Wireless Personal Communications (2019) 109:2133–2145
https://doi.org/10.1007/s11277-019-06672-9

Privacy Preserving Time Efficient Access Control Aware Keyword Search Over Encrypted Data on Cloud Storage

P. Shanthi[1] · **A. Umamakeswari**[1]

Published online: 11 September 2019

Abstract
Cloud computing delivers storage service to users accessed via Internet. Infrastructure used to store outsourced data is under the control of the cloud service provider. The extensive use of virtualization technology in infrastructure leads to security concern for users using public storage service. Hence, data confidentiality becomes a primary challenge in the cloud environment. Development of new technologies to protect data privacy and to provide processing capabilities to the data storage is the current requirement. This paper proposes a novel approach for access control aware keyword search over encrypted. The proposed Ciphertext-Policy Attribute-Based Keyword Search scheme allows only the authorized data users to search data stored on cloud. Encrypted index set stored along with the ciphertext on provider storage. Index set is partitioned over index server to perform parallel search. The proposed model ensures the confidentiality of data and then returns only ranked documents that match the query given by data requester. The experimental result shows that the search time reduces when using term-partitioned index set. Also, ensures security by allowing search on encrypted data without leaking any information to cloud server.

Keywords Access control · Cloud computing · Data privacy · Information security · Cryptography

1 Introduction

Cloud computing, an Internet-centric way of computing provides everything-as-a-service (infrastructure, databases, development platform and so on), thereby helping governments, enterprises, private and public institutions and many research organizations. It uses three primary models of deployment and accessibility, namely public, private and hybrid clouds. As new technology becomes popular, many new issues arise and cloud computing is not an exception. Cloud computing brings benefits to both cloud service consumers and cloud

✉ P. Shanthi
shanthi234@yahoo.co.in

A. Umamakeswari
a_umamakeswari@yahoo.com

[1] School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu, India

service providers. Besides the benefits, many challenges are posed to the cloud community such as managing large infrastructures, integrating real and virtualization technologies, allowing only authorized parties to access data, data persistence and so on. Among these, security concerning confidentiality, secrecy, and protection of data in the cloud environment is the most important. Many users, devices, and applications share the virtual cloud infrastructure. Hence data is compromised when control is delegated to cloud [1]. Data and the computations on data are concealed to protect private and sensitive information such that they are indescribable by outsiders [2].

Data encryption ensures confidentiality of data stored on third-party servers. Classical encryption methods make data unusable even by the data owner. The first work proposed to overcome this issue supports Controlled and hidden search and query isolation [3]. Following that many schemes [4–8] are developed to perform the search on encrypted data. All these proposed solutions for keyword search on encrypted data use the following elemental procedure. Data owner identifies keyword set for each file, calculates index vector and combines all index vectors to generate index file. Next, data owner uploads both encrypted documents and index files produced from the document collection to cloud storage. Cloud server then accepts queries from data users. Encrypted data query is sent to the cloud server. The server performs the search on stored index file and gives the relevant results relevant. The user selects documents needed and downloads them from the server. Finally, the user decrypts the file with the secret key. This process ensures the privacy both of data and keywords.

A single keyword search on encrypted data preserves the privacy of data [9–11]. But this keyword search yields massive coarse results. Hence searching on encrypted data is further improved to increase the search accuracy and the user searching experience. Multi-keyword search on encrypted data [12–16] enhances searching functionalities. Although many works are done to study multi ranked keyword search on encrypted data, all those solutions retrieve top matched documents and returns the identities of those files, including the unauthorized documents identifieres.

Access Control Lists (ACLs), a traditional access control technique and conventional public key cryptosystems are not suitable for cloud environment. ACLs increase the list with the number of users and the public key cryptosystems require certificate verification for data sharing. To overcome the limitations of these techniques, Sahai, and water [17] introduced Attribute-based encryption (ABE), public key cryptosystem that suits well for a cloud environment. ABE provides both confidentiality and access control for the encrypted data on cloud storage. Key-Policy Attribute-Based Encryption (KPABE) [17, 18] associates key with the policy using an access structure whereas Ciphertext-Policy Attribute-Based Encryption (CPABE) [18–21] associates data with the policy. An authorized search performed on encrypted data [22–29] using ABE. A group of users is allowed for searching in the cloud environment with this public-key encryption technique. Search functionalities enhanced to use Multi-keyword search query and ranked query for retrieving top matching authorized documents. In this paper, a new approach is proposed to integrate access control with searchable encryption. The work provides fine-grained access control as well as efficient keyword search technique. Keyword search is performed securely by two search keys SE1 and SE2 generated for each user using his unique id.

