Position Based Sentence Search for Encrypted Unstructured Data in Cloud Environment

By

Muhammad Zaman Fakhar

2011-NUST-MS-PhD- CSE (E)-23

MS-11 (CSE)

Submitted to the Department of Computer Engineering in fulfillment of the requirements for the degree of

MASTER OF SCIENCE
In
SOFTWARE ENGINEERING

Thesis Supervisor

Dr. Shoab Ahmed Khan

College of Electrical & Mechanical Engineering

National University of Sciences & Technology

2013

DECLEARATION

I hereby declare that this thesis has been built on my personal efforts under the genuinesupervision of my supervisor Dr. Shoab Ahmed Khan. All the data sources have been referenced and there is no plagiarized data contained in this research. No data of this thesis has been shared as a part of any other research work to be presented in any other institute or university for the fulfillment of degree requirement.

_____

Student Signature

ACKNOWLEDGEMENT

I am grateful to Allah who gave me the strength and courage to accomplish this task in best possible way. With His Sympathy I have been able to complete this work.My sincere and heartfelt thanks to my affectionate and loving parents, family members for their prayers and cooperation in achieving the completion of this task

In completion of this study I am thankful to many people, as I firmly believe, without their help, guidance and most sincere cooperation this accomplishment would not have been possible.

I am grateful to my supervisor **Prof Dr. Shoab Ahmed Khan**whose guidance and assistance made it possible for me to accomplish this task. He has been a source of encouragement and inspiration to me throughout this task completion. His guidance, assistance and unsurpassed knowledge provided me necessary support for carrying out this research.

I appreciatively acknowledge the guidance of my Guidance Committee members**Dr. Farooque Azam, Dr. Muhammad Abbas and Dr. Asia Khaunum**. Their suggestions were very valuable for the development and completion of this research.

I appreciatively acknowledge the coordination of **NS.Madiha Waris** in the development and completion of this research.

DEDICATION

To my parents and teachers

ABSTRACT

Over recent years cloud computing has attained foremost commercial success. Cloud computing minimizes resource wastage risk by reducing the entrance barrier for cloud service providers. By extensive usage of cloud services unstructured data volume is increasing exponentially. Therefore security considerations to save data from hackers are top risks to adapt cloud environment. To avoid illicit use of unstructured data placed on the cloud different encryption techniques and standards have been proposed by the researchers. Data searching becomes a challenging task by adopting the existing searchable encryption techniques. In this paper a new technique Position Based Sentence Search (PBSS) has been proposed. This technique facilitates user with sentence searching on encrypted unstructured data in cloud environment. PBSS provides an efficient ranked sentence search by means of preprocessed indexes. Documents are ranked by two ranking parametersStandard Deviation (SD) and Term Frequency (TF). There is no need of decrypting documents during the search process as in existing techniques. Decryption is done on the retrieval of the documents by user. Cloud server or malicious users have no knowledge ofdocuments content placed on the cloud therefore PBSS achieves security and privacy.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

A comprehensive introduction based on the basic concepts and terms included in the research has been discussed in this chapter.

With the commencement of cloud computing composite data management systems from local sites are transformed to viable public cloud. Data owners are encouraged to outsource the data management systems to public cloud to achieve flexible and commercial benefits [1]. Cloud computing is all about transferring services, applications and data. Also attaining commercial assistances, location transparency, and centralized facilitation are the significant resources in cloud computing. On a shared collective platform like cloud data can be retrieved easily with less revenue and improved assistance [2].

The demand for the enhancement of IT architecture, organization and infrastructure is increasing with the growth of IT services. For fulfilling this demand a massive amount of cost and time is required. These business issues are overwhelmed by the help of cloud computing which provides minimal time and cost to outsource data over it [3]. Cloud computing services are accessible via internet. The services are delivered on the basis of internet for all kind of market users such as government organizations and financial health care markets [4]. Cloud storage has the capability to save a bulk of data for a large number of users. This minimizes the storage capacity problem. To provide different competences multiple isolated applications and services are disseminated over the internet in cloud environment [3].

The services are distributed across the world from the data centers. The convenient and on demand network access to a shared pool of computing resources in minimal amount of time and effort is what cloud computing provides [5], [6].Cloud defines web as a space where computing exists as a service and sharing accessories on the web are data, applications, storage space, processing power and operating systems. They are always ready to be shared whenever required on the web [7].

Among all the policy issues in cloud computing technology the issues which are needed to be highlighted are privacy issues, security overhead, space consumption, reliability and liability

among others. Most important from all of these issues is security issue [6]. Although cloud computing provides a promising platform for internet computing there is a lot of probability that the data placed on the cloud may face security hazards. Outsourcing of enterprise data on the third party cloud platform is the reason of security issues faced on the cloud [8]. The lack of security is actually a barrier for cloud adoption [9]. A Critical example is Amazon system whose failure increased the cloud computing concerns in a huge manner [10].

Whenever a user subscribes to a cloud it becomes easy for a hacker to access the data illegally [11]. In cloud platform the security issues which are faced involve infrastructure security, data security, application service security, virtualization security, regulatory compliance, data location, data segregation, data recovery, investigative support, long term viability and data availability [12], [13], [14].

1.1. SEARCHABLE ENCRYPTION

Sensitive data placed on the cloud may be hacked by the malicious third party therefore security measures are required at both cloud customer and cloud provider level for data confidentiality and data privacy [15].

When sensitive data storage is done on the cloud existence of large number of users can cause cloud security to be affected. Thus for achieving data privacy complex data has to be outsourced on the cloud after encrypting it. Therefore to hide data from hackers and malicious attackers a protected system is needed. Searchable encryption is a technique by which the outsourced data placed on cloud can be kept private. By following this technique potential data storage can be guaranteed to be safe. Searchable encryption will let this data to be impossible to hack when searched. In searchable encryption only encrypted data is kept on the cloud server to be searched when required. Original data is not of any concern if placed on the cloud. Processing of encrypted data placed on cloud server is done without decrypting it. The encrypted data is placed on the cloud in the form of codewords which are unable to be hacked by untrusted parties [16].By using searchable encryption technique the encrypted data will be accessible by trusted authenticated users only. The authenticated users are able to search through this data and retrieve desired results.

1.2. MOTIVATION

Due to the vast use of cloud service the amount of unstructured data on cloud server is increasing exponentially. According to the latest surveys about risks involved in adapting cloud environment cloud security is top ranked. As the cloud computing is fast and efficient more and more organizations and people are shifting their computing resources towards cloud. The vast usage of cloud introduces different level of security threats to the cloud users. To reduce security threats user are using different encryption techniques to store their confidential data to cloud.

The use of encryption technique makes search difficult on the encrypted data. Therefore searchable encryption techniques have been introduced by different researchers. These techniques in literature are not efficient and accurate so there is a need of fast, efficient and accurate search technique which should be secure for cloud environments. The linear searchable technique takes a lot of time in decryption of documents before searching. There is no technique for sentence search in encrypted unstructured data. Considering the security risks and efficiency of search technique PBSS has been proposed for encrypted unstructured data. PBSS fulfills the security requirements and the searching is efficient and accurate.

1.3. SCOPE OF THESIS

The data owners outsource the confidential data to cloud after apply encryption techniques. These encryption techniques make search operations difficult to perform on data. The data user for searching in documents need to decrypt documents which take a lot of time. Different authors have proposed searchable encryption techniques which are not efficient and accurate. These searchable encryption techniques lack the ranking of document for most relevance. For encrypted unstructured data a secure, efficient and accurate searchable encryption technique is always needed. Moreover no sentence search technique has been proposed in literature so far. Therefore there is a need of secure sentence search technique for encrypted unstructured data.

