

An Effective Fine-Grained Search Algorithm for Data Access with Cloud Security

Lokka Nithya Ravi Kumar¹ and Gobi Natesan²

^{1,2}Department of Computer Science and Engineering, Dr. Mahalingam College of Engineering and Technology, Pollachi
Email: lokkanithya213@gmail.com, gobi@mcet.in

Abstract—Cloud computing allows the users to access the group of resources available in remote servers but it has many challenges while securing the user's data. However, the security is ensured by authenticating the hierarchy of user's. Authentication provides access control by checking user's credentials in the database of authorized users. The need for the users to use the private cloud is to securely store the data and also to minimize the communication cost. The cloud security is improved by verifying the user's hierarchy and reduces communication cost by using two non-colluding cloud servers; hence a searchable encryption method has been developed. Search is performed over index and returns a token of matching documents for the user to get access to the encrypted data. The data owner need not be online, to interact with the group of users. Besides, the files cannot be found if the user's hierarchy does not satisfy the access policy, even the search query containing the required keyword.

Index Terms— Cloud computing; Searchable encryption; Query privacy; Keyword search; Communication cost.

I. INTRODUCTION

In cloud computing, users lose their confidentiality of sensitive data while transferring the files through the internet. To improve the security, the sensitive data is encrypted before it transfers to the cloud. Users can't learn the content of encrypted files without the decryption key. In searchable encryption (SE) scheme, data owner encrypts the keywords as index and uploads them to cloud server together with the ciphertext. To protect the sensitive information, files will be encrypted while storing in the cloud. To perform a search, the user generates a token and sends to the cloud server. With the help of the token, cloud server searches for corresponding files and returns them to the user. The token generation algorithm encrypts the keyword and secret key to form a new token.

Searchable encryption helps the users to retrieve the required documents with use of related keyword, and in the search process, cloud servers would not get any information about the keyword and the contents of the stored files. Most of the searchable encryption schemes and most of the existing schemes are limited to single-user setup where the data owner is also the single user who generates the encrypted data on the cloud to perform encrypted keyword searches on data. There is a way to extend single-user to multiple-users is to share their public keys. So, every user can upload the encrypted data, and every user can also search the required files over all the ciphertexts. On the other hand, the search privilege is categorized according to the hierarchy of users.

For instance, cloud server stores sensitive data of two organizations, A and B. Somebody in Organization A wants to learn about the —share market, so the user searches for corresponding files in the cloud. We can assume that totally 50 files are returned but only 20 files belong to Organization.

A. So, the user can only decrypt those 20 files out of 50 files. Though the user learns those 20 files user has the right to obtain, the man gets the knowledge that there are other 30 files tagged —share market in Organization B and many other Organization files about —share market also have been retrieved. The user in Organization A can learn about the—share market in Organization B from this side information. In many cases, any information about the confidential files should not be revealed. If the user does not have the right to decrypt the ciphertext, then the user should not obtain any information about the files even if it contains the required keyword. Obviously, for the security of the files, the servers do not need to return the ciphertext which cannot be decrypted by users even they contain the corresponding keywords. This means the ciphertext should not be found by the servers if the user cannot decrypt them. The ciphertext should not be found by the removed users so that the attribute-based encryption (ABE) is used for access control. In attribute- based access control system, attributes are used to determine the access permission. This results in fine-grained access control for users.

A. Motivations

These are the few motivations available for this system such as,

Authorization

A cloud service manager is fully trusted and also responsible for generating and distributing the secret key for data users. However, this technique not only experiencing high communication cost, but also can turn out to be bottleneck and single point of failure for the entire system. Therefore, it is significant to learn searchable encryption system without the third party. In this paper, the aim is to construct a secure searchable encryption scheme without the third party.

Traceability of users

If a removed user leaks the secret key to others for profit, then it is difficult to find out the original key owner since the secret key is shared by multiple users in the group. As such, the privacy of user will experience severe problem. Thus, it is necessary for the system to find out the removed user who purposely leaks the keys.