Contributions

- Integrates fine-grained access control with searchable encryption to provide data and keyword privacy.

- Improves the system performance by partitioning the index set and searching the cluster of cloud nodes in parallel.
- The proposed scheme ensures data and keyword privacy and this also compares computational complexity of proposed with existing system.

Organization The paper is organized as follows. Section 2 discusses about the prior works in this area. Section 3 describes the system model. Section 4 presents the structure of index and searching process. Section 5 illustrates the results of the experiments and finally Sect. 6 concludes the work.

2 Related Works

Verification of the data scheme [12] searches for multiple keywords by ranking them based on similarity. Ranked search encryption [13] uses keyword frequency and order-preserving encryption. Fuzzy multi-keyword search [14] is done using hashing technique based on location. An asymmetric encryption [15] that preserves a scalar product is presented and constructed two systems that support kNN computation. Also, the scheme is against practical attacks at different levels. Ruj et al. [16] proposed a solution for the mentioned problem wherein that performs the search on encrypted data by integrating access control with data confidentiality thereby allowing only authorized users to decrypt the stored information. But user attribute privacy is preserved against the provider.

ABE is used to manage keys for the files stored on the cloud server (CS) in [23]. The operation such as searching and retrieving top-ranked files are done in encrypted environment. Data Owner (DO) encrypts data indexes by embedding the access policy related attributes and sends the encrypted files to the cloud server. Data Consumer (DC) gets trapdoor from DO and sends the trapdoor to CS. Search performed by checking the authorization first and returns top files to DC for decryption using his attributes. ABE based keyword search with user revocation [24] enables authorization at the file level. Proxy re-encryption used for revocation and owner defines the access policy for the data. Scheme [25] proposes a technique which supports attribute-based functionality with proxy re-encryption and also searchable property maintained by updating search keywords. It is secure against CCA-attack.

Schemes [23–25] proposed authorized keyword search but when the number of encrypted files increases, the query processing performance degrades. Keyword Search by Outsourcing ABE (KS-OABE) [26] proposed search of the keyword to overcome the previous problem. But their system outsources key-issue and decryption. The property of supporting multi-user access control of ABE when integrated with searchable encryption ensures confidentiality of data and also preserves privacy. Attribute-Based Encryption with Keyword Search (ABEKS) [22] is developed based on the above idea. The work uses KPABE for access control in searchable encryption. Their work uses extracted document keywords as attributes and associates decryption key with the access policy constructed using the extracted keywords. Also, ABEKS applies CPABE to encrypt plaintext document for direct access control associated with users attributes. However, their work leaks the document's identifier containing the query words.

Controllable privacy-preserving search helps in managing the lifetime and search privileges of data in cloud storage. A fine-grained keyword search using Public-Key Encryption [30] preserves the privacy of data. But their work uses dual system encryption technique

that increases the computation overhead. These existing systems leak information such as user attribute, user identity, access information, etc. to the provider to perform the search. Our system preserves the privacy of both data and user which is the essential requirement for storing data in untrusted third-party storage.

3 Proposed System Model

This work integrates access control with confidentiality. Embed access control with data and perform the search on encrypted data. User authorized data alone is given a result of the search operation. The system model shown in Fig. 1 consists of five elements. Data owner (DO) uploads encrypted files along with the generated encrypted inverted index of the file collection to cloud server. Attribute-Based Encryption Authority (ABEA) is responsible for registering the user, creating searchable keys and attribute keys for the user. Data User (DU) downloads files from Cloud Server by sending the trapdoor for the query keywords received from Trapdoor Generating Authority. Trapdoor Generating Authority (TGA) generates trapdoor for the keywords given in the query. Cloud Server (CS) stores the encrypted files and searches for the user query on the stored collection.

3.1 Preliminaries

Mathematical background [20]: Follows the formal definition of Bilinear Map, Lagrange Coefficient and Access structure that forms the basis of CPABE.

