

A GLORIFIED SECURE SEARCH SCHEMA OVER ENCRYPTED SECURED CLOUD STORAGE WITH A HIERARCHICAL CLUSTERING COMPUTATION

MYNAM SUSHMA DEVI^{#1} and P. TULASI MAHALAKSHMI^{*2}

[#]PG Scholar, Kakinada Institute Of Engineering & Technology Department of Computer Science and Engineering, JNTUK,A.P, India

^{*} Assistant Prof, Dept of CSE, Kakinada Institute of Engineering & Technology, JNTUK, A.P, INDIA

Abstract— There are many secure objections in a multi-user strategy. First, all the users habitually keep the same secure key for trapdoor origination in a symmetric SE scheme. In this case, the revocation of the user is big demanding. If it is needed to revoke a user in this scheme, we demand to rebuild the index and disseminate the new secure keys to all the authorized users. Second, symmetric SE schemes frequently consider that all the data users are trustworthy with believable in nature, which is not at all practical and a dishonest data user will make use advantage of it which also may lead to many security issues. There may be also chances of; a dishonest data user may disseminate his/her secure keys to the unauthorized ones. In the proposed work, we improve the SE scheme to grasp these challenge problems. In our enhancement model, a hierarchical clustering method is designed to guide more search semantics with an additional feature of making the system to cope the demand for fast cipher text search in large scale environments such as Big Data scenarios. The proposed hierarchical approach clusters the cloud documents with the minimum relevance threshold, and then divides the resulting clusters into sub-clusters until the necessity on the maximum size of cluster is reached. In the search module part of the proposed system, this advent model can influence a linear computational complexity against an exponential size increase of document collection. In the manner to authenticate the authenticity of search production, an anatomy name minimum hash sub-tree is modelled in this proposed model of the system. In this paper we target on secure fetching of cloud data of an outsourced encrypted data without loss of abstraction and providing privacy and security by assigning attribute key to the data. Further we enhance the model with a multi level trust privacy preserving scheme.

Index Terms— Cloud Computing; Searchable Encryption Keyword Search; Ranked Search Anonymization; Hierarchical Clustering Method.

I. INTRODUCTION

Highlight a Cloud computing has been a long dreamed vision of computing as a utility, where the cloud customers can remotely store their data into cloud so they can enjoy the

on-demand high-quality applications and services from a shared pool of configurable computing resources. Great flexibility and economic savings in cloud computing are motivating the individuals and enterprises to outsource their local complex data management system into the cloud. In order to protect data privacy and combat unsolicited accesses in the cloud and beyond, sensitive data, for instance, emails, personal health records, photo albums, tax documents, financial transactions, etc. may have to be encrypted by data owners before outsourcing them to the commercial public cloud; this, however, obsoletes the traditional data utilization service based on plaintext keyword search. The trivial solution of downloading all the data and decrypting locally is clearly not practical, due to the huge amount of bandwidth cost in cloud scale systems. Moreover, besides from eliminating the local storage management, storing data into cloud serves no purpose unless and until they can be easily searched and utilized by users.

Thus, exploring privacy-preserving and effective search service over encrypted cloud data is of paramount importance. Considering the potentially large number of on-demand data users and huge amount of outsourced data documents in the cloud, this problem is particularly challenging as it is extremely difficult to meet also the requirements of performance, system usability and scalability. On the one hand, to meet the effective data retrieval need, the large amount of documents demand the cloud server to perform result relevance ranking, instead of returning undifferentiated results. Such ranked search system enables data users to find the most relevant information quickly, rather than burdensomely sorting through every match in the content collection [3]. Ranked search can also elegantly eliminate unnecessary network traffic by sending back only the most relevant data, which is highly desirable in the “pay-as-you use” cloud paradigm. For privacy protection, such ranking operation, however, should not leak any keyword related information.

For protection of privacy, such ranking operation, however, should not leak any keyword related information. On the other hand, to improve the search result accuracy as

well as to enhance the user searching experience, it is also essential for such ranking system to support multiple keywords search, as single keyword search often yields far too coarse results. As a common practice indicated by today's web search engines (like Google search), data users may tend to provide a set of keywords instead of only one as the indicator of their search interest to retrieve the most relevant data. Along with privacy of data and efficient searching schemes, real privacy is gained only if the user's identity remains hidden from the cloud service provider i.e. CSP as well as the third party user on the cloud server [1].