Communication Cost

The single user searchable encryption scheme extends to multiple users' searchable encryption to share secret key with other users. A different approach was proposed [4] in order to make multiple users to search; each user has to refer to the data owner to get the search token which requires per-query interaction with the data owner. However, this implies the data owner must be available all the time and experience high communication cost. In this paper, a secure searchable encryption scheme is constructed along with low communication cost.

B. Our Contributions

In this paper, a fine-grained search algorithm is proposed by using the attribute-based encryption technique to satisfy the access control. The main contributions are as follows.

- The cloud server can search over the files using keyword, but they can't search over the ciphertext.
- The files cannot be found if the user's hierarchy does not satisfy the access policy, even the search query containing the required keyword.
- Two non-colluding cloud servers (the servers will not communicate with each other) are used to ensure the removed users cannot retrieve the required files even if they collude with one of the servers.

C. Organization

The organization of this paper is as follows. Some of the related works along with preliminaries are mentioned in Section 2. Few research questions are given in Section 3. Goals, methods and system model of fine-grained search algorithm are given in Section 4. The security analysis is mentioned in Section 5. The efficiency analysis will be described in Section 6. Finally, conclusion will be made in Section 7.

II. RELATED WORK

The searchable encryption notation was introduced by Reference [13] for improving the security of encrypted data. Then a sequence of SSE schemes have been proposed Reference [4, 13] directing at better efficiency and security. In scheme [4], the data owner dynamically manages the search capability of the receivers and the

scheme is based on the broadcast encryption. Some schemes for multiuser setting are proposed [8, 9]. In particular, multi-user setting and traceability are the two significant progresses in SE. Reference [17] uses bloom filter to verify the search results. Though, only the data owner can update the files. Reference

[1] uses the attribute-based encryption to control the decryption of the ciphertext. The ciphertext could not be decrypted by the requesting the user which will be exposed to the server, since the search work has nothing to do with user's attributes. Recently, proposed an efficient and flexible searchable encryption scheme which supports both multi-keyword ranked search and parallel search. And then a—Greedy Depth first Search algorithm is proposed in Reference [15] to improve the efficiency of multi-keyword ranked search. In the cloud setting, access control policies need to be enforced to protect our resources. Reference [14] proposed a verifiable searchable encryption scheme in which the keywords are secure under the guessing attack. Users can efficiently verify the correctness of the retrieved files efficiently by using bloom filter.

However, the above scheme deals with symmetric encryptions. The most important application of fuzzy IBE is attribute-based encryption (ABE) Reference [6], where a party can encrypt the data for all the users who have certain set of user attributes. According to the different ways of access control, ABE can be divided into two categories: ciphertext-policy ABE and key-policy ABE. In some cases, the attributes in the ciphertext and the access hierarchy should be protected. Anonymous ABE schemes are proposed to solve the problem in Reference [12]. In order to undermine the power of centralized authority, multi-authority ABE schemes are proposed. Recently, some ABE schemes with user revocation Reference [16] have been proposed.

Preliminaries

A. Bilinear Maps

Let G_1, G_2 be the two cyclic groups, and z be a generator of G_1 , and $e : G_1 \times G_2 \rightarrow G_T$ be a map with the following properties.

1. Bilinearity: $e(z, z)^{ab} = e(z, z)^{ba}$, $a, b \in \mathbb{Z}$.
2. Non-degeneracy: There exist $x, y \in G_1$ such that $e(x, y) \neq 1$.
3. Computable: For $x, y \in G_1$, $e(x, y)$ has the ability to solve the program in efficient manner.

B. Computational Bilinear Diffie–Hellman (CBDH) Assumption

Given a cyclic group G of order p , and $g, g^a, g^b, g^{ab} \in G$ for unknown random $a, b \in \mathbb{Z}^*$. The CBDH assumption is that it is computationally inflexible to compute the value e .

$$P[A(g, g^a, g^b, g^{ab}) = e] \leq \epsilon$$

where, ϵ is negligible.