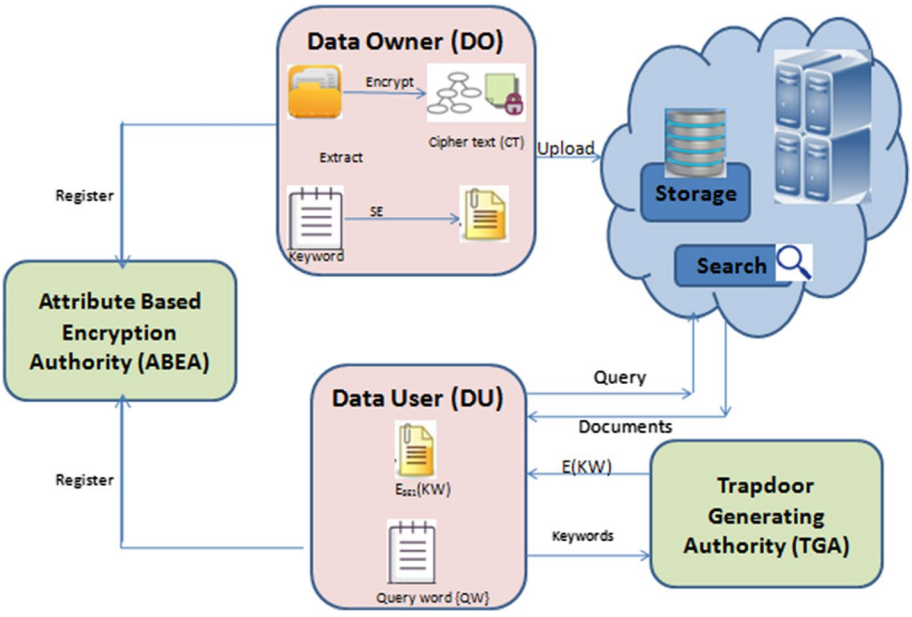

Fig. 1 System model of the proposed CPABKS

Definition 1 A Bilinear Map is a mapping function *e: G x G -> G_T*, which holds:

1. *G* and G_T are cyclic groups of same prime order p and *g* is the generator of the group. *e*(.,.) is efficiently computable;
2. For all *a*, $b \epsilon G$ and *x*, $y \epsilon Z_p$ then $e(a^x, b^y) = e(a, b)^{xy}$;
3. *Non-degeneracy:* $e(g, g) \neq 1$

Definition 2 A Lagrange Coefficient is given by, for iϵZp and a set, *A* of elements in *Zp* :

$$\Delta_{\mathrm{i,A}}(x) = \prod_{j \in A, j \neq i} \frac{x-j}{i-j}$$

Definition 3 Let $U = \{U_1, U_2, \ldots, U_N$ be the set of users. A collection $A \subseteq 2^{\mathrm{U}}$ is monotone if for all B, B', if $B \in A$ and $B \subseteq B'$ then $B' \in$ A. An access structure τ is a monotone collection of non-empty subsets of {U1, U2,…, Un}, i.e., A$\subseteq A \subseteq 2^{\mathrm{U}} \backslash \{\emptyset\}$. Therefore only the sets in *A* are called the authorized sets, and other sets are called the unauthorized sets.

3.2 Proposed System Definition

This work proposes a novel mechanism to perform authorized keyword search by integrating CPABE access control mechanism. Our CPABKS scheme uses the following algorithms for authorized search.

- **ABEA.Setup** $(\lambda, \boldsymbol{U}) \rightarrow (\boldsymbol{PK}, \boldsymbol{MSK})$

  Attribute Authority runs this algorithm using security parameter λ and description of universal set of attribute U as input. As output PK, public key and MSK, master secret key are generated.

  Let $U = \{A_1, A_2, \ldots, A_n\}$. This chooses n random numbers r_1, r_2,…, r_n and generates public component for each attribute as $PC_i = g^{r_i} \epsilon G_i$, G_1 and G_2 are two multiplicative cyclic groups of prime order p. Also randomly chooses $\alpha, \beta, x \epsilon Z_p and s \epsilon Z *_p$. Thus $\mathrm{PK} = \left(G_1, g, g^\beta, e(g,g)^\alpha, PC_1, PC_2, \ldots, PC_n\right)$ and $\mathrm{MSK} = (x, \beta, g^\alpha, A_1, A_2, \ldots, A_n)$.
- **ABEA.KeyGeneration** $\left(\boldsymbol{MSK}, \boldsymbol{U_{id}}, \boldsymbol{A}\right) \rightarrow \left(\boldsymbol{SK}_{\boldsymbol{U_{id}}}, \boldsymbol{SE1}_{\boldsymbol{Uid}}, \boldsymbol{SE2}_{\boldsymbol{Uid}}\right)$

  This algorithm takes MSK, user unique id and A, theattribute set as input. Produces $SK_{U_{id}}$, the private key and two search keys *SE*1 and *SE*2 for the user. Choose random μ and $\mathrm{r_i} \in \mathrm{Z_p}$ for each i ϵ A.