This research solves the problemof sentence search for encrypted unstructured data. The technique proposed and implemented in this research is secure, efficient and accurate. Ranking of documents is calculated on most relevance criteria. A preprocessing of data is required to achieve secure sentence search in efficient way. After preprocessing of data a secure encrypted index is generated. The encrypted index is used for performing the search operations. The

security is achieved as the index and the original documents are encrypted and efficiently searchable. The data on cloud become secure and privacy preserved.

1.4. PROBLEM DEFINITION

Sentence search on encrypted cloud data is not possible till now. Only keywords searching techniques have been discussed in literature. The problem identified in this research is sentence searches on the encrypted cloud data. The traditional search mechanism for unstructured data is linear search in which user has to decrypt documents for searching. This decryption of documents takes a lot of time for searching. The aim of this research is to provide a sentence search mechanism which does not decrypt the documents to perform search operation. To avoid the decryption time a preprocessed index is needed on which search operation can be performed. The preprocessed index should also be secure and does not reveal any information about the document or searched sentence. The search needed to be efficient and accurate. The questions to be answered in this research are as follows:

1. How to create preprocessed index to perform search operation?
2. Will the preprocessed index be secure and encrypted?
3. How encrypted search can be provided in efficient and accurate manner?
4. How sentence based search can be achieved using positions of keywords in the documents?
5. How the accuracy for most relevant document in search can be achieved?
6. How the proposed model achieves data confidentiality i.e. the server is not aware of any of the information about the searched keyword as well as the data contents?

1.5. RESEARCH CONTRIBUTION

The contributions of this research are as follows:

1. Encrypted index of the document is created.
2. Encrypted index is secure for cloud environment and provide secure search.
3. Two level of security have been introduced in PBSS.

4. Sentence search is provided with most relevance criteria.
5. The retrieved searched documents are in the ranked order based on term frequency and standard deviation of keywords positions.
6. Overhead of decryption before searches is reduced.
7. The search using PBSS has been proved secure and efficient.
8. A new technique for sentence search is proposed in this research.
9. PBSS provides efficient and accurate sentence search as compared to the keyword based search.

The proposed technique will reduce the overhead of decryption before searches and will reduce the search time on a considerable scale. The preprocessed index will be encrypted which will not reveal the identity of the documents or the document contents. The decryption is only needed when the relevant documents are returned to the user.

1.6. BASIC CONCEPTS

1.6.1. CLOUD SERVICES

For running applications on internet various cloud services are available. There are a lot of service providers like Google who provide services like Gmail, Google Documents and Google Calendar [17]. Cloud computing services are classified into three types which are Software as a service (SaaS), Platform as a service (PaaS) and Infrastructure as a service (IaaS) [4], [14], [18],[17], [19], [20], [21].

1.6.1.1. Software as a Service

Theservices and applications to the customers are rentedin SaaS i.e. ready to use applications instead of installing those applications on their computers. Web content delivery services, and online word processing tools act as SaaS services. Google and salesforce.com act as SaaS service providers [18], [19], [21].

1.6.1.2. Platform as a Service

Services to the customers are offered as a platform for development environment to run their applications in PaaS. Google App Engine is an example of PaaS by using which Python and Java based applications can be deployed [20], [21].

1.6.1.3. Infrastructure as a Service

Abstracted hardware and operating systems as well as virtual machines are offered over the network in case of IaaS. GoGrid, Flexiscale and Amazon are the examples which offer this service [18], [21].

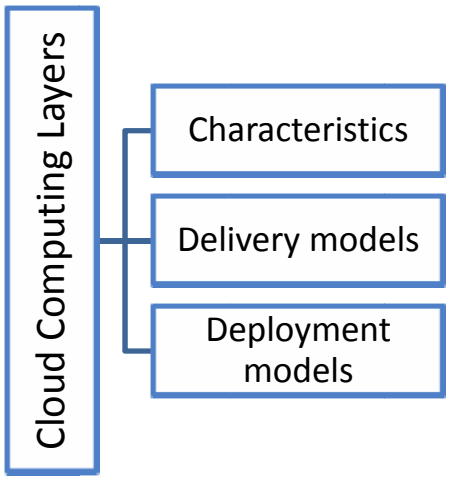

Figure 1.1: Cloud environment architecture derived from [22]

Figure 1.1 shows the basic cloud environment architecture. In the figure characteristics of cloud computing include on demand self-service, broad network access, resource pooling, rapid elasticity and measured service. Delivery models include IaaS, PaaS and IaaS. Deployment models include public, private, community and hybrid models.

1.6.2. CRYPTOGRAPHIC CONCEPTS

Cryptography is used to keep the data in secret form so that it becomes unreadable for the illegal users [23]. Cryptographic algorithms are classified into symmetric and asymmetric algorithms [24]. For symmetric cryptography same set of keys are used for encryption and decryption. Asymmetric cryptography which is also called public key cryptography contains a set of two keys. Th public key is shared publicly and the private key is not shared to anyone except the owner of the key [23], [25].

The proposed technique is based on symmetric cryptography for encrypting the data before sending it to the cloud server. The data can be made secure by any techniques discussed above. A stream cipher has also been used in the proposed technique which is a pseudo random generator and is used to seed on a private key. For hashing purpose a random sized block of input data is transformed into a statistical unique output block of data having a fixed length. Different sizes of output lengths are produced using different hash algorithms. Examples of hash algorithms are SHA-1 (varies from SHA-256 and SHA-512), Tiger and WHIRLPOOL [25].

1.6.3. DOCUMENT SEARCHING TECHNIQUES

Searchable Encryption is a technique of searching on the encrypted data. The existing works on searchable encryption have weakened due to identification of access patterns by hackers [26]. A number of techniques for searching a document have been discussed in [25] which are word sub-match search, exact match search, regular expression search, case insensitivity search, proximity based queries search, natural language search, linear search and pre-processed search.

1.6.4. BLOOM FILTERS

Bloom filters are located in the main memory for giving fast query response and for the adoption of IP traceback [27], [28], [29]. On every member element of the bloom filter a set of hash functions are applied and hence the query speed can be increased [30].

Bloom filter constitutes an array of ‘m’ bits which are initialized to ‘0’. On adding an element in the bit array a number of hash functions are applied to word. The input to all of the hashes is the element to be inserted into the bit array. The output from each hash function is an index to the array bit. Based on the return value of each hash function the bit having that offset i.e. returned output value bit is set to ‘1’. In order to check the presence of certain element in the bit array, it is checked that if any of the bit is set to ‘0’. If it is ‘0’ this element is not stored in the bit array [31].

1.7. THESIS OUTLINE

Chapter 2 presents the literature background related to index based searchable encryption techniques proposed by various authors. In chapter 3 the proposed technique in detail has been given. Chapter 4 provides implementation details of the proposed technique. Chapter 5 presents

the results and analysis obtained by implementing the proposed technique. The thesis has been concluded with a set of contributions and recommended future work in Chapter 6.

Chapter 2: Literature Review

The description of various techniques for searchable encryption by different authors has been given in this chapter. Each paper included in literature review has been studied on the basis of methodology adopted by the author, advantages of and limitations if this technique is followed.

Critical complex data when placed on the cloud server can undergo security complications. Therefore this data when searched by users should be present in encrypted form to vanish the hacking inference. For unstructured data to be outsourced on the cloud it is necessary to preprocess it for ensuring security. Therefore encrypted preprocessed indexes are needed to be created for which literature review in this chapter is helpful. In the literature various techniques for indexing and searching the unstructured data have been given. Also the techniques for performing sentence match searching have been described.

2.1. INDEXING BASED TECHNQIUES FOR UNSTRUCTURED DATA

N. Cao et al. addressed the problem associated with secure ranked multiple keyword search from large amount of data placed across the cloud [32]. The searchable encryption system for multiple keywords has been produced by facilitating the relevance ranking of the search results and hence attaining only the accurate retrieval of the results.