The information encryption, be that as it may, would essentially bring down the ease of use of information because of the trouble of looking over the scrambled information [6]. Essentially encoding the information may even now bring about other security concerns. Case in point, Google Search utilizes SSL (Secure Sockets Layer) to scramble the association between pursuit client and Google server when private information, for example, reports and messages show up in the indexed lists [7]. In any case, if the inquiry client clicks into another site from the query items page, that site may have the capacity to recognize the hunt terms that the client has utilized. On tending to above issues, the searchable encryption (e.g., [8], [9], [10]) has been as of late created as an essential way to deal with empower seeking over scrambled cloud information, which continues the accompanying operations. Firstly, the information proprietor needs to create a few watchwords as indicated by the outsourced information. These watchwords are then encoded and put away at the cloud server. At the point when a pursuit client needs to get to the outsourced information, it can choose some significant watchwords and send the ciphertext of the chose catchphrases to the cloud server. The cloud server then uses the ciphertext to coordinate the outsourced scrambled catchphrases, and in conclusion gives back the coordinating results to the inquiry client. To accomplish the comparative hunt effectiveness and accuracy over scrambled information as that of plaintext catchphrase seek, a broad group of exploration has been produced in writing. Wang et al. [11] propose a positioned catchphrase inquiry plan which considers the importance scores of watchwords.

Sadly, because of utilizing request safeguarding encryption (OPE) [12] to accomplish the positioning property, the proposed plan can't accomplish unlink ability of trapdoor. Later, Sun et al. [13] propose a multi-watchword content hunt plan which considers the importance scores of catchphrases and uses a multidimensional tree method to accomplish effective inquiry question. Yu et al. [14] propose a multi watch word top-k recovery plan which utilizes completely Homomorphic encryption to encode the record/trapdoor and ensures high security. Cao et al. [6] propose a multi-watchword positioned look (MRSE), which applies direction machine as the catchphrase coordinating guideline, i.e., return information with the most coordinating catchphrases. Albeit numerous inquiry functionalities have been produced in past writing towards exact and proficient searchable encryption, it is still troublesome for searchable encryption to accomplish the same client experience as that of the plaintext pursuit, similar to Google look. This for the most

part ascribes to taking after two issues. Firstly, inquiry with client inclinations is exceptionally main stream in the plaintext look [15], [16]. It empowers customized look and can all the more precisely speak to client's prerequisites, however has not been completely concentrated on and bolstered in the encoded information area. Furthermore, to further enhance the client's experience on looking, an imperative and major capacity is to empower the multi watch word seek with the far reaching rationale

A. Contribution and plan of this paper

In this research work, we analyse the security problems in cloud storage and propose a solution for the same. Our contribution can be summarized as follows:

1. For the first time, we define the problem of secure ranked keyword search over encrypted cloud data, and provide such an effective protocol, which fulfils the secure ranked search functionality with no relevance score information leakage against keyword privacy.

2. Thorough security analysis showed that our improved searchable scheme with hierarchical clustering approach using CRSA and B-tree indeed enjoys "as-strong-as possible" security guarantee compared to previous searchable symmetric encryption (SSE) schemes.

3. Extensive experimental results demonstrate the effectiveness and efficiency of the proposed solution.

II. LITERATURE SURVEY

In [2], main focus is to find the solution of multi keyword ranked search over encrypted cloud data (MRSE) while preserving strict system-wise privacy in the cloud computing paradigm. A variety of multi- keyword semantics are available, an efficient similarity measure of "coordinate matching" i.e. as many matches as possible, to capture the data documents' relevancy to the search query is used. Specifically "inner product similarity", i.e., the number of query keywords appearing in a document, to quantitatively evaluate such similarity measure of that document to the search query is used in MRSE algorithm. The important drawback of this paper is that the user's identity (ID) was not kept hidden. Hence, whoever puts the data on Cloud Service Provider (CSP) was known to the CSP. This might be risky in some situations. Hence, this limitation is overcome in proposed system. For instance, a user may want to store old email messages encrypted on a server managed by Yahoo or another large vendor, and later retrieves certain messages while travelling with a mobile device.

In [3], the schemes are efficient as no public-key cryptosystem is involved. Indeed, this approach is independent of the encryption method chosen for remote files. The main theme taken from the paper is of storing data remotely on other server and retrieving that data from anywhere through mobile, laptop etc.