III. RESEARCH QUESTIONS

- 1) How does the server work to reduce the cost of communication?
- 2) How to grant permission for users based on user hierarchy?

IV. FINE GRAINED SEARCH ALGORITHM

A. Goals

To enable the fine-grained search for effective use of encrypted data in cloud, the system design should achieve performance and security goals as follows.

Fine-grained search

To design a searchable encryption scheme which allows the users whose attributes satisfies the access policy to conduct the search of the corresponding document. And cloud server provides the corresponding results which can be decrypted by users, instead of returning all the files which contain the required keyword.

Correctness

To make sure that the server retrieves the required keyword over the ciphertexts and returns the correct results.

Key privacy

A valid key can only be generated by authority using secret parameters of the system. Any user should not be able to modify a key or construct a key to include different attributes.

Ciphertext privacy

Given a ciphertext, if the user's attributes do not satisfy the access policy, then the user will not be able to decrypt the ciphertext, even if the user colludes with other users as long as he/she does not possess the required attributes.

Query privacy

The search query is generated by the search key which is consisting of the masked attribute keys and the hidden keyword. The adversary will not be able to gain any useful information about user's attribute keys or the keyword. The only information an adversary can learn from result of a search query is the number of searched documents.

Fine-grained search

The file can be retrieved if and only if the attributes of users satisfy the access policy and the required keywords matches with file keywords.

Verification

User could verify whether the cloud server has faithfully executed the search operations when he received the search results.

Revocability

The authority should be able to selectively revoke the users, and the revoked users cannot obtain the required files even he/she colludes with one of the servers.

B. System model

We consider the fine-grained search algorithm for multiple-users. The data users can securely share their encrypted documents to the cloud servers without losing their privacy.

Some notations of this system are mentioned in the Table I.

TABLE I. NOTATIONS USED IN THIS SYSTEM

Notation	Semantic
F	The plaintext document or document collection, denoted as a set of m documents $F = \{ , , \dots , \}$. The encrypted document collection for F , denoted as $= \{ , , \dots , \}$.
W	The secret key for encryption.
I	The keywords of the encrypted documents .
Q	The index which contains the token and the keywords w of encrypted documents, which is attached with the identifier Id of the document.
TQ	The query generated from search request.
M	The encrypted form of Q , which is called the token for the search request.
Id	The number of documents in the data collection.
	Unique identifier for every user.
	Unique identifier for every document.

The components of the system are as follows,

Authority

AU constructs the search system and generates the system parameters and the query keys for users, and AU can control the enrollment and removal of users based on access rules.

Data Owner

Files are encrypted with the secret key of data owner. Then, the encrypted files are uploaded to Server 2, and the encrypted secret keys are uploaded to Server 1. And then users generate the indices of the encrypted files and upload to cloud servers.

Users

Users can also retrieve the files which contain their required keywords by using their query keys with the help of cloud servers. Every valid user can upload the ciphertext and retrieve the required files. When receiving the search results, users first get the secret keys and then decrypt the ciphertext with the secret keys.

Cloud Service Provider

- i. Server 1 (Cloud Service Manager)

It implements the search work for users and then returns the corresponding encrypted secret keys.

- ii. Server 2 (Cloud Data Server)

It implements the search work for users and then returns the corresponding ciphertext.

V. METHODS

A. CSP Setup

An efficient Fine-Grained Search (FGS) algorithm is used for encrypted documents in cloud, which performs better than the state of the art. A mutual trust is attained between the Data Owner (DO) and the Cloud Service Provider (CSP). The data users need not sacrifice the confidentiality of their sensitive documents while using the cloud.

To improve such concerns, encryption of the sensitive documents before it transfers to the cloud has become an important risk. Encryption can preserve confidentiality, integrity and accessibility of sensitive document, but no one knows the contents of encrypted documents without decryption keys. One solution is to download and decrypt the ciphertext locally, and then search for the desired results in plaintext document. For many of the applications, this approach would be unrealistic. Another method which makes cloud server decrypts the ciphertext, run the query and sends the results back to the user. This allows the server to learn the document being enquired and makes encryption less useful.