For building multi keyword based searchable mechanism the ‘coordinate matching’ approach has been used. This approach is used to apprehend relevance of documents according to searching query. The method of ‘inner product similarity’ has been used for the evaluation of similarity measure quantitatively. Against two different threat models the two schemes for multi keyword ranked search for encrypted cloud data have been proposed by the authors for achieving privacy requirements. The technique proposed by the authors has been validated through mathematical designs showing low overhead over computation and communication.

According to the attack capabilities two threat models have been considered by the authors which are (1) Known cipher text model and (2) Known background model. Known cipher text model is contains the information about the searchable index and encrypted dataset to be

outsourced by the data owner. Known background model contains additional information about the trapdoor correlations and statistical information of the original dataset.

For both of the threat models privacy preserving schemes have been proposed by the authors to be followed for multi keyword ranked searching.

Privacy preserving scheme in known cipher text model constitutes a four step process to carry on the multiple keyword ranked searching. The algorithm for processing these steps is as follows:

2.1.1. ALGORITHM FOR KNOWN CIPHER TEXT MODEL

S1: Setup: In this step the data owner randomly generates a $(n+2)$ bit vector as S and two $(n+2)\times(n+2)$ invertible matrices$\{M1, M2\}$. The secret key S_k is also generated in the form of a 3-tuple as$\{S, M1, M2\}$.

S2: BuildIndex(F, S_k): A binary data vectorD_i for every document F_i is generated by data owner where each binary bit$D_i[j]$represents whether the corresponding keywordW_j appears in the documentF_i. From Disub index for every plaintext is generated and at the end the sub index for each encrypted document is generated.

S3: Trapdoor $\breve{W}$: $t$keywords to be searched in $\breve{W}$are taken as input from userand one binary vector Q is generated in this step. Each bit $Q[j]$ indicates whether$Wj \in \breve{W}$is true or false and a trapdoor $T_{\breve{W}}$ is generated by applying specific encryption and splitting processes.

S4: Query$(T_{\breve{W}}, k, I)$: By using trapdoor the server calculates similarity scores of every documentF_i. After performing calculation all results are sorted and the top-k ranked id list $F_{\breve{W}}$ is returned by the server.

The keyword privacy can be broken by using scale analysis attack therefore inknown background scheme a more advanced way for multi keyword ranked searching has been proposed.Privacy preserving scheme in known background model constitutes a four step process to carry on the multi keyword ranked searching. The algorithm for performing these steps is as follows:

2.1.2. ALGORITHM FOR KNOWN BACKGROUND MODEL

S1: Setup($1n$): To eliminate fixed value of random variable all vectors are extended to $(n + U + 1)$dimension where $U$is the number of dummy keywords inserted. The data owner randomly generates a $(n + U + 1)$-bit vector as S and two $(n + U + 1) \times (n + U + 1)$invertible matrices$\{M1, M2\}$.

S2: BuildIndex(F, S_k): During extension of dimensions the $(n + j + 1)th$entry in$\overrightarrow{D_i}$where $j \in [1, U]$is set to a random number$\varepsilon(j)$.

S3: Trapdoor$\breve{W}$: The entries in Q are set to 1 on the basis of results obtained by selection of Vout of Udummy keywords.

S4: Query$(T_{\breve{W}}, k, I)$: The similarity proportion is calculated by$r(x_i + \sum \varepsilon(v)i) + t_i$formula by the cloud server where vthdummy keyword is included in Vselected ones.

Traditional Boolean keyword searchable encryption techniques do not support multi keywords ranked search over encrypted cloud data while preserving privacy as proposed in this paper. The limitation of the technique is the linear traversing of the whole index of all the documents for every search request.

N. Cao et al. proposed a searchable encryption system by using secure ranked search [26]. Facility of relevance ranking search with accurate result retrieval has been provided. Statistical measuring approach has been used for building secure searchable ranked index.

Traditional searchable encryption techniquesproposed in [33], [34], [35], [36], [37] allow keyword search through conventional Boolean search. They did not check the relevant documents during search. When these traditional techniques are applied to large data set for searching there is no exact match searchfor the users who do not have pre knowledge of encrypted cloud data. The document retrieval accuracy was a big issue in traditional methods.

The authors defined problem definition in such a way that the data owner wants to outsource data on the cloud server after encryption. Before outsourcing the data to cloud server secure searchable index is needed to be created from keywords taken. When data user wants to search the document containing a specific keyword a trapdoor of the searched keyword is created and

sent to the cloud server. Cloudserver sends those selected documents to the data user in which keyword is matched.

Inverted index structure for solving ranked search problem has been proposed by the authors. The numerical score assignment has been done on the basis of ranking function to find the most relevant documents against a keyword. The ranking function can be customized according to requirement of search. The ranking function finds out the relevance scores of all the matching documents against a keyword to be searched. For calculating the relevance scores of the documents 'TF' and'IDF' rule has been used. 'TF' is the term frequency which is a numerical value to show the number of times a keyword is appearing in a document. 'IDF' is the inverse document frequency which is calculated by dividing the total number of documents in the whole collection by the number of documents which contain the specific keyword to be searched.

The basic techniqueproposedby the authors includes two phases which are (1) Setup Phase and (2) Retrieval Phase.

2.1.3. ALGORITHM FOR SETUP PHASE

The Setup phase is comprised of two algorithms i.e. Key Generation and Build Index.

S1: In key generation step data owner executes the 'Key Generation algorithm' and initialize public and private key parameters.

S2: In build index step the unique words are extracted from the documents to create searchable index.

The data owner encrypts the documents and outsources them to the cloud server along with the indexes containing relevance scores in encrypted form. Order preserving mapping has been used to encrypt the indexes. The owner distributes the secret parameters to a group of authorized usersthrough broadcast encryption.

2.1.4. ALGORITHM FOR RETRIEAVAL PHASE

The retrieval phase is completed by two algorithms (1) Trapdoor generation by the user and (2) Search index by cloud server.

S1: In trapdoor generation it is formed against a specific keyword using Trapdoor Generation algorithm and trapdoor is outsourced to the cloud server.

S2: Search index algorithm searches the index on the cloud server and a list of matched documents ID's along with the order preserved encrypted relevance score is obtained as a result. The matched documents obtained are sent to the user in descending order of their ranks.

The ranked keyword search technique improves system usability by matching documents and delivering in descending ranks. It is useful in the practical deployment of privacy preserving data hosting services in cloud computing. No implementation details have been provided by the authors instead only mathematical proofs have been given to validate the results.

Curtmola et al. proposed an improved searched technique based on the previous work on index based searches on cloud servers [38]. Existing security definitions and shortcomings in those definitions have been highlighted. Two new adaptive and non-adaptive models for encryption have been proposed to prove the new security definitions and to provide multiuser setting.

Curtmola proposed a technique which is based on the combination of a lookup table(T) and array(A). For saving the list of document identifiers in which the word is founda linked list is generated. Linked lists are then encrypted and before encryption a key is associated with each element in the list to encrypt the next element in the list. This techniqueconsists of four steps which are (1)$Keygen$, (2)$BuildIndex$, (3) $Trapdoor$ and (4)$Search$. The algorithm for performing these steps is as follows:

2.1.5. ALGORITHMOR AN IMPROVED SEARCHED TECHNIQUE

S1: Three random keys are generated for$Keygen(k)$based on a security parameter. These keys combine to form a private key$K$.

$$K = (s, y, z) \qquad \textbf{Equation 2-1}$$

S2: For $BuildIndex(k, i)$ document index is built using a given key. The distinct words are collected from the documents. An array (A) is constructed.For each unique worda list of documents is formed which contains that word. Key generation takes place for the

encryption of 0^{th} element in the documents. Following operations are performed for each element in the document.