In [4], an overall summary of the benefits of a cryptographic storage service, for instance, reducing the legal exposure of both customers and cloud providers, also achieving regulatory compliance is provided. Besides all this, cloud services that could be built on top of a cryptographic storage service such as secure back-ups, archival, health

record systems, secure data exchange and discovery is stated.

The [5], the paper has defined and found solution to the problem of effective yet secured ranked keyword search over encrypted cloud data. For the first time, paper has defined and solved the most challenging problem of privacy preserving multi-keyword ranked search over encrypted cloud data (MRSE), and established a set of strict privacy requirements for such a secure cloud data utilization system to become a reality. The proposed ranking method proves to be efficient to return highly relevant documents which correspond to the submitted search terms. Idea of proposed ranking method is used in the proposed system in order to enhance the security of data and the documents on Cloud Service Provider (CSP).

The [6] paper tells the significance of protecting individual's privacy in cloud computing, it also provides some privacy preserving technologies used in cloud computing services. Paper tells that it is quite important to take privacy into account while designing cloud services, if all this involves the collection, processing or sharing of personal data. From this paper, main idea taken is of privacy preserving of data. This paper importantly describes privacy of data but doesn't allow indexed search as well as the user's identity is not kept hidden. Thus, these two main limitations are overcome in the proposed system.

In [7], the paper mainly focuses on existing and new algorithms for assigning anonymous IDs and their examination with respect to trade-offs between communication and computational requirements. These algorithms are built on top of a secure sum data mining operation using Newton's identities and Sturm's theorem. The main focus in this paper is of assigning anonymous ID to the user on the cloud.

In [8], paper's main idea is to formalize and provide solution to the problem of effective fuzzy keyword search over encrypted cloud data while maintaining keyword privacy. This basic theme is taken but it is mainly for multi-keyword ranked search (MRSE scheme) in the proposed system.

In [9], design of secure cloud storage service which addresses the reliability issue with near-optimal overall performance is mainly proposed. Achieving fine-graininess, scalability, and data confidentiality of access control simultaneously is a problem for which no solution is found yet.

The paper [10] addresses this challenging open issue on one hand, defining and enforcing access policies based on data attributes, and, on the other hand, allowing the data owner to refer most of the computation tasks involved in fine-grained data access control to untrusted cloud servers without disclosing the underlying data contents.

In [11], authors have proposed a privacy-preserving public auditing system for data storage security in Cloud Computing scheme. It utilizes homomorphic linear authenticator and random masking to guarantee that the TPA

would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which eliminates the burden of cloud user from the difficult, tedious and possibly expensive auditing task, it also alleviates user's fear of his/her outsourced data leakage.

In [12], authors have defined and solved the problem of privacy preserving query over encrypted graph-structured data in cloud computing (PPGQ), and established a set of strict privacy requirements for such a secure cloud data utilization system to become a reality. The basic idea taken from this paper is of privacy preserving over encrypted cloud data. The limitation in this is that the query is not indexed to provide fast searching. This drawback is overcome in the proposed system.

III. EXISTING SYSTEM

In existing method data owner can be encrypted the documents and the cloud server, documents can be decrypted in cloud. In order to meet the practical search requirements, search over encrypted data should support the following three roles. First, the searchable encryption systems should care multi-keyword search, and offer the same user experience as searching in Google search with diverse keywords; single keyword search is far from appropriateness by only recurring very partial and imprecise search results. Although the existing schemes aim at providing integrity confirmation for different data storage systems, the problem in supporting both public auditability and data dynamics has not been fully addressed. How to achieve a protected and effective design to flawlessly integrate these two important components for data storage service remains an open stimulating task in Cloud Computing.

A. Issues in Existing System

1. Issue in mobile cloud computing.
2. Investigation of searchable encryption technique to achieve efficient searching over outsourced encrypted data
3. It fails to offer sufficient insights towards the construction of full functioned searchable encryption.

IV. PROPOSED SYSTEM

We define and solve the challenging problem of authentication and access control for multi-keyword ranked search over encrypted cloud data, and establish a set of firm authentication and access control requirements for such a secure cloud data application system to become a truth. Among various multi-keyword semantics, we had chosen the effective standard of "coordinate matching". Here we are providing foolproof security in proprietors upload side as well as on the download side. For better security client splitting that single file into nine different blocks and providing a unique identification number for each chunk. Extract keywords from the documents before encrypting and generate the synonyms for each keyword This will form hierarchical cluster of indexes that increases the search efficiency. Cluster

is a group of similar documents. Documents again form subgroups and this will continue till each document form a separate group. This forms the hierarchy of index to search the set of required documents.