To address this issue, a common solution has been working is Searchable Encryption (SE) scheme. In this scheme, the data owners encrypts the keywords as secure searchable index and upload them to cloud server along with the encrypted document. To perform the search, the user generates a token and sends to cloud server. With the help of the token, cloud server can perform keyword search over the encrypted document and return the corresponding documents. This scheme allows the server to search over encrypted document on behalf of user without learning information about the keyword or plaintext. By combining the searchable encryption scheme along with the secure cloud storage system the security requirements can be achieved.

Algorithm

A Fine Grained Search algorithm is composed of following concepts which has been implemented to secure the user documents,

Cloud Service Provider Setup $(, , \dots) \rightarrow (, ,$

$\{, j\})$: The agreement algorithm takes the identities of some users $(, , \dots)$ as input, and output the public information $(, , \{, j\})$ in order to generate the group public key.

Key Generation $(, , \{, j\}) \rightarrow (R, A, dk)$: The group public key generation algorithm takes public information $(, , \{, j\})$ as input, and outputs the group public key (R, A) and user's secret keys dk .

Encryption $(R, A, w) \rightarrow I$: The encryption algorithm takes input as group encryption key (R, A) and the corresponding keyword w , and produce a keyword index I which will be stored on the cloud server.

Token $(dk, w) \rightarrow TQ$: On input the secret key dk and a keyword w , user runs the Token algorithm to generate a token TQ .

$Test(I, TQ) \rightarrow 0 \text{ or } 1$: The cloud server runs test algorithm to check the given ciphertext I contain the keyword w specified by the token TQ . The test algorithm takes keyword index and token as input, and outputs 0 or 1.

1. The testing is done with the help of correctness analysis, which checks whether the keyword present in the token matches with the ciphertext available in the cloud

B. User And File Lists View

This phase helps the cloud server to manage and handle the list of users and the list of files uploaded by the data owner in the cloud server. This is achieved by displaying the lists in the server side in such a way that it clearly displays whichever user uploads the files along with the filename, which will help the cloud server to find who have uploaded which file into the cloud server. The list is not editable by cloud server because it is just fetched directly from the database, so that the document is secure.

C. Generate Index

The data owner registers the details and also includes the login details. This phase helps the owner to upload the documents which are needed to be secured by using RSA encryption algorithm. This ensures the documents to be protected from unauthorized user. Data owner has a collection of documents $F = \{f_1, f_2, \dots, f_n\}$ that the user wants to outsource to the cloud server in encrypted form while still possession the ability to search on them for effective utilization.

This scheme, will allow the data owner to build a secure searchable index I from document collection F , and then generates an encrypted document collection for F . Then the data owner outsources the encrypted collection and the secure searchable index I to the cloud server, and securely allocates the information about the token generation and document decryption to the authorized data users. Besides, the data owner is responsible for their documents stored in the cloud server. While updating, the data owner generates the information locally and sends it to the cloud server.

D. Group Public Key Generation

To update the group from current system, the scheme encrypts the secure searchable index and the ciphertext stored on the server, without updating the remaining genuine user's secret keys dk . In particular, if the scheme revokes a user from the current system, the new group public key can be set as $pk' = pk \setminus \{pk_i\}$. Otherwise if a user has been added to the current system, the new group public key can be set as $pk' = pk \cup \{pk_i\}$.

The data owner calls the encryption algorithm to re-encrypt the secure searchable index I and the ciphertext C stored on the cloud server. Given the broadcast information (pk, j) , each user U_i , $i \neq j$ updates their own secret key.

The group public key used for encryption of documents is generated using the list of users in the group, $pk = (pk_1, pk_2, \dots, pk_n)$. Let $(u_1, u_2, \dots, u_n)$ be the respective identities of the users. Parse as u_i where $i = 1, 2, \dots, n$ is a fixed number. Randomly choose $a_i, b_i \in \mathbb{Z}_p$, compute and broadcast $pk_i = (u_i^{a_i}, u_i^{b_i})$, $i = 1, 2, \dots, n$. Compute the signature of such that, $j = 1, 2, \dots, n$. Broadcast (pk, j) where $j = 1, 2, \dots, n$, $j \neq i$, i.e. the rows of following Matrix.