$$\mathrm{SK}_{U_{id}} = \left(\mathrm{D} = \mathrm{g}^{\frac{\alpha + \mathrm{r}U_{id}}{\beta}}, D_i = \mathrm{g}^{\frac{\mathrm{r}U_{id}}{\mathrm{A_i}}}, H(i)^{\frac{r_i}{\mathrm{A_i}}}, D_i^* = \mathrm{g}^{\mathrm{r_i}}\right), SE1_{Uid} = \mu \text{and } SE2_{Uid} = \mathrm{g}^{\frac{\mathrm{x}}{\mu}}$$

  where $\mathrm{r}U_{id}$ is random number for unique user identity.
- **DO.Encrypt** $(\boldsymbol{PK}, \boldsymbol{M}, \boldsymbol{\tau}) \rightarrow \boldsymbol{CT}$

  Data owner gives the message M, the public key PK and the access structure τ to produce ciphertext. Cipher text CT is computed as:

$$CT = (\tau, \mathrm{C}_0 = M.e(g,g)^{\alpha s}, C = g^{\beta s}, \forall x \in A : C_x = g^{sPC(x)}, C_x' = H(A(x))^s$$

- **DO.Index** $\left(\boldsymbol{KW}, \boldsymbol{SE1}_{\boldsymbol{Uid}}\right) \rightarrow (\boldsymbol{EW})$
  Data Owner index algorithm takes keyword set and key SE1 to generate encrypted index set $Ew = \left(Ew_1, \ldots Ew_m\right)$ where $EW_i = H_1\left(w_i\right)^{SE1} \epsilon G_1$.
- **CS.Postindex** $\left(\boldsymbol{EW}, \boldsymbol{SE2}_{\boldsymbol{Uid}}\right) \rightarrow (\boldsymbol{IW})$
  Cloud server converts the encrypted index set uploaded by data owner to searchable index set IW using data user DU_i key SE2 upon request from the user U_i. Thus outputs $Iw = H_2(e(Ew, SE2))$.
- **DU.Trapgen** $\left(\boldsymbol{W}, \boldsymbol{SE1}_{\boldsymbol{Uid}}\right) \rightarrow (\boldsymbol{TW})$
  Data user generates query containing the words from the keyword set W and sends the query along with his own search key $SE1$ to TGA for trapdoor generation. TGA outputs $\mathrm{TW} = H_1\left(w_i\right)^{SE1}$ to the data user.
- **CS.PostTrap** $\left(\boldsymbol{TW}, \boldsymbol{SE2}_{\boldsymbol{Uid}}\right) \rightarrow (\boldsymbol{PTW})$
  PostTrap algorithm takes the trapdoor given by user along with the requested user's search key $SE2$ to generate PostTrapWord (PTW). It calculates PTW as H_2 $(e(H_1(W_i), g)^x)$.
- **CS.Check** $(\boldsymbol{Iw}, \boldsymbol{PTW}) \rightarrow \{1, 0\}$
  Cloud server checks the searchable index set IW for the requested PTW. This step returns the user top k documents that matches the request. Cloud returns only the list of documents authorized by the user.
- **DU.Dec** $(\boldsymbol{CT}, \boldsymbol{SK}_{\boldsymbol{U_{id}}}) \rightarrow \boldsymbol{M}$
  Given ciphertext and the user secret key, the decryption algorithm recursively solves from down to top to get the plaintext.

$$Dec\left(CT, SK_{U_{id}}\right) = M.e(g,g)^{\alpha s} / \frac{e\left(D_i, C_x\right)}{e\left(D_i^*, C_x^*\right)} / e(g,g)^{r_{Uid} s} = M$$

4 Index Structure and Searching Process

The general procedure for searching large collection uses an intermediate index structure. The access control aware keyword search process consists of five main components. DO, DU, CS, Attribute authority (AA) and Trapdoor generator (TGA). Data owner extracts the keyword from the document set and creates posting lists. Each distinct keyword of the document collection is permuted and encrypted using searchable encryption. The owner then sends the ciphertext document lookup table to the cloud server. Data user generates query containing multiple keywords and sends the keyword set to trapdoor generator. Trapdoor generator produces a trapdoor for the keywords in the query and sends that to the data user. Data user provides the encrypted trapdoor, and the search key to the CS. The server performs searching using the given encrypted trapdoor and returns only identifiers of the documents authorized for his attributes.