1. A key is generated for each element to encrypt the next node in the set
2. A tuple is generated which contains document identifier, the key to next tuple and the index of the next element in the list.
3. Before increment of the counter the tuple is inserted into array(A).
4. A lookup table is constructed after array construction It possesses the information related to the value associated with every word in the document set which is as follows:

$$< addressof(A[N_0]), k_{i_0} > \quad \textbf{Equation 2-2}$$

To ensure the security XOR operation is performed between the above value and with the hashed word, with the key y as follows:

$$value = < addressof(A[N_0]), k_{i_0} > \oplus f_y(w_i) \quad \textbf{Equation 2-3}$$

A using a pseudo-random function$F$ is used to store the above values placed in the lookup table to be combined with the key$z$.

$$T[F_z(w_i)] = value \quad \textbf{Equation 2-4}$$

$Trapdoor(w)$is calculated using the following formula:

$$T_w = (F_z(w), F_y(w)) \quad \textbf{Equation 2-5}$$

S3: For $Search(I, Tw)$this function first picks the document index $I$and trapdoor valueT_w. The first element of this trapdoor represents the key to the lookup table which is retrieved as:

$$value = T[T_w{\downarrow_1}] \quad \textbf{Equation 2-6}$$

XOR operation is performed on second part of the trapdoor and the above value to get the tuple having following data.

$$< addressof(A[N_0]), k_{i_0} > = value \oplus T[T_w\downarrow_2] \quad \textbf{Equation 2-7}$$

The above formula is useful for server to retrieve the information to decrypt all the elements in the linked list whose head is stored in $A[N_0]$. The data is useful to construct a list of document identifiers which contain the word$w$. The list is then returned to the client.

The technique proposed by Curtmola et al. is faster as compared to previous ones. For a large datasetconstant search time algorithm has been provided. For the implementation of this technique the array(A) and trapdoorT_w are needed to be updated whenever the document is added or removed which is a drawback.

Park et al. proposed two approaches for enabling searchable encryption on cloud server [39]. The approaches described are(1) efficiency and (2) searching in cloud data center. Two techniques of efficiency and group search for practical keyword index search–I and search–II have been proposed by these authors. The analysis of group search security and keyword index search security has been done. Similarly the efficient performance of proposed encrypted database has been calculated. Safe and secure search has been assured by analyzing keyword index search security without re-encryption of all the documents.

The steps involved in conducting the efficient searched based technique proposed by the authors are$SysPara$,$KeyGen$,$IndGen$,$DocEnc$, $TrapGen$, $Retrieval$ and Dec. The$SysPara(1k)$ takes a security parameter $k$and outputs a system parameter 1. 1 indicates the elements to set out the encrypted database system.

2.1.6. ALGORITHM FOR CONDUCTING EFFICIENT SEARCH

- **S1:** In$KeyGen(1)$functkion it takes 1 as an input and generates user's group session key set$\{gk\}$, index generation key set$\{ik\}$ and document encryption key set$\{dk\}$.
- **S2:** The $IndGen(ik\ , W)$function it takes input index generation key set $ik$and a keyword set$W$. Output is index list table.
- **S3:** The $DocEnc(dk, D)$ function encryptsdocument using encryption key $dk$and a document$D$.

S4: The $TrapGen(w, ik)$function takes a keyword $w$and index generation keyik. Trapdoor T_w is obtained by the encryption of keyword $w$and index generation key ik.

S5: The $Retrival(T_w)$function takes trapdoor T_w as an input. If trapdoor is present in the index list then the output is obtained in the form of encrypted documents that map the identifiers of the matching values in the index list table.

S6: In $Dec(E(D), dk$ function the document is decrypted by document encryption key $dk$and encrypted document$E(D)$.

Indexes are not secure in the above described technique and any third party might be able to deduce contents of data from the index. The common keywords from any two documents can even be traced by adversary.

Sun-Ho Lee and Im-Yeong Lee proposed a technique for searchable encryption by which user can share data with others [40]. The search has been made possible by securely generating encrypted searchable index and re-encrypting it. The technique proposed for getting the desired results constitutes key generation and the keys for re-encryption are generated if the user wants to share own data with other users. The user generates the trapdoor of keywords witha secret key for performing search. The decryption of data is performed only by the authorized users.

Data confidentiality, quick search speed, and efficiency in communication volume due to only one round check for communication process have been achieved by using this technique. On the other hand there is no technical and experimental proof for the validation for this technique.

Jin Li et al. formalized the problem of operational fuzzy keyword search over encrypted cloud data with the preservation of keyword privacy [41]. The authors proposed a technique which solves the problem of exact match search or closest match for the input keywords. Two advanced techniques for fuzzy keyword set generation have been developed. A symbol based trie-traverse searching technique has also been proposed. Efficiency of the proposed system has been analyzed through experimentation. Each document placed on the server has been indexed for distinct set of keywords. Fuzzy keyword sets have been constructed by using two advanced techniques i.e. wildcard-based fuzzy set construction and gram-based fuzzy set construction. On the basis of algorithms defined in the paper statistical evaluation of results have been shown by the authors.

Tamboli et al. proposed a technique of fuzzy keyword search for retrieving exactly matched documents [42]. It provides the ability to search the closest possible matching document if the exact match does not exist. For quantification of keyword similarity the idea of edit distance has been used. This system provides the facility to encrypt the text files, image files, and video files. Security has been ensured by the encryption of the data to be searched by the user. Various algorithms have been proposed for the creation of fuzzy set. The basic technique follows the process such that at first the keyword is taken as an input and in a database fuzzy set is maintained. The documents which are matched are returned by the database. A private key id is used for the decryption of searched document. The technique provides an efficient search and concept of exact search is also introduced. Experimental results are shown which do not truly demonstrate the analytical scenario of the experiments.

N. Caoet al. proposed feature based index to solve the problem of privacy preserving queries over graph structured data in cloud computing [43]. Security has also been achieved by the authors for fulfilling strict privacy requirements. The principle of filtering and verification has been used to validate the proposed technique. On the basis of known background threat model a secure inner product computation technique has been used for achieving various privacy requirements.

The algorithm for proposed framework is defined below:

2.1.7. ALGORITHM FOR SECURE INNER PRODUCT COMPUTATION

S1: FSCon: Graph dataset is taken by this function and a feature set is built on the basis of this dataset.

S2: KeyGen: This function takes a secret key as an input a symmetric key is obtained as a result.

S3: BuildIndex: This function takes graph data set and symmetric key as an input and a searchable index is received as and output.

S4: TDGen: In this function graph data set and symmetric keys are taken as input and a trapdoor is obtained as output.

S5: Query: Trapdoor and searchable index are taken as input values to retrieve the data set to be searched from the graph.

The feature set building, key generation and preprocessed index generation takes place at data owner level whereas query algorithm is run by the cloud server. Privacy requirements for index, trapdoor and feature set have also been fulfilled by using this technique. Mathematical proofs in the form of algorithms have been given to prove the validity of the proposed technique results.

Narayanet al. proposed a technique to provide privacy preserving electronic health record management system (EHR) [44]. EHR has been constructed by the combination of both attribute based cryptography and public key encryption with keyword search. The purpose of the research was to provide a dynamic, flexible, privacy preserved and a scalable system which guarantees the security achievement. The main aim was to keep the patients' files confidential and to assure no data leakage on the cloud server. The actual data to be outsourced on the cloud is encrypted in the form of searchable indexes.

The algorithm proposed for constructing EHR is as follows:

2.1.8. ALGORITHM FOR EHR CONSTRUCTION

S1: Store File: A new file and a locator tag are stored by using this function where each file is stored in the form of encrypted indexes.

S2: Set Access: An access to the encrypted files is granted to only the privileged users by using this function.

S3: Revoke Access: The access to encrypted files is revoked by the healthcare provider to some users whom the provider does not allow to access the data.