TABLE II. GROUP PUBLIC KEY GENERATION

	...	Public Key Information
	$\emptyset$...	$(,)$
	$\emptyset$...	$(,)$
$\vdots$	$\vdots$ $\vdots$ $\ddots$ $\vdots$	$\vdots$
	... $\emptyset$	$(,)$
Example	...	(R, A)

where,

$(u_1, u_2, \dots, u_n)$: n distinct users
 $(pk_1, pk_2, \dots, pk_n)$: Public key of n users $i = 1, 2, \dots, n$
 (R, A) : new group public key information
 $\emptyset$: $i \neq j$

E. Index Generation

Data Owner has a collection of documents $F = \{d_1, d_2, \dots, d_n\}$ that the owner wants to outsource to the cloud. The data owner builds a secure searchable index I from document collection F , and then generates an encrypted document collection C . The data owner outsources the encrypted collection and the secure searchable index I to the cloud server, and distributes the key information and document decryption to the authorized data users.

F. Upload Encrypted Data to Server

This phase explains about the process after encryption using RSA. The Data Owner outsources the encrypted collection C_i and the secure searchable index I to the cloud server as shown in Figure 1, and securely distributes the key information of token generation and document decryption to the authorized data users.

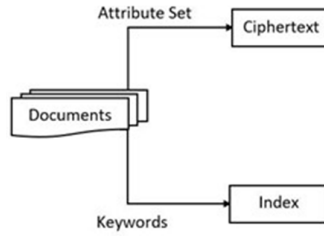


Fig. 1. Encrypted Document and Index

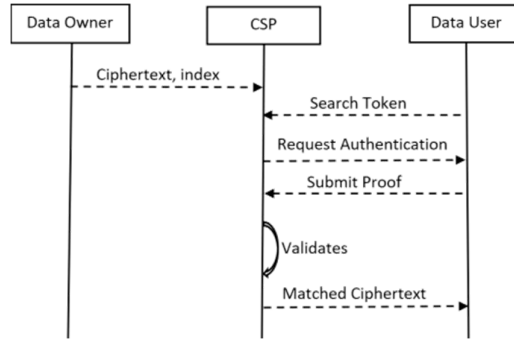


Fig. 2. Protocol of Fine-grained search scheme

This phase explains the protocol of fine-grained search scheme in Figure 2. This phase is used to help the client to search the document using the multiple key words concept and get the accurate result list based on the user query. The user is going to select the required document and get secret key to decrypt the document. Then the user can download the document and decrypt the document. Data users are authorized ones to access the documents of data owner. With keyword w , the authorized user can generate a token TQ according to search control mechanisms to fetch the encrypted documents C_i from cloud server. Then, the data user can decrypt the documents with the shared secret key.

With query keywords, the authorized user can generate a token TQ to fetch encrypted documents from cloud. Then, the data user can decrypt the documents with the shared secret key. The Data Owner will encrypt the document and outsource the ciphertext and index to cloud server. The Data User performs search, by generating a token TQ and sends as a query to the cloud server. With the help of the token, cloud server performs secure keyword search over the encrypted document C_i and return the corresponding documents. The encrypted

document that satisfies the search query will be retrieved and will be sent to the data user upon completion of the correctness test.

VI. SECURITY ANALYSIS

Theorem (Verification) Users can verify whether the cloud servers can retrieval the whole ciphertext.

Proof:

The search results returned from Server 1 is,

$$k$$

and the result returned form Server 2 are as,

$$k$$

where, denotes the number of document .

If cloud servers faithfully implement the retrieval procedure over the whole ciphertexts, then $\{ \} = \{ \}$. If one of the two non-collude cloud servers does not perform the processes and returns some random results, then $\{ \}$ $\{ \}$.