Keyword search [27] uses three data structures lookup table T, intermediate hash table H and Array A containing the address of the first document containing the keyword. This indexing technique covers the index structure from the cloud server and reduces the search time using HPC. However, posting list for the query keyword is searched sequentially and hence retrieval time is in the order of O(n), n is the number of documents in the list. Our idea is to reduce the search time further by using term-based partitioning of the inverted index. Inverted index set is partitioned and stored in index servers. A query consisting of multiple keywords executed in parallel and

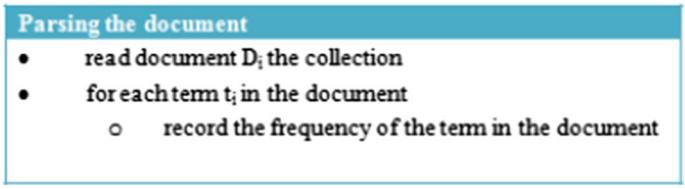

Fig. 2 Parsing document before upload

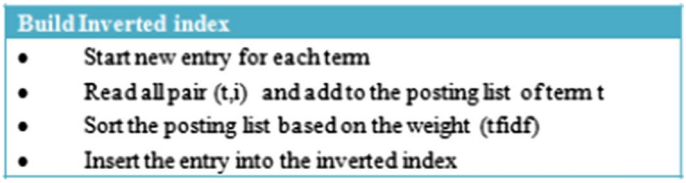

Fig. 3 Inverted index construction

searching process improved further by directing the query to the index server containing the term. As a result this returns only the top k nodes that match the given query.

4.1 Inverted Index Construction

The first step is preprocessing the document collection. Preprocess is the primary step in searching as it reduces search time and storage space. The document collection is tokenized to identify meaning semantic units. And then removes common words that form least relevant for selection from the detected list. Figure 2 gives the steps involved in the process of parsing the document.

The next step is to build inverted index set IW. Figure 3 shows the process of creating the inverted index. The posting list is sorted by Term-Frequency Inverse Document Frequency (TF-IDF) for ranked retrieval. The Inverted index set and the original documents are encrypted as shown in Table 1.

D_{id} is document id and I_W is encrypted index set. Ciphertext policy ABE, a kind of attribute based encryption uses access structure (τ) to encrypt data and secret keys (k) are generated over user attributes.

4.2 Trapdoor Generation

Secure search over encrypted data is performed by generating trapdoor for the keywords in the query. Data user requests TGA for the trapdoor. The generated trapdoor (TW) is converted to post-trapdoor (PTW) by the Cloud server before searching the index.

Table 1 Owner generated upload data format

D_{id}	I_W	(Access policy)	$E(PT)_k$

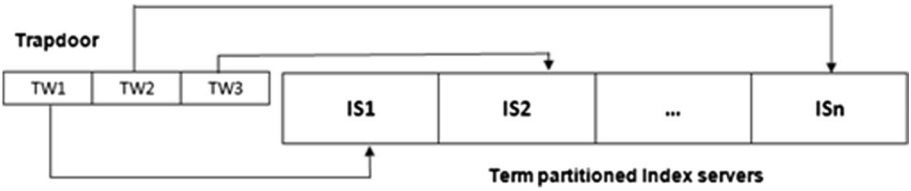

Fig. 4 Lookup table

$$T_w = H_1(W)^{SE1}$$
$$PTW = H_2\left(e\left(T_w, SE2\right)\right)$$

where SE1 is Searchable encryption key1, SE2 is Searchable encryption key2, H1 and H2 are random oracle functions.

4.3 Lookup Table and Ranked Retrieval

The query consisting of multiple keywords is split into K sub-queries as shown in Fig. 4. Each sub-query contains the query keywords for the index server such that the keyword is available in the partition. Index server, on receiving the keywords fetches the corresponding inverted list. Finally results from all partitions are collected and are processed centrally.

The response time is improved by partitioning the inverted index among $S = \{IS_1, IS_2, \ldots, ISm\}$ index servers. The storage imbalance of the index servers participating in searching process is kept under a satisfactory value by taking the storage load of index servers. Server load is given by

$$\mathbf{ISLoad}\left(\mathbf{IS_i}\right) \approx \frac{|L|}{M}, \mathbf{for\ 1} \leq \mathbf{i} \leq \mathbf{M} \tag{1}$$