S4: Delegate: A private key for health care provider is built with the help of this function.

S5: Keyword Search: When health care provider has a secret key then the search can be easily supported. The search is provided on the encrypted indexed files such that the cloud server gets no knowledge about the data placed over it.

The proposed technique has ensured data confidentiality and data privacy issues. This technique has higher computational costs due to re encryption of records while updating the data.

Ming Li et al. addressed the problem associated with private keyword searches on encrypted data of personal health records [45]. This problem became very challenging when a keyword was

being searched by multiple users. Authors have proposed a scalable and fine grained authorization framework for searching over encrypted personal health records. In public domain this framework provides high level scalability for personal health record applications. The proposed technique supports document privacy, query privacy, multiple keywords searching, and revocation of search capabilities. Two types of keyword indexes have been generated. All the files of personal health record of owner are saved as one index. Secondly individual index for each single file inside the personal health record is saved.

The algorithm proposed for authorized private keyword search is as follows:

2.1.9. ALGORITHM FOR AUTHORIZED PRIVATE KEYWORD SEARCH

S1: Setup: This function takes an input security parameter and outputs a master private key and a public parameter.

S2: GenIndex: This function encrypts the index and outputs a searchable index.

S3: GenCap: This function outputs a search trapdoor to be matched with the searched keyword.

S4: Search: This function returns the corresponding file if the searched keyword matches with the trapdoor.

With the help of theoretical evaluation it has been shown that the proposed technique is helpful in practical usage.

Treesa Maria Vincent Mrs.J.Sakunthala discussed various concepts for searchable encryption in cloud environment [46]. Concepts of encrypted storage, relevance score, one-to-many order preserving mapping, privacy enabled data searching technique and ranked search concepts have been defined. Authors aimed to propose a framework which rank order the documents retrieved as a result of a searched keyword. The resultant framework follows the process of searching in such a way that initially to search a document a user inputs a keyword. A trapdoor of a keyword is sent to the cloud server. When the cloud server gets the search request it searches the index and corresponding set of documents are sent to the user. In order to have secured ranking and improved retrieval of documents frequency based scoring technique has been proposed by the authors.

Rajeev Bedi et al. discussed two different frameworks for privacy preserving cloud storage on the basis of their feasibility, security and running overhead [47]. Framework I constitute data organization design, key management, user's access rights, data operations and symmetric encryption for data privacy preservation over cloud server. Framework II constitute key management, symmetric and asymmetric encryption to secure sensitive data of users and bloom filters for retrieving cipher text.

2.1.10. ALGORITHM FOR PRIVACY PRESERVING FRAMEWORKS

Algorithm discussed for framework I is as follows:

S1: The data sent by the owner is encrypted with the help of a key. Indexes are generated.

S2: User requests to retrieve some keyword from encrypted indexes.

S3: The identity of user is verified and after verification the searched keyword is checked if it matches or not.

S4: The authenticated users are given certificates and the retrieved document containing the searched keyword is sent to certified user only.

S5: The document is decrypted by the user to obtain the original data.

Algorithm discussed for framework II is as follows:

S1: By symmetric encryption with the help of a key each of the documents is encrypted. By asymmetric encryption with the help of a key pair the keywords present in each document are encrypted. Bloom filter is produced and encrypted indexed keywords are placed on the cloud server.

S2: User is given authorization from owner and certificate is assigned. When user searches for a keyword his authorization is checked and search is performed for certified users only.

S3: The requested keywords are checked for similarity on the cloud server. If match occurs the specific document containing the keyword is returned to the certified user.

The analysis of both the frameworks shows that framework II is better as compared to framework I w.r.t key management, owner privacy preservation, and computing overhead.

Ming Li et al.identified the challenges and problemsagainst the flexible, efficient and privacy assured outsourced cloud services [48]. Authors have focused on two kinds of search functionalities i.e. ranked keyword search and search over structured data. Existing techniques for carrying out search over the outsourced data, their advantages and limitations have been studied by the authors. On the basis of these existing techniques a framework for privacy assured search in cloud has been given.

The algorithm for the framework proposed for privacy assured keyword search is as follows:

2.1.11. ALGORITHM FOR PRIVACY ASSURED KEWYORD SEARCH

S1: Setup:A security parameter is taken as an input to thisfunction and a secret key is obtained as an output. This secret key is used later by the authorized users and owner.

S2: IndexGen: From the dataset for each file index is generated and is out sourced to the cloud server. The data files are also encrypted.

S3: TrapdoorGen: A trapdoor is generated as a result of this function with the help of a secret key.

S4: Search: Similarity match is done in this function and a keyword is searched. The searched keyword is matched with the trapdoor produced in step S3. If matched it is returned to the user.

The problems associated with privacy assured keyword search functionalities have also been studied by the authors for cloud data services.

Swaminathan et al. introduced a framework for confidentiality of ranked ordered large set of documents [49]. The framework provides protection of searched keyword, whole set of documents and data center security. Relevance scoring method and order preserving technique has been used to protect data and provide efficient search. The proposed technique processes in such a way that at first pre-processing occurs for all the stored documents. The encrypted indexes are formed and a secure term frequency table is built so that accurate information retrieval is

attained. When a user wants to search a keyword form the encrypted indexes it is hashed using the same key used during indexing. Using a search function the match is done to retrieve the documents containing searched keyword. Results have been obtained to show the efficiency of the proposed technique with the help of mathematical evaluation.

2.2. ANALYSIS

For highlighting the efficiency analysis of the above discussed techniques a brief comparison has been given below.

N. Cao et al. [32] proposed a searchable encryption technique for multiple keyword searches from large amount of data placed on the cloud using secure ranking. For this purpose the concept of coordinate matching and inner product similarity has been used. Linear traversing of the whole index for every search request is an overhead for this technique.

N. Cao et al. [26] addressed the problem associated with keyword search on the basis of secure ranking. No implementation details have been provided by the author. Only mathematical proofs have been provided for finding results.

Curtmola et al. [38] reviewed the index based searches in existed techniques and proposed an improved technique based on traditional ones.No results and analysis have been discussed in this research. Only the theoretical details have been given.

Park et al. [39] described efficiency analysis and searching on cloud servers. This research does not support the data security at server level. Any third party might be able to deduce the contents of data from the index placed over the cloud. Therefore this technique is not secure enough for cloud servers.

Sun-Ho Lee and Im-Yeong Lee [40] proposed a searchable encryption technique by index generation. No experimental evaluation of the results has been given. Only the theoretical evaluation has been done.

Jin Li et al. formalized the problem of operational fuzzy keyword search over encrypted cloud data with the preservation of keyword privacy [41]. No string searching has been proposed. Only algorithm for multiple keywords has been given.

Tamboli et al. proposed a technique of fuzzy keyword search for retrieving exactly matched documents [42].Experimental results are shown which do not truly demonstrate the analytical scenario of the experiments.

N. Caoet al. proposed feature based index to solve the problem of privacy preserving queries over graph structured data in cloud computing [43]. Mathematical proofs in the form of algorithms have been given to prove the validity of the proposed technique results. The technique lacks the implementation details.

Narayanet al. proposed a technique to provide privacy preserving electronic health record management system (EHR) [44]. The proposed technique has ensured data confidentiality and data privacy issues. This technique has higher computational costs due to re encryption of records while updating the data.

Ming Li et al. [45] addressed the problem associated with private keyword searches on encrypted data of personal health records. A scalable and fine grained authorization framework for searching over encrypted personal health record system has been proposed. Theoretical evaluation has been done to validate the proposed technique.

Treesa Maria Vincent Mrs.J.Sakunthala [46] discussed various concepts for searchable encryption in cloud environment. Authors proposed a framework which rank order the documents retrieved as a result of a searched keyword. Frequency based scoring has been done.