The synonym search gives the relevance between the documents and clustering index method by greatly reducing the searching time as it will search only the sub group which is matched to the query keyword. When the query is entered by the user, it will start searching from the root of the hierarchy, till the relevant subgroup matched. Then a rank the documents using Jaccard coefficient measure and gives the top ranked documents as a result for the users.

A. PROPOSED SCHEME

In this section, we give a detailed description of our scheme. This system architecture as shown in the Fig.2 explains the working components namely, User, Application Server and Cloud Storage. keyword, it will be transferred over the internet and the application server process the user request by matching the index for the query keyword. It fetches the matched encrypted document from the cloud storage, decrypt it and send it back to the user. The user can specify the number of documents required in search results. The top k ranked documents will be fetched by the server and give it to the user. This will include number of steps such as Key Generation, Encryption/Decryption, Feature Vector and Synonym Generation, Document Index Generation, Query Formatting, Searching and Ranking.

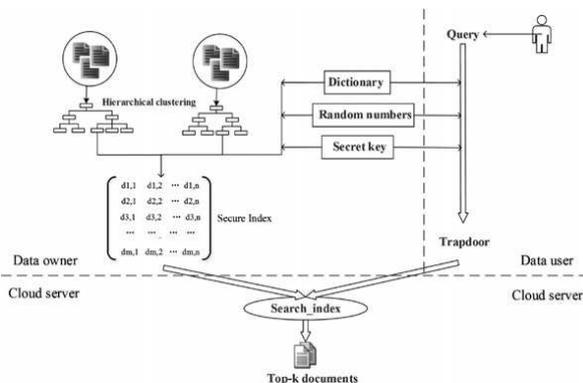


Fig.1 Proposed Architecture

- We introduce a relevance score in searchable encryption to achieve multi-keyword ranked search over the encrypted secure cloud data. In addition to that, we construct an efficient index to improve the search efficiency.

- By modifying the secure cloud storage system in the AMRSED, we solve the trapdoor disassociation problem and mask access pattern of the search user from the cloud server.

- We give thorough security analysis to demonstrate that the EMRS can reach a high security level including confidentiality of documents and index, trapdoor confidentiality, trapdoor disassociation, and covering access pattern of the search user. Moreover, we implement extensive experiments, which show that the AMRSED can achieve enhanced efficiency in the terms of functionality and search efficiency compared with existing proposals.

B. Security Requirements

Specifically, the AMRSED aims to provide the following four security requirements:

- Confidentiality of Documents and Index: Documents and index should be encrypted before being subcontracted to a cloud server. The cloud server should be prevented from prying into the subcontracted documents and cannot infer any associations between the documents and keywords using the index.

- Trapdoor Confidentiality: Since the search user would like to keep her searches from being exposed to the cloud server, the cloud server should be prevented from knowing the exact keywords contained in the trapdoor of the search user.

- Trapdoor disassociation: The trapdoors should not be linkable, which means the trapdoors should be totally different even if they contain the same keywords. In other words, the trapdoors should be randomized rather than determined. The cloud server cannot infer any associations between two trapdoors.

- Covering Access Pattern of the Search User: Access pattern is the sequence of the searched results. In the EMRS, the access pattern should be totally masked from the cloud server. Specifically, the cloud server cannot learn the total number of the documents stored on it or the size of the searched document even when the search user retrieves this document from the cloud server.

A secure cloud storage system is built on the cloud server to support adding, updating and deleting documents and covering the access pattern of the search user from the cloud server. In the secure cloud storage system, all documents are divided into fixed-size blocks. These blocks are indexed by a sequence of random integers generated by a document-related seed. In the view of a cloud server, it can only see the blocks of encrypted documents uploaded and downloaded. Thus, the blind storage system leaks little information to the cloud server. Specifically, the cloud server does not know which blocks are of the same document, even the total number of the documents and the size of each document. Moreover, all the documents and index can be stored in the blind storage system to achieve a searchable encryption scheme..