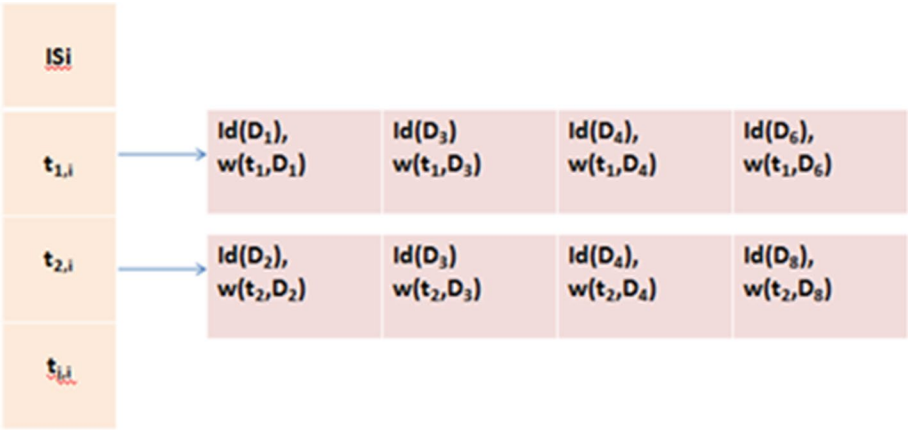

Fig. 5 Index server structure

where M is the number of index servers and L is the number of posting entries in the inverted index. The query is concurrently executed by all index servers. The way of storing the documents inverted index is shown in Fig. 5.

Searching the document collection for the query may produce coarse result. Hence the documents are scored based on TF-IDF value and only the top k documents matching the user needs are returned. Term-frequency inverse document frequency is calculated using Eq. 2.

$$\mathbf{Score}(\mathbf{w}(\mathbf{t},\mathbf{D})) = \mathbf{log}\left(f_{i,f}+\mathbf{1}\right) \times log\left[1+\frac{\mathbf{n}}{\sum_{k=1}^{n} X(f_{f,k})}\right] \tag{2}$$

where, t term, f is frequency and n is the number of documents.

4.4 Example Application Scenario

In order to share, search manage the vast amount of medical data available, cloud computing gives users HIPAA compliant computer hardware and software over the Internet. An enraging issue in data sharing and cloud retrieval system is the user privacy and data confidentiality. For example physician or cardiologist could query patient details given by primary care doctor, a surgeon could view an x-ray taken at care center. Patients embeds the access policy with the encrypted electronic health record before sending to cloud storage and could share with user's like physician or cardiologist. Data users retrieve only the authorized health records by giving trapdoor to the cloud server.

5 Experimental Analysis

The system enhances the basic Java implementation of CPABE toolkit of Stanford University to validate our scheme CPABKS proposed for authorized search on encrypted data. The experiment conducted using Intel Core i5 processor running on Ubuntu 13.0 with 8 GB RAM. A data set of 10,000 keywords is taken for experimentation. Our first work is to integrate access control with the searchable encryption. The initial phase processes the data collection and constructs the inverted index. The inverted index and the document collection are encrypted using CPABKS. TGA generated keywords matched with the encrypted document index set. The document list matching the given keyword is retrieved parallel by processing partitioned list using cluster nodes. As the document identifiers are encrypted based on the access structure, only the documents matched by the user's attribute given as a result. To produce only authorized documents as a result to the user the scheme generates keys are used for the searching process. Our second contribution is to provide parallel search over encrypted data in order to reduce search time. To implementation of our proposed system uses an HPC cluster of 32 nodes. In our proposed work, search time is the primary metric considered to improve the performance.

5.1 Complexity Analysis

Table 2 compares the complexities of our system with the existing system. The Keygeneration and decryption algorithm performance are similar. Encryption algorithm of our system needs

Table 2 Comparison of computation complexity

Scheme	KeyGen	Enc	Dec
[12]	(2S + 1)e	(2N + 3)e	(2N + 3)p + Ne
Ours	(3S + 2)e	(2N + 2)e + p	(1 + N)p + 2e

S: number of attributes of data user

N: number of attributes in data owner's policy

e: an exponentiation operation

p: a paring operation

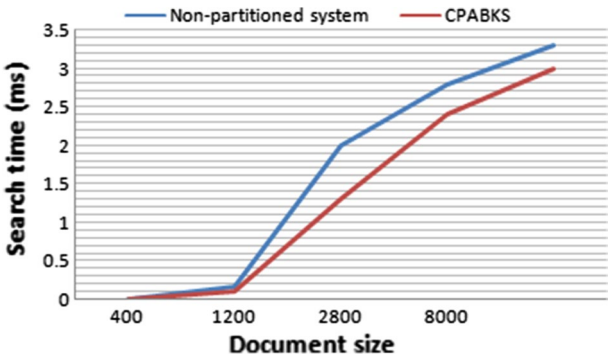

Fig. 6 Comparison of CPABKS with non-partitioned index set system

a pairing operation by cloud server as we bind two search keys to integrate authorization with search. But term-based partitioning improves response time which is very essential need for cloud storage.