Rajeev Bedi et al. [47] discussed two different frameworks for privacy preserving cloud storage on the basis of their feasibility, security and running overhead. Performance of both frameworks has been checked using statistical measures.

Ming Li et al. [48] identified the challenges and problems against the flexible, efficient and privacy assured outsourced cloud services. Ranked keyword search and search over structured data has been focused by the authors. On the basis of these existing techniques a framework for privacy assured search in cloud has been given. No experimental or statistical evaluation has been done to validate the proposed framework.

Swaminathan et al. [49] introduced a framework for confidentiality of ranked ordered large set of documents. The framework provides protection of searched keyword, whole set of documents and data center security. Mathematical evaluation has been done to validate this framework.

The techniques discussed by various authors in literature have been summarized in Table 2.1. This table comprises the application context of each domain, the discussed technique by each author, and the validation approach for the corresponding technique.

Table 2.1: Summary of reviewed techniques

Author	Focus and Context	Technique	Validation
N. Cao et al. [32]	**1.** Multiple keyword searching having secured ranked search using coordinate matching and inner product similarity **2.** Achieving privacy and efficiency	Two multi-keyword ranked search techniques based on indexing	Mathematicalreasoning
N. Cao et al. [26]	**1.** Single Keyword search based on secure ranking **2.** Relevance scoring for ranked providing ranked search	Indexed search	Mathematical reasoning
Curtmola et al. [38]	**1.** Keyword search by the proposal of a new technique based on the previous studies **2.** Security definitions approval using two new adaptive and non-adaptive models of encryption	Indexing using Lookup Table and Linked lists	Theoretical reasoning
Park et al. [39]	**1.** Keyword search and Efficiency analysis **2.** No focus on security	Indexed search	Mathematical reasoning
Sun-Ho Lee and Im-Yeong Lee [40]	**1.** Searchable encryption technique for keyword search **2.** Efficiency analysis	Indexed search	Theoretical reasoning
Jin Li et al. [41]	**1.** Operational fuzzy keyword search **2.** Keyword privacy preservation	Indexed search	Statistical evaluation on the basis of Mathematical reasoning

Tamboli et al. [42]	**1.** Fuzzy keyword searching **2.** Retrieval of closest possible matched document if exact match does not exist	Linear search	Experimental evaluation
N. Cao et al. [43]	**1.** Feature based index technique for handling queries over graph structured data **2.** Security achievement by using inner product computation technique	Indexed search	Experimental evaluation based on mathematical reasoning
Narayan et al. [44]	**1.** Provision of privacy preserved electronic health record system **2.** Attribute based cryptography and public key encryption used for keyword search	Indexed search	Comparative analysis
Ming Li et al. [45]	**1.** Private keyword search on encrypted data of personal health records **2.** Scalable and fine grained authorization framework **3.** Privacy preservation	Indexed search	Theoretical reasoning
Treesa Maria Vincent Mrs.J.Sakunthala [46]	**1.** Discussion of various searchable encryption concepts **2.** Frequency based scoring **3.** Efficiency analysis	Indexed search	Analysis not done
Rajeev Bedi et al. [47]	**1.** Two frameworks for privacy preserving cloud storage **2.** Key management, user's access rights, data operations and symmetric encryption for data privacy preservation over cloud server	Indexed search	Statistical analysis
Ming Li et al. [48]	**1.** Ranked keyword search **2.** Privacy assured search in cloud based on existing techniques **3.** Flexible and efficient search mechanism	Indexed search	Analysis not done
Swaminathan et al. [49]	**1.** A confidential framework for ranked ordered search **2.** Protection of keyword, data center and whole set of documents	Indexed search	Mathematical reasoning

Chapter 3: PBSS Technique

This chapter focuses on the model proposed against the problem definition. The features and limitations of the proposed system have been defined. The algorithmsof the proposed technique have been discussed.

3.1. PROBLEM STATEMENT

Sentence search on encrypted cloud data is not possible till now. Only keywords searching techniques have been discussed in literature. The problem identified in this research is sentence searches on the encrypted cloud data. The traditional search mechanism for unstructured data is linear search in which user has to decrypt documents for searching. This decryption of documents takes a lot of time for searching. The aim of this research is to provide a sentence search mechanism which does not decrypt the documents to perform search operation. To avoid the decryption time a preprocessed index is needed on which search operation can be performed. The preprocessed index should also be secure and does not reveal any information about the document or searched sentence. The search needed to be efficient and accurate. The questions to be answered in this research are as follows:

1. How to create preprocessed index to perform search operation?
2. Will the preprocessed index be secure and encrypted?
3. How encrypted search can be provided in efficient and accurate manner?
4. How sentence based search can be achieved using positions of keywords in the documents?
5. How the accuracy for most relevant document in search can be achieved?
6. How the proposed model achieves data confidentiality i.e. the server is not aware of any of the information about the searched keyword as well as the data contents?

3.2. PROBLEM SOLUTION

For the problem discussed in section 3.1 PBSS technique is proposed for sentence searching. PBSS is fast accurate and efficient for sentence searches as it provide more accurate result in short period of time. PBSS preprocesses the document and generates the document index for searching. The preprocessed index is fast and accurate for searching. Documents are returned on most relevance which is found by mean SD and TF discussed in section 3.4.2.9. PBSS technique comprise of two main steps Indexing which is preprocessing of documents and Searching to find most relevant document according to the searched sentence. Two security levels have been implemented in PBSS trapdoors and codewords for both Indexing and Searching. Bloom filters have been used to make the codewords more secure and confidential. Different hash algorithms are used to obtain the required security level which can be customized. The data owner can use more secure and customized hash algorithms for PBSS. The documents are made secure by using AES standard 256 bit encryption technique.

Cloud server is unaware of the document content as documents are in encrypted form. During searching the cloud server cannot learn any information about the searched sentences or the encrypted index. The searched sentence keywords are sent to cloud after applying two security parameter discussed earlier so cloud server or malicious users are unaware of what is searched.

3.2.1. BLOOM FILTER FOR SECURITY ACHIEVEMENT

Bloom filter is a data structure used for fast membership test. Bloom filters are used in PBSS for achieving second security level Generate Codeword discussed in section 3.4.1.4.

Bloom filter is an array where different element can be stored on different bit positions. A fixed length array is initialized to ‘0’ all bit positions. Fixed numbers of hash function are applied on any element to get bit positions for the element to be stored in the bloom filter.Example of bloom filter with three hash function is shown in Figure 3.1.

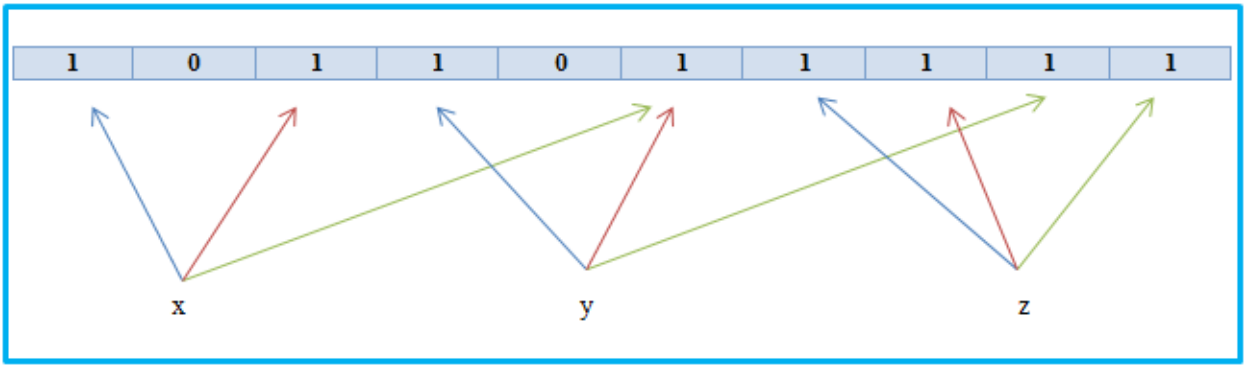

Figure 3.1: Bloom filter using three hash functions

In Figure 3.1x, y and z are three words added to the bloom filter using three hash functions each line shows one hash function and these tree line combine to represent a complete word in the bloom filter. The bit position indicated by hash function is made '1' if its '0'. Two or more hash functions can point to same bit position and this problem is called false positive rate. The false positive rate can be minimized by the use of more than three hash function to represent a word and the length of array can also be increased accordingly.

