

$$\mathrm{Adv}(A)=|\Pr[b'=b]-\tfrac{1}{2}|=|(\tfrac{1}{2}+\tfrac{1}{|R|})-\tfrac{1}{2}|=\tfrac{1}{|R|}.$$

Obviously, the advantage of A is negligible. So RESVMC is IND-CCA if an attacker can just encrypt or decrypt strings adaptively by the encryption and decryption Oracles of outsourced strings.

(2) Suppose there are t ciphertexts $\{P_1,P_2,\ldots,P_t\}$ in the database and their corresponding plaintexts are $\{np_1,np_2,\ldots,np_t\}$, an adversary A can perform the following attack: (i) A chooses a minimum string, for example, a string consisting of a space, and encrypts it by encryption Oracle of query string and gets the ciphertexts Q_0. Then, A does the following calculations:

$Q_0 *P_i-Q_0 *P_j=d^2(p_j,q_0)-d^2(p_i,q_0)>0\Rightarrow|np_i-nq_0|<|np_j-nq_0|$.

So A gets an ordered result $np_i<np_j<\ldots<np_t$ because nq_0 is small enough.

(ii) A repeats (i) k times and chooses k query strings $\{nq_1,nq_2,\ldots,nq_k\}$ ($nq_i<nq_j$, $i\in[0,k-1]$, $j\in[1,k]$, $i<j$), then A can get the plaintext value of P_i by the following reasons: Suppose $dis=Q*P=p*q-0.5*||p||^2$, the value of dis decreases gradually when nq comes close to np, and dis is least when $nq=np$, and dis increases gradually when nq moves away from np. So A can get the plaintext of P by successive approximation.

According to the above analysis, RESVMC is unsafe if an attacker can encrypt strings adaptively by encryption Oracle of query strings.□

Performance Analysis

We evaluate the performance of RESVMC by comparing it with fuzzy string retrieval based on OPES[12], unpadded_RSA supporting multiplication homomorphism and Paillier supporting addition homomorphism. Performance metrics include three aspects: (1) encryption and decryption

overheads; (2) computation overheads; (3) storage/communication overhead. The experiments were conducted on a campus-level cloud computing platform named "Qing Cloud", which is running on a cluster consisting of ten servers and bases on Hadoop and KVM.

Fig. 1 reflects the average encryption and decryption overheads of strings with different lengths, where the dimension of RESVMC is d=10 and the bucket numbers of input and output distributions in OPES are b_1=6 and b_2=4.