5.2 Performance Analysis

The access control aware system without parallelization increases search time in linear with the number of the document in the collections for the query keywords. This work improves that by partitioning the inverted index set and storing the partitions in the index server by considering the load of the server. Figure 6 shows that searching partitioned index set reduces time. Finally, compares the search time of the proposed system with non-partitioned search system by varying the number of documents in the inverted index.

Response time for the given query plays main role secure information retrieval. Our experimental result shows that index partitioning is superior when the queries are processed in batch and it suits best for cloud environment. Figure 7 proves that there is improved response time as the number index server increases.

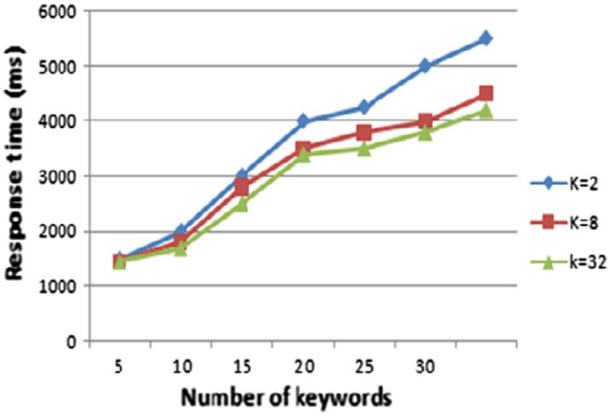

Fig. 7 Response time for varying number of query keywords

6 Conclusion

This paper proposed authorized Search on encrypted data stored on a distrusted server using public key encryption. The access control mechanism is done using CPABE which returns only authorized files to data requester and besides, improves searchable encryption performance by incorporating term-based inverted index partitioning. Also reduces search time as the index set is partitioned and achieves good response time as the keywords given in query increases. This approach is beneficial for large index set. In the future, we are planning to enhance the system to support user revocation so that encrypted data stored is secure from the users removed from the secure retrieval system.

References

1. Zissis, D., & Lekkas, D. (2012). Addressing cloud computing security issues. *Future Generation computer systems*, *28*(3), 583–592.
2. Pfleeger, C. P., & Pfleeger, S. L. (2002). *Security in computing*. Upper Saddle River: Prentice Hall.
3. Song, D. X., Wagner, D., & Perrig, A. (2000). Practical techniques for searches on encrypted data. In *Proceedings of 2000 IEEE symposium on security and privacy, 2000. S P 2000* (pp. 44–55).
4. Goh, E.-J. (2003). Secure indexes, Cryptology ePrint Archive, Report 2003/216.
5. Curtmola, R., Garay,J., Kamara, S., & Ostrovsky, R. (2006). Searchable symmetric encryption: improved definitions and efficient constructions. In *Proceedings of the 13th ACM conference on computer and communications security, CCS'06, ACM, New York, NY, USA* (pp. 79–88).
6. Liu, Q., Wang, G., & Wu, J. (2009). An efficient privacy preserving keyword search scheme in cloud computing. In: *International conference on computational science and engineering, CSE'09* (Vol. 2, pp. 715–720).
7. Boneh, D., Di Crescenzo, G., Ostrovsky, R., & Persiano, G. (2004). Public key encryption with keyword search. In C. Cachin & J. Camenisch (Eds.), *Advances in cryptology—EUROCRYPT'04* (Vol. 3027, pp. 506–522)., Lecture notes in computer science Heidelberg: Berlin.
8. Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., et al. (2005). Searchable encryption revisited: consistency properties, relation to anonymous ibe, and extensions. In V. Shoup (Ed.), *Advances in cryptology, CRYPTO'05* (Vol. 3621, pp. 205–222)., Lecture notes in computer science Heidelberg: Berlin.