To check for the membership of the element or word in the bloom filter three hash function are applied as in above example. If all the bit position indicated by hashed values are '1' in the bloom filter array it means the element is present. If any of the bits is '0' the element is not a member of this bloom filter.

The security level in PBSS is achieved by bloom filter in such a way that trapdoors are added to the bloom filter using five hash algorithms. The bits positions obtained are concatenated which are called codewords for that trapdoors.

The space needed to store a bloom filter is small as compared to the amount of data belonging to the set being tested. The time required to check the membership of an element is not dependent on the number of elements contained in the set. False negatives are not possible but false positives are possible to occur and their frequency can be controlled [31].

3.3. PBSS FRAMEWORK

The basic techniquefor proposed model constitutes two phases which are:

S1: Indexing

S2: Searching

Three entities are involved in complete operations:

1. Data Owner
2. Data User
3. Cloud Server

For all documents of the data owner indexing is performed. When data owner uploads the document to the server its encrypted index is created and the original document after encryption (asymmetric or symmetric) is uploaded to the cloud. Now the cloud has encrypted indexes of documents and the encrypted documents. During search of sentences two possible scenarios can be used to achieve security and privacy of search sentence. The sentence which data user wants to search can be sent directly to cloud after converting to codewords. The codewords are generated by the same steps as in indexing. For this case the data owner has to share the codeword generation steps with data user and the secret keys as well. In second case the data users can send the searched sentence to the data owner who will convert them to codewords and will send to cloud server to perform search using PBSS.

When sentence in form of codewords is sent to the cloud the cloud server check the documents indexes and select those documents which contain the codewords. After the data selection from indexes PBSS algorithm is performed and matches the searched sentence in the selected documents data. On the basis of standard deviation (SD) and term frequency (TF) the documents are ranked. The ranked documents are then decrypted and original documents are returned to data user on request.

If extracted keywords from sentence are present at consecutive positions in the document then it is returned. If extracted keywords are present at different positions then SD of all positions is calculated and TF of the extracted keywords is calculated in the documents. The documents are ranked by given formula:

$$Rank = 0.4 \times SD - 0.6 \times F$$ **Equation 3-1**

Figure 3.2 depicts the basic framework of the proposed PBSS technique. The figure shows the process flow of indexing and searching performed at data owner and data user end respectively. The data confidentiality and privacy is achieved in PBSS as the cloud is unaware of the original documents as they are in encrypted form and the document index is also encrypted. The search process is also achieves confidentiality and privacy as the index or the searched sentence do no reveal any identity of data. The searched sentence is converted to useful keywords which are made secure by following security parameters and converted to unpredictable codewords.

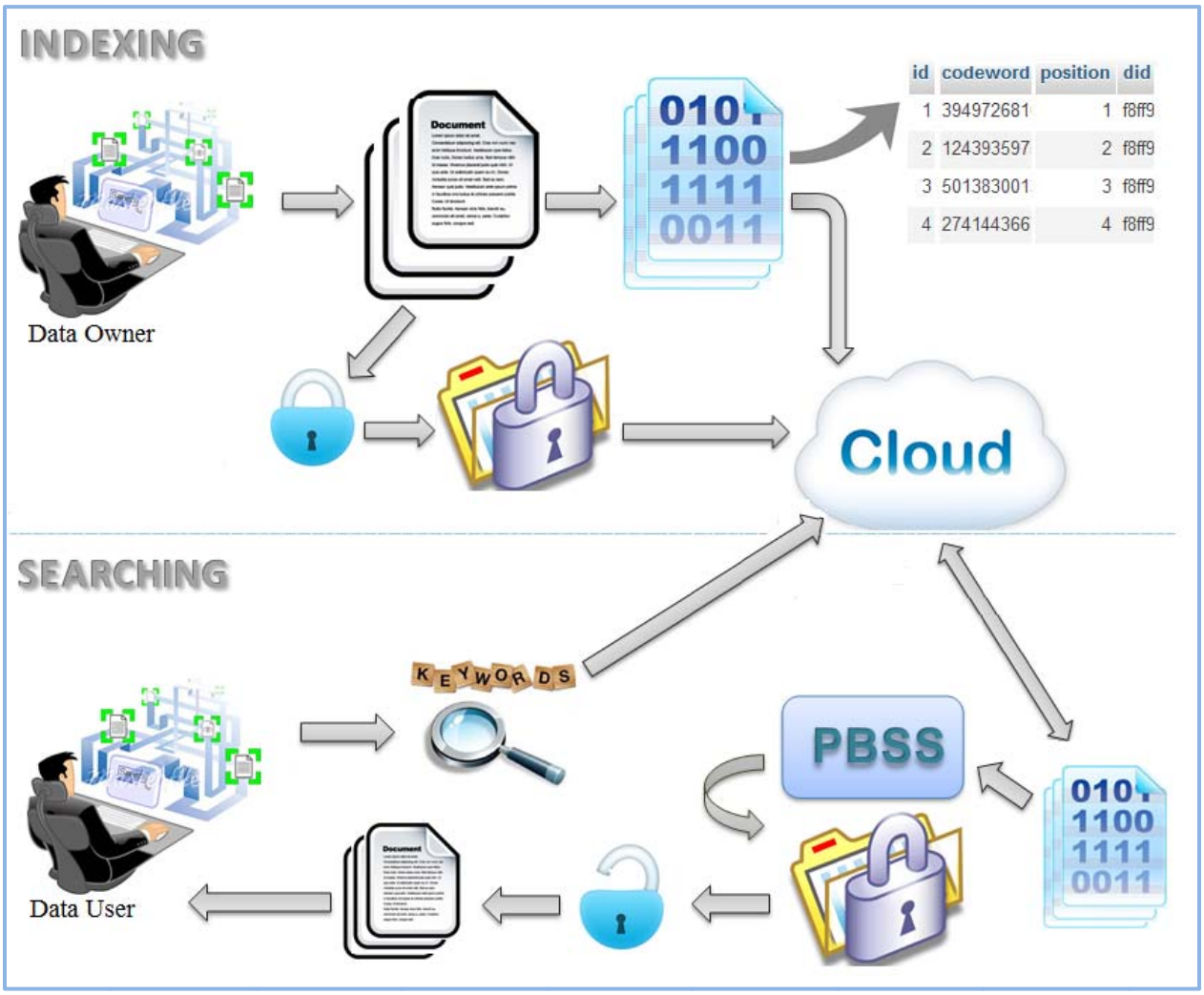

Figure 3.2:PBSS Framework

3.4. PBSS TECHNIQUE

In PBSS index of the document is generated. The index is generated by converting all keywords form the document to codewords and then storing the codewords to secure database for searching. All distinct keywords after removal of stop words are represented by codewords and codewords are generated by following two security parameters explained in the indexing process. The proposed scheme consists of two phases which are indexing and searching.

3.4.1. INDEXING

The steps performed in Indexingare as follows:

3.4.1.1. Input Document

All documents from data owner which need to be outsourced to the cloud are scanned. All distinct keywords form the document are collected and stop words are removed from this collection. Stop words are those words which help to complete the sentence but itself has no meaning. Examples of stop words are‘the’, ‘a’, ‘an’, ‘be’, ‘either’, ‘my’, ‘not’, ‘of’. Only meaningful key words are indexed during indexing which will reduce the space overhead of indexes and faster indexing is achieved.