VII. RESULT ANALYSIS

The evaluation is done by using the following metrics such as,

- Correctness analysis to test whether the search keyword and keywords in the index are matched or not.
- Communication Cost is analysed using Parity Based Cryptography, which will provide the results by comparing the communication cost and number of users.

A. Correctness Analysis

The cloud server runs to check whether a given ciphertext contains the keyword w specified by the token TQ . The test takes the index and token as input, and output as 0 or 1. Once the value is responded from the CSP, the validity of the value can be judged through the verification equations correctly. The correctness of the scheme is tested, for generating a token for each keyword.

Consider the given token, the cloud server computes searchable ciphertext for each keyword. Since the token has verification functionality, the probability of cloud server output is irresistible. Once the user gets the keyword, then the user can compute token of any keyword with the valid secret key. This violates secret key privacy, which is important in the system. One possible is, the traditional Searchable Encryption scheme has been modified by using multilinear maps.

$$(\cdot, k). ((\cdot, k), f(Id)) = (\cdot, k)$$

Proof

Let the output ciphertext be $\cdot$, the group public key signature values be

Meanwhile the token and secret key would be like $[k] ((k)) k$.

$$(\cdot, k). ((\cdot, k), f(Id))$$

=

$$((k))$$

=

$$((\cdot))$$

=

=

$$=(\cdot)$$

B. Communication Cost Analysis

The group public key generation during encryption phase $(\{ \cdot, j \}, \cdot, \cdot)$ in the Cloud Service Provider for encrypting, and decrypting the required documents introduces the communication overhead of $O(\cdot)$. Then, the Cloud Service Provider reduces the communication cost by using pairing computation which results in $O(\cdot)$. The

Searchable Encryption scheme validates that the communication cost grows with the number of users in the group. Finally, the overall communication cost will be $O(n)$. Communication Cost is measured by ().

C. Efficiency Analysis

We assume n is the number of user attributes in the system. This mechanism is implemented using Java language and pairing-based cryptography (PBC) library. The process is conducted on a computer with Intel(R) Core(TM) i3-2120M CPU processor running at 2.50 GHz, 4 GB RAM. In the CSP setup phase, the match time of one file is fixed, and the total time of search will increase along with the growth of the number of the files.

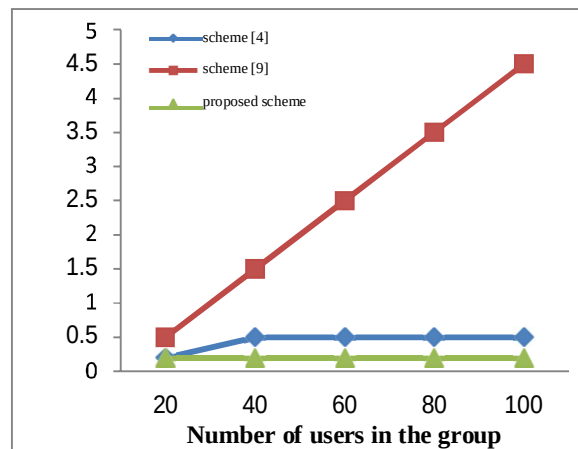


Fig. 3. Communication Cost Comparison

TABLE III. FEATURE COMPARISON WITH OTHER RELATED WORKS

	[1]	[4]	[9]	[17]	Proposed Scheme
Communication overhead					
Data Owner stay offline					
Verification					
Collusion-resistant					

VIII. CONCLUSION

In the existence, some information will not be known to the users who have no rights to obtain them, even the =information. Each user will have their own secret key, and each user generates trapdoors without getting any help from data owner or the third party. So, a novel searchable encryption scheme is proposed. The files cannot be found if the user's hierarchy does not satisfy the access policy, even the files contain the required keywords. The system model of searchable encryption is improved by using two non- colluding cloud servers. The removed user cannot obtain the required files even if the user colludes with one of the cloud servers. The user can easily verify whether the results consist of required files. The security proof and experimental evaluation illustrates that FGS scheme is secure and efficient.