9. Chang, Y.-C., & Mitzenmacher, M. (2005). Privacy preserving keyword searches on remote encrypted data. In J. Ioannidis, A. Keromytis, & M. Yung (Eds.), *Applied cryptography and network security* (Vol. 3531, pp. 391–421)., Lecture notes in computer science Heidelberg: Berlin.
10. Wang, C., Cao, N., Li, J., Ren, K., & Lou, W. (2010). Secure ranked keyword search over encrypted cloud data. In *The 30th international conference on distributed computing systems, ICDCS'10* (pp. 253–262).
11. Li, J., Wang, Q., Wang, C., Cao, N., Ren,K., & Lou, W. (2010). Fuzzy keyword search over encrypted data in cloud computing. In *IEEE conference on computer communications, INFOCOM'10* (pp. 1–5).
12. Sun, W., Wang, B., Cao, N., Li, M., Lou, W., Hou, Y. T., et al. (2014). Verifiable privacypreserving multi-keyword text search in the cloud supporting similaritybased ranking. *IEEE Transactions on Parallel and Distributed Systems, 25*(1), 222–233.
13. Wang, C., Cao, N., Li, J., Ren, K., & Lou, W. (2010). Secure ranked keyword search over encrypted cloud data. In *Proceedings of ICDCS* (pp. 253–262).
14. Wang, B., Yu, S., Lou, W., & Hou, Y. (2014). Privacy-preserved multi-keyword fuzzy search over encrypted data in the cloud. In *Proceedings of INFOCOM* (pp. 2112–2120).
15. Wong, W. K., Cheung, D. W., Kao, B., & Mamoulis, N. (2009). Secure knn computation on encrypted databases. In *Proceedings of SIGMOD* (pp. 139–152).
16. Ruj, S., Stojmenovic, M., & Nayak, A. (2012). Privacy preserving access control with authentication for securing data in clouds. In *2012 12th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGrid)* (p. 556e63).
17. Sahai, A., & Waters, B. (2005). Fuzzy identity-based encryption. In *Springer EUROCRYPT 2005*.
18. Yu, S., Wang, C., Ren, K., & Lou, W. (2010). Achieving secure, scalable, and fine-grained data access control in cloud computing. In *IEEE INFOCOM 2010*.
19. Boneh, D., & Franklin, M. (2001). Identity-based encryption from the weil pairing. In *Springer CRYPTO 2001*.
20. Waters, B. (2011). Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Springer PKC 2011*.
21. Cheung, L., & Newport, C. (2007). Provably secure ciphertext policy abe. In *ACM CCS 2007*
22. Han, F., Qin, J., Zhao, H., & Hu, J. (2014). A general transformation from KP-ABE to searchable encryption. *Future Generation Computing Systems (FGCS), 30,* 107e15.
23. Li, R., et al. (2014). Efficient multi-keyword ranked query over encrypted data in cloud computing. *Future Generation Computer Systems, 30,* 179–190.
24. Sun, W., Yu, S., Lou, W., Hou, Y., & Li, H. (2014). Protecting your right: Attribute-based keyword search with fine-grained owner-enforced search authorization in thecloud. In *Proceedings of INFOCOM* (pp. 226–234).
25. Liang, Kaitai, & Susilo, Willy. (2015). Searchable attribute-based mechanism with efficient data sharing for secure cloud storage. *IEEE Transactions on Information Forensics and Security, 10*(9), 1981–1992.
26. Li, J., Lin, X., Zhang, Y., & Han, J. (2016). KSF-OABE: Outsourced attribute-based encryption with keyword search function for cloud storage. *IEEE Transactions on Services Computing*, *10*(5), 715–725.
27. Kaci, A., & Bouabana-Tebibel, T. (2014). Access control Reinforcement over searchable encryption. In *The 15th IEEE international conference on information reuse and integration e IEEE IRI 2014*, San Francisco, USA.
28. Curtmola, R., Garay, J., Kamara, S., & Ostrovsky, R. Searchable symmetric encryption: improved definitions and efficient constructions. In *Proceedings of the 13th ACM conference on computer and communications security* (p. 79e88). New York, NY: ACM.
29. Kaci, A., Bouabana-Tebibel, T., & Challal, Z. (2014). Access control aware search on the cloud computing. In *The third international conference on advances in computing, communication and informatics e ICACCI 2014*, New Delhi, India.
30. Fan, C. I., & Huang, S. Y. (2013). Controllable privacy preserving search based on symmetric predicate encryption in cloud storage. *Future Generation Computer Systems, 29*(7), 1716–1724.

P. Shanthi received her Bachelor's degree in Computer Science and Engineering from SASTRA (formerly Shanmugha College of Engineering), Thanjavur, Tamil Nadu and Master's degree in Industrial Engineering from Guindy College of Engineering, Chennai, Tamil Nadu. She has 10 years of teaching experience and her research interests include information security and cloud computing. She is currently working as Assistant Professor in SASTRA. She has published papers in reputed journals.

Dr. A. Umamakeswari is currently working as Dean in School of Computing, SASTRA, Thanjavur, Tamil Nadu. She received her Bachelor's degree in Engineering from A.C.C.E.T., Karaikudi in 1989, Master's degree in 1994 from NIT (formerly REC), Trichy and Doctorate from SASTRA in 2009. She has 29 years of work experience and her research interests are in the area of Computer Vision, Embedded Systems, Software Engineering and Internet of Things. She has presented papers in conferences and published papers in reputed journals. She has done collaborative projects and also organized international conferences.