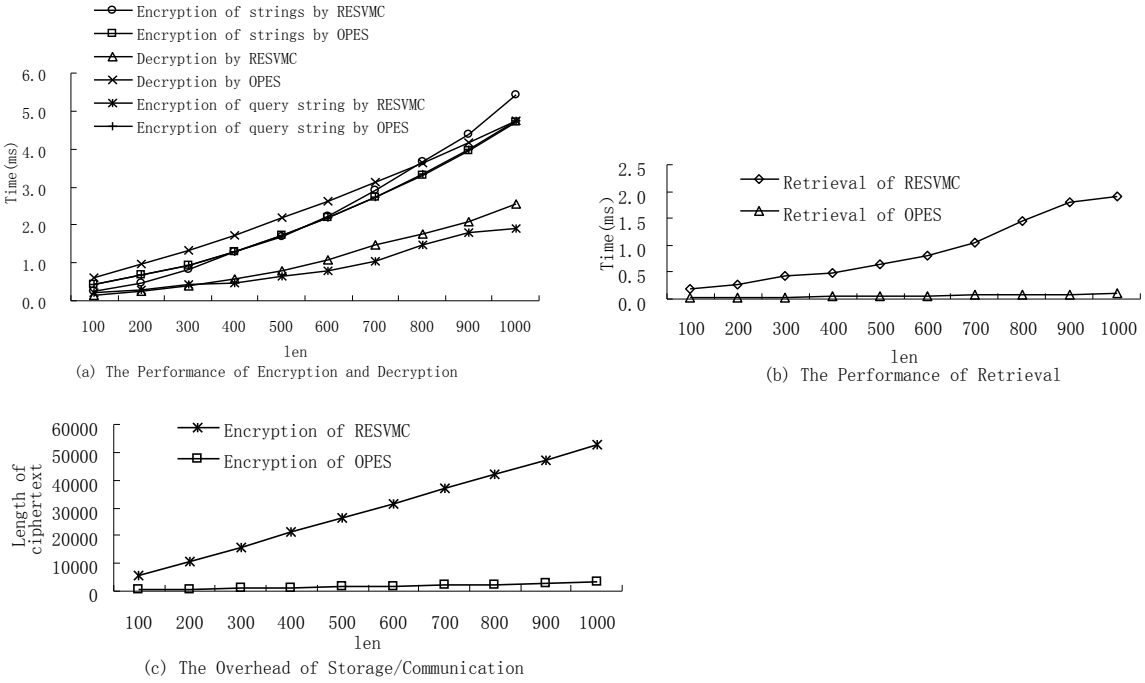

Figure 1. Finite The Performance Comparison between RESVMC and OPES

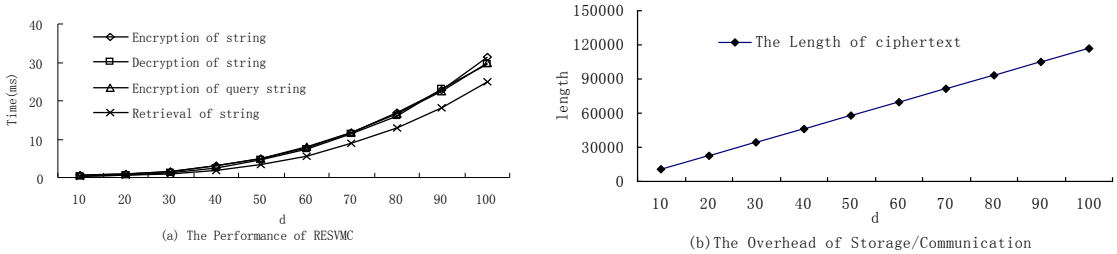

Figure 2. Finite The Performance of RESVMC with different dimensions

As illustrated in Fig. 1, RESVMC has good performance on encryption and decryption, especially decryption. But it has larger retrieval and storage/communi-cation overheads than OPES.

Fig. 2 describes the performance of RESVMC with different dimensions when the length of string is 200. According to Fig. 2, the overheads of encryption, decryption, retrieval and storage/communication increase with the increase of dimension d, and the computational complexities of encryption and decryption are $O(d^2)$, the retrieval and storage/communication complexities are $O(d)$.

In summary, RESVMC has the following characters: (1) the performances of encryption and decryption are nice; (2) fuzzy string retrieval are larger; (3) the overhead of storage/communication is larger; (4)the overheads of all performance indicators increase with the increase of dimension.

Conclusion

Aimed at the privacy protection of cloud data, we design a retrivable encryption scheme based on vector and matrix calculations (RESVMC), which supports fuzzy retrieval on encrypted strings. Security analysis and performance evaluation show: (1)RESVMC is IND-CCA if an adversary can

only encrypt or decrypt strings adaptively by the encryption and decryption Oracles of outsourced strings, and it is unsafe if the adversary can encrypt strings adaptively by encryption Oracle of query strings; (2) RESVMC has good performance on encryption, decryption, but the overheads of multiplication, division and fuzzy string retrieval are larger, and the overhead of storage/communication is larger too; (4) the overheads of all performance indicators increase with the increase of dimension.

In the future, we plan to improve the fuzzy retrieval load and storage/communication load of the scheme and find a balance between the dimension and the performance indicators.

Acknowledgements

This work was supported in part by National Science Foundation of China under Grant No.61640203 and No.61363003, the Natural Science Foundation of Guangxi Province of China under Grant No.2016GXNSFAA380115, National Science and Technology Support Programme Project under Grant No. 2015BAH55F02, the science foundation of Guangxi university under Grant No.XBZ120257 and No. XJZ151321.

References

[1] Gartner: Assessing the Security Risks of Cloud Computing(ID Number: G00157782), 2008.
[2] R.W. Huang, X.L. Gui, S. Yu and W. Zhuang: Study of Privacy-preserving Framework for Cloud Storage, Computer Science and Information Systems, Vol. 8(2011) No.3, pp.801.
[3] Q. Liu, G.J. Wang and J. Wu: Proc. IEEE Symp. Computational Science and Engineering(China, 2009). Vol. 1, p.715.
[4] D. Bonech, G.D. Crescenzo, R. Ostrovsky and G. Persiano: Proc. Eurocrypt 2004(Switzerland,2004). Vol.1, p.506.
[5] D.X. Song, P. Wagner and P. Perrig: Proc. IEEE Symp. Security and Privacy(USA, 2000). Vol.1, p. 44.
[6] W.C. Wang, Z.W. Li, R. Owens and B. Bhargava: Proc. the 2009 ACM workshop on cloud computing security(USA, 2009). Vol.1, p.55
[7] S.M. Bellovin and W.R. Cheswick: Technical Report 2004/022(IACR ePrint Cryptography Archive, America, 2004), p.1.
[8] Y. Ohtaki: Proc. the 3th International Conference on Availability, Reliability and Security(SPAIN, 2008). Vol. 1, p.1083.
[9] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren and W.J. Lou: Proc. the 29th Conference on Computer Communications (USA, 2010). Vol. 1, p.1.
[10] C. Wang, N. Cao, J. Li, K. Ren and W.J. Lou: Proc. the 30th International Conference on Distributed Computing Systems(Italy, 2010). Vol. 1, p.253.
[11] A. Boldyreva, N. Chenette, Y. Lee and A. O'Neill, Proc. the 28th Annual International Conference on Advances in Cryptology(Germany, 2009),Vol. 1, p.224.
[12] H. Hacıgümüş, B. Iyer and S. Mehrotra: Proc. the 9th International Conference on Database Systems for Advanced Applications (Korea, 2004). Vol. 1, p.633.
[13] Z. Hui, D.G. Feng, M.N. Zhang and H. Cheng: A Secure Index Against Statistical Analysis Attacks, Journal of Computer Research and Development, Vol.54(2017) No.2, p.295.(In Chinese)
[14] K.X. Wang, Y.X. Li, F.C. Zhou and Q.Q. Wang: Multi-Keyword Fuzzy Search over Encrypted Data, Journal of Computer Research and Development, Vol.54(2017) No.2, p.348. (In Chinese)