REFERENCES

- [1] Cao N, Wang C, Li M, Ren K, Lou W, (2014), —Privacy- Preserving Multi-Keyword Ranked Search Over Encrypted Cloud Datal, IEEE Transactions on Parallel and Distributed Systems, vol. 25, issue. 1, pp.1.
- [2] Coron JS, Lepoint T, Tibouchi M, (2013), —Practical Multilinear Maps Over the Integersl, 33rd Annual Cryptology Conference, Springer, pp. 476–493.
- [3] Cui B, Liu Z,Wang L, (2016), —Key-Aggregate Searchable Encryption for Group Data Sharing via Cloud Storagel, IEEE Transactions on Computing, vol.65, issue. 8, pp. 2374–2385. Curtmola R, Garay J, Kamara S, Ostrovsky R, (2006),

- [4] —Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions], 13th ACM Conference on Computer and Communications Security, pp. 79–88.
- [5] H Wang, X Dong, Z Cao, (2018), —Secure and Efficient Encrypted Keyword Search for Multi-User Setting in Cloud Computing], Peer-to-Peer Networking and Applications, Springer, vol. 11, issue. 49, pp. 1–11.
- [6] Gorbunov S, Vaikuntanathan V, Wee H, (2015), —Attribute - based encryption for circuits], Journal of the ACM, vol. 62, issue. 6, pp. 45.
- [7] Jun Ye, Jianfeng Wang, Jiaolian Zhao, Jian Shen, Kuan-Ching Li, (2017), —Fine-Grained Searchable Encryption in Multi-User Setting], Soft Computing, Springer, vol. 21, issue. 20, pp. 6201-6212.
- [8] Kermanshahi SK, Liu JK, Steinfeld R, (2017), —Multi-User Cloud Based Secure Keyword Search], Australasian Conference on Information Security and Privacy, Springer, pp. 227–247.
- [9] Kiayias A, Oksuz O, Russell A, Tang Q, Wang B, (2016), —Efficient Encrypted Keyword Search for Multi-user Data Sharing], European Symposium on Research in Computer Security, Springer, LNCS vol. 9878, pp. 173-195.
- [10] Li J, Chen X, Xhafa F, Barolli L, (2015), —Secure Deduplication Storage Systems Supporting Keyword Search], International Journal of Computer and System Sciences, vol. 81, issue. 8, pp. 1532–1541.
- [11] Li J, Lin X, Zhang Y, Han J, (2016), —Outsourced Attribute- Based Encryption with Keyword Search Function for Cloud Storage], IEEE Transactions on Services Computing, vol. 10, issue. 5, pp. 1-1.
- [12] Nishide T, Yoneyama K, Ohta K, (2008), —Attribute-Based Encryption with Partially Hidden Encryptor-Specified Access Structures], International Conference on Applied Cryptography and Network Security, Springer, pp. 111–129.
- [13] Song DX, Wagner D, Perrig A, (2000), —Practical techniques for searches on encrypted data], Symposium on security and privacy, IEEE, pp. 44–55.
- [14] Wang J, Ma H, Tang Q, Li J, Zhu H, Ma S, Chen X, (2013), —Efficient Verifiable Fuzzy Keyword Search Over Encrypted Data in Cloud Computing], Computer Science and Information Systems, vol. 10, issue. 2, pp. 667–684.
- [15] Xia Z, Wang X, Sun X, Wang Q, (2016), —A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data], IEEE Transactions on Parallel Distributed Systems, IEEE, vol. 27, issue. 2, pp. 340–352.
- [16] Yang C, Ye J, (2015), —Secure and Efficient Fine-Grained Data Access Control Scheme in Cloud Computing], High Speed Networks, Springer, vol. 21, issue. 4, pp. 259–271.
- [17] Zheng Q, Xu S, Ateniese G, (2014), —VABKS: verifiable attribute based keyword search over outsourced encrypted data], IEEE conference on computer communications, pp. 522– 530.