C. Advantages of proposed system

1. It addresses both spatial and temporal domains, which leads to detecting various malicious changes in spatial and time domains.
2. It is faster and lower complexity compared to existing algorithms, making it practical and suitable for real-time applications
3. Hiding Capacity of the secret data bits is high.
4. Hiding capacity was based on the pixel number corresponding to the two highest peaks of the image histogram

D. SYSTEM MODEL

The overview of the proposed system is illustrated in Figure 1. We assume that the parties are semi-honest and do not collude with each other to bypass the security measures. In Figure 1, steps and typical interactions between the participants of the system are illustrated. In an offline stage, the data owner creates a search index for each document. The search index file is created using a secret key based trapdoor generation function where the secret keys are only known by the data owner. Then, the data owner uploads these search index files to the server together with the encrypted documents. We use symmetric-key encryption as the encryption method since it can handle large document sizes efficiently. This process is referred as the index generation henceforth and the trapdoor generation is considered as its one of the steps.

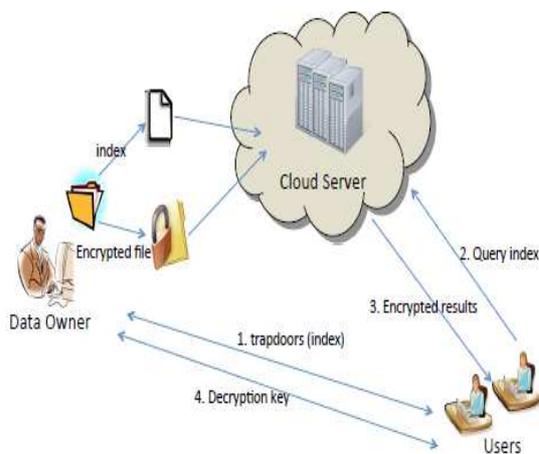


Fig .2 System Model

When a user wants to perform a keyword search, it first connects to the data owner. He learns the trapdoors (cf. Step 1 in Figure 2) for the keywords he wants to search for, without revealing the keyword information to the data owner. Since the user can use the same trapdoor for many queries containing the corresponding search term, this operation does not need to be performed every time the user performs a query. After learning the trapdoor information, the user generates the query (referred as query generation henceforth) and submits it to the server (cf. step 2 in Figure 2). In return, he receives metadata for the matched documents in a rank ordered manner as will be explained in subsequent sections. Then the user retrieves the encrypted documents (cf. Step 3 in Figure 2) he chooses after analyzing the metadata that basically conveys a relevancy level of the each matched document, where the number of documents returned is specified by the user. Finally, the user interacts with the data owner in order to decrypt the documents and get the corresponding plaintext (cf. Step 4 in Figure 2); and in the process the data owner does not learn the documents that it is assisting to decrypt.

The multi-keyword search method explained in this section, it checks whether queried keywords exist in a

document or not. If the user searches for a single or more keywords, there will possibly be many correct matches where some of them may not be useful for the user at all. Therefore, it is difficult to decide as to which documents are the most relevant. We add ranking capability to the system by adding extra index information for frequently occurring keywords in a file. With ranking, the user can retrieve only the top matches where is chosen by the user. In order to rank the documents, a ranking function is required, which assigns relevancy scores to each document matching to a given search query. One of the most widely used metrics in information retrieval is the term frequency [15]. Term frequency is defined as the number of times a keyword appears in a document. Instead of using term frequency itself, we assign relevancy levels based on the term frequencies of keywords.

We assume that there are η levels of ranking in our proposed method for some integer $\eta \geq 1$. For each document, each level stores an index for frequent keywords of that document in a cumulative way in descending order. This basically means that i^{th} level index includes all keywords in the $(i + 1)$ th level and the keywords that have term frequency for the i^{th} level. The higher the level, the higher the term frequency of the keywords is. For instance, if $\eta = 3$, level 1 index includes keywords that occur at least once in the document while levels 2 and 3 include keywords that occur at least, say 5 times and 10 times, respectively. There are several variations for relevancy score calculations [15] and we use a very basic method. The relevancy score of a document is calculated as the number representing the highest level search index that the query index matches.

All the keywords that exist in a document are included in the first level search index of that document. The other higher level indices include the frequent keywords that also occur in its previous level, but this time they have to occur the number of times, which are at least the term frequency of the corresponding level. The highest level includes only the keywords that have the highest term frequency. In the oblivious search phase, the server starts comparing the user query against the first level indices of each document. The matching documents found as a result of the comparison in the first level are then compared with the search indices in the other levels according to the Algorithm 1.

In this method, some information may be lost due to the ranking method employed here. Rank of two documents will be the same if one involves all the queried keywords frequently and the other involves all the queried keywords frequently except one infrequent one. The rank of the document is identified with the least frequent keyword of the query. We tested the correctness of our ranking method by comparing with a commonly used formula for relevance score calculation [13], which is given in the following section.

E. ALGORITHM FOR RANK SEARCH

```

for all documents  $R_i$  do
  Compare(level1 index of  $R_i$ , query index)
  j = 1
  while match do
    increment j
  Compare (level j indices of  $R_i$ , query index)
  end while
  rank of  $R_i$  = highest level that match with query index
end for
    
```

While this new method necessitates an additional r-bit storage per level for a document, it reduces the communication overhead of the user since matches with low rank documents will not be retrieved unless the user requests. Considering η search indices are stored instead of a single search index per document, storage overhead for indexing mechanism increases η times due to ranking. This additional cost is not a burden for the server since the index sizes are usually negligibly small compared to actual document sizes.

V. IMPLEMENTATION

The proposed system of this project is divided into five major modules and described as below.

1. DATA OWNER
2. SERVER
3. USER
4. CLOUD SERVER AND ENCRYPTION
5. RANK SEARCH

A. MODULES DESCRIPTION

1) USER

After receiving trapdoor keys from the data owner, query index is generated, which is essentially equivalent to performing hash operations. Subsequent to the retrieval of encrypted documents, the user should perform one blinding operation over an RSA encryption, one signing for authentication purposes and symmetric-key decryption operations. These operations are equivalent to 3 modular exponentiations, 2 modular

multiplications and one symmetric-key decryption operation per document retrieved.

2) SERVER

The server only does the search operation, which is binary comparison of r-bit query index with database indices, each of which is again r-bit binary sequence. If the ranking is used, the query index should also be compared with indices of the matching documents. So the server performs additional

binary comparison of r-bit indices for each matching document, where η is the number of levels.

3) DATA OWNER

The data owner creates the indices and symmetric-key encryptions of all documents; but these operations are performed only once in the initialization phase. Other than this, data owner is active while the user learns the trapdoors and decryption keys, which requires 2 modular exponentiations for each.

4) CLOUD SERVER AND ENCRYPTION

This module is used to help the server to encrypt the document using RSA Algorithm and to convert the encrypted document to the Zip file with activation code and then activation code send to the user for download. Cloud server stores the encrypted document collection C and the encrypted searchable tree index I for data owner. Upon receiving the trapdoor TD from the data user, the cloud server executes search over the index tree I , and finally returns the corresponding collection of top- k ranked encrypted documents. Besides, upon receiving the update information from the data owner, the server needs to update the index I and document collection C according to the received information. The cloud server in the proposed scheme is considered as “honest-but-curious”, which is employed by lots of works on secure cloud data search

5) RANK SEARCH

These modules ensure the user to search the files that are searched frequently using rank search. This module allows the user to download the file using his secret key to decrypt the downloaded data. This module allows the Owner to view the uploaded files and downloaded files. The proposed scheme is designed to provide not only multi-keyword query and accurate result ranking, but also dynamic update on document collections. The scheme is designed to prevent the cloud server from learning additional information about the document collection, the index tree, and the query.

VI. CONCLUSION

In this paper, we proposed hierarchical clustering method to ensure high data security and integrity in the cloud for secure data transmission. we have proposed a multi-keyword ranked search scheme to enable accurate, efficient and secure search over encrypted mobile cloud data. We find out the problem of maintaining the semantic relationship between different plain documents over the related encrypted documents. We propose the hierarchical clustering architecture to adapt to the requirements of data explosion, and include the synonym search to increase the relevance between documents and relevance between the query and the documents. In addition, we analyse the search efficiency and security by encrypting all the entities in the model such as indexes, keywords, synonyms and query keywords. Security analysis have confirmed that proposed scheme can effectively achieve confidentiality of documents and index, trapdoor

confidentiality, trapdoor disassociation, and covering access pattern of the search user. Wide performance appraisals have shown that the proposed scheme can achieve better efficacy in terms of the functionality and computation overhead compared with existing ones. The future work would concentrate on using Elliptic Curve Cryptography (ECC) encryption technique for better performance. Further, we intend to analyze the behaviour of our proposed system for multiuser environment

REFERENCES

- [1] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Confidentiality-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 222–233, Jan. 2014.
- [2] W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis, "Secure kNN computation on encrypted databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2009, pp. 139–152.
- [3] M. Naveed, M. Prabhakaran, and C. A. Gunter, "Dynamic searchable encryption via blind storage," in *Proc. IEEE Symp. Secur. Confidentiality*, May 2014, pp. 639–654.
- [4] H. Pang, J. Shen, and R. Krishnan, "Confidentiality preserving similarity-based text retrieval," *ACM Trans. Internet Technol.*, vol. 10, no. 1, p. 4, 2010.
- [5] H. Liang, L. X. Cai, D. Huang, X. Shen, and D. Peng, "An SMDP-based service model for interdomain resource allocation in mobile cloud networks," *IEEE Trans. Veh. Technol.*, vol. 61, no. 5, pp. 2222–2232, Jun. 2012.
- [6] M. M. E. A. Mahmoud and X. Shen, "A cloud-based scheme for protecting source-location privacy against hotspot-locating attack in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 10, pp. 1805–1818, Oct. 2012.
- [7] Q. Shen, X. Liang, X. Shen, X. Lin, and H. Y. Luo, "Exploiting geodistributed clouds for a e-health monitoring system with minimum service delay and privacy preservation," *IEEE J. Biomed. Health Inform.*, vol. 18, no. 2, pp. 430–439, Mar. 2014.
- [8] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Commun. Mobile Comput.*, vol. 13, no. 18, pp. 1587–1611, Dec. 2013.
- [9] H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-based authentication for cloud computing," in *Cloud Computing*. Berlin, Germany: Springer-Verlag, 2009, pp. 157–166.
- [10] W. Sun, et al., "Confidentiality-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in *Proc. 8th ACM SIGSAC Symp. Inf., Comput. Commun. Secur.*, 2013, pp. 71–82.
- [11] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Confidentiality-preserving multi-keyword fuzzy search over encrypted data in the cloud," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 2112–2120.
- [12] E. Stefanov, C. Papamanthou, and E. Shi, "Practical dynamic searchable encryption with small leakage," in *Proc. NDSS*, Feb. 2014.
- [13] Y. Yang, H. Li, W. Liu, H. Yang, and M. Wen, "Secure dynamic searchable symmetric encryption with constant document update cost," in *Proc. GLOBECOM*, Anaheim, CA, USA, 2014.
- [14] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Roşu, and M. Steiner, "Highly-scalable searchable symmetric encryption with support for Boolean queries," in *Proc. CRYPTO*, 2013, pp. 353–373.
- [15] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Confidentiality-preserving multikeyword ranked search over encrypted cloud data," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 222–233, Jan. 2014.
- [16] W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis, "Secure kNN computation on encrypted databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2009, pp. 139–152.
- [17] M. Naveed, M. Prabhakaran, and C. A. Gunter, "Dynamic searchable encryption via blind storage," in *Proc. IEEE Symp. Secur. Confidentiality*, May 2014, pp. 639–654.
- [18] H. Pang, J. Shen, and R. Krishnan, "Confidentiality preserving similarity-based text retrieval," *ACM Trans. Internet Technol.*, vol. 10, no. 1, p. 4, 2010.
- [19] J. Yu, P. Lu, Y. Zhu, G. Xue, and M. Li, "Toward secure multikeyword top-k retrieval over encrypted cloud data," *IEEE Trans. Dependable Secure Comput.*, vol. 10, no. 4, pp. 239–250, Jul./Aug. 2013.

- [20] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. EUROCRYPT*. Berlin, Germany: Springer-Verlag, 2011, pp. 568–588.
- [21] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. EUROCRYPT*, 2004, pp. 506–522.
- [22] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Proc. TCC*, 2007, pp. 535–554.
- [23] B. Zhang and F. Zhang, "An efficient public key encryption with conjunctive-subset keywords search," *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 262–267, Jan. 2011.

AUTHOR PROFILE



MYNAM SUSHMA DEVI is a student of Kakinada Institute Of Engineering & Technology affiliated to JNTUK, Kakinada pursuing M.Tech (Computer Science & Engineering). Her Area of interest includes Cloud Computing and its objectives in all current trends and techniques in Computer Science.



P TULASIMAHALAKSHMI ^{M.TECH} is Working as M.Tech Assistant Professor, Department of Computer Science & Engineering of Kakinada Institute Of Engineering & Technology affiliated to JNTUK, Kakinada, A.P, India.