3.4.1.2. Generate Master Key and Split

A Master Key is generated which is used to get eight split keys. The eight split keys are used in trapdoor generation step to generate secure and unpredictable trapdoors. Master key is generated by taking the hash of user selected password i.e. $SHA512$of the security parameter.

$$M_k = SHA512(password) \quad \textbf{Equation 3-2}$$

$M_k$is then split into a set of eight keys. These eight splitkeys are used in trapdoor generation step to secure the trapdoors.

$$M_k: k_1, k_2, k_3, k_3, k_4, k_5, k_6, k_7, k_8 \quad \textbf{Equation 3-3}$$

3.4.1.3. Generate Trapdoor

Trapdoor generation is the first security parameter to secure the keywords from document. All the keywordsfrom document are concatenated with all eight key and hash is applied on each concatenated value. The eight hashed values are then concatenated with coma inside to get a trapdoor. The hash function used in this step is of data owner choice. Any hash algorithm can be

used by data owner for generating trapdoors. The trapdoor obtained from above technique is represented in formula as shown below.

$$Trapdoor: f(k_1 + word), f(k_2 + word) \dots f(k_8 + word) \quad \textbf{Equation 3-4}$$

3.4.1.4. Generate Codeword

The generate codewords step is second security parameter which enhances the security of keyword's trapdoor generated in the above step. The trapdoor is sent to bloom filter to get codewords. Codewords are bit positions obtained for each trapdoor when crc32 hash algorithm is applied on it. Depending on the document length the false rate can increase or decrease. If length of file is large then large false positive rate is expected. To avoid this problem the number of bit positions for a trapdoor can be varied. Currently we are using 5 bit positions to storea trapdoor. It means one trapdoor is stored on five bit positions. For this purpose five time crc32 hash algorithm is applied on the trapdoor with five uniform random numbers. Using five uniform random numbers enables the bloom filter to always generate same bit position for one trapdoor. The codewords generation formula is as follows:

$$codeword = 5\ hash\ algos\ (Trapdoor) => 5\ bit\ postions \quad \textbf{Equation 3-5}$$

3.4.1.5. Find Positions

Find positions of the keywords extracted from the document. Position of the keywords is the absolute place where keyword is present in the document. These positions will be used during sentence search in PBSS. All keywords have been converted to codewords and codeword will have the same position in document which its original keyword has.

3.4.1.6. Upload to Cloud

Upload encrypted document index, encrypted document and documents id on to the cloud server. As all the data is in encrypted form the cloud is unable to identify any document content or indexes. This makes the index and documents secure and private. The uploadingof sensitive and confidential data to cloud is secure and privacy preserved.

3.4.2. SEARCHING

The steps performed during search are as follows:

3.4.2.1. Input Sentence

When a user searches for a sentence stop words are removed from it. The important keywords are extracted from the input sentence. The stop words removal produces less number of combinations in Find Combinations step which will be explained further in its section. As stop words are also removed in indexing step which makes the indexing faster and search as well.

3.4.2.2. Generate Master Key and Split

Mater key is generated and split in to eight distinct keys. This step is same as performed during indexing step. The formula for generating master key and then splitting it into eight keys is shown below:

$$M_k = SHA512(password) \quad \textbf{Equation 3-6}$$

Formula for splitting master key to eight split keys:

$$M_k : k_1, k_2, k_3, k_3, k_4, k_5, k_6, k_7, k_8 \quad \textbf{Equation 3-7}$$

3.4.2.3. Generate Trapdoor

For all the keywords of the input sentence a trapdoor is generated. The generate trapdoor step is same as explained in indexing. When a user searches a sentence stop words are removed from it. All extracted keywords are concatenated with each of the eight keys and hash is applied on them such that the following operation is performed:

$$Trapdoor: f(k_1 + keyword), f(k_2 + keyword) \ldots f(k_8 + keyword) \quad \textbf{Equation 3-8}$$

The eight keys used in above equation to be concatenated with the keyword are same which were produced during Generate Master Key and Split. These keys remain the same throughout the process.

3.4.2.4. Generate Codeword

The trapdoors generated in the above steps are converted to codewords. The extracted keywords from the sentence are converted to codewords which will be matched in the document indexesduring search using PBSS technique. It is done in such a way that the trapdoor obtained

for extracted keyword's trapdoor in above step is sent to the bloom filter and five hashes are applied on it to get the bit positions. These bit positions are then combined to form a codeword.

$$codeword = 5\ hash\ algos\ (Trapdoor\ for\ keyword) => 5\ bit\ postions$$ **Equation 3-9**

3.4.2.5. Select Query

Find documents from encrypted indexes which contain search sentence keywords. The extracted keyword's codewords from the searched sentence are checked in the encrypted indexes of documents. The documents which contain the extracted keyword's codewords are returned with codewords and position of matched codewords.

3.4.2.6. Arrange Selection

The resulted documents form Select Query step are now arranged in such a way the all the codewords and their position are placed under the document which contains them. The codewords and their positions are properly placed under the document from which they have been extracted and stored in index during indexing.

3.4.2.7. Find Combinations

Arranged Selection is taken as input and combinations of codewords are found which can make a possible sentence match. All possible combinationsare considered which can be a part of the sentence for example if user searches 'Searchable encryption techniques in literature' then possible combination may be as follows:

1. Searchable encryption techniques in literature
2. encryption techniques in literature
3. techniques in literature

These can be the possible combinations for sentence match. PBSS considers all possible combination of sentence during sentence match.

3.4.2.8. Calculate Mean SD and TF

Calculate SD of all possible combinations found in Find Combinations Step and take mean of these SD values. Find TF of all searched sentence keywords matched in documents.

3.4.2.9. Rank and Return Document List

Documents are arranged in ascending order of their ranks. The ranks are calculated by formula given below. The document with the minimum rank value is most relevant. The document with

minimum SD means that spread of search sentence keyword is less in the document. If spread is less than possibility of sentence match is highest. If TF is high the document is most relevant. Combining these two parameter documents are ranked by given formula:

$$Rank = 0.4 \times SD - 0.6 \times TF$$ **Equation 3-10**

After calculation of ranks the ranked document list is returned to the user. Documents are only decrypted on data user request.

3.4.3. GRAPHICAL REPRESENTATION OF PBSS

In this section the step involved in Indexing and Searching steps are shown in detail. The flow diagram explains the steps involved in Indexing and Searching.

3.4.3.1. Graphical representation for Indexing

The flow diagram of the Indexing is shown in Figure 3.3 on next page.The steps briefly explain the process involved in indexing. First of all a document is taken as input and keywords are extracted from it after removing stop words. Trapdoors are generated for all keywords by generate trapdoor step. These trapdoors are sent to bloom filter to get codewords.

The bloom filter applies hash on trapdoors and returns five distinct bit positions for the trapdoor. The five distinct bit positions combine to give a codewords. Finally in find position step the codewords are given positions. The positions assigned to codewords are same as their keywords have in the document.

3.4.3.2. Graphical representation for Searching

The graphical model of Searching is shown in Figure 3.4. The flow of searching is shown in detail in Figure 3.4.For search after the input sentence the processes up to Generate Codeword are same as in indexing. These steps ensure the security levels have been achieved so that sentence or keywords searched should not be predicted by cloud server or malicious users.

After Generate Codeword related document are selected form the document indexes in Select Query step. The selected documents are then arranged in Arrange Selection step. All possible combinations of the arranged data are found in Find Combinations Step.

The mean SD of all combinations is calculated and TF of searched sentence keywords in the relevant documents is calculated in Calculate Mean SD and TF step. The final ranks of the

documents are calculated in Rank and Return Document List step and the ranked document list is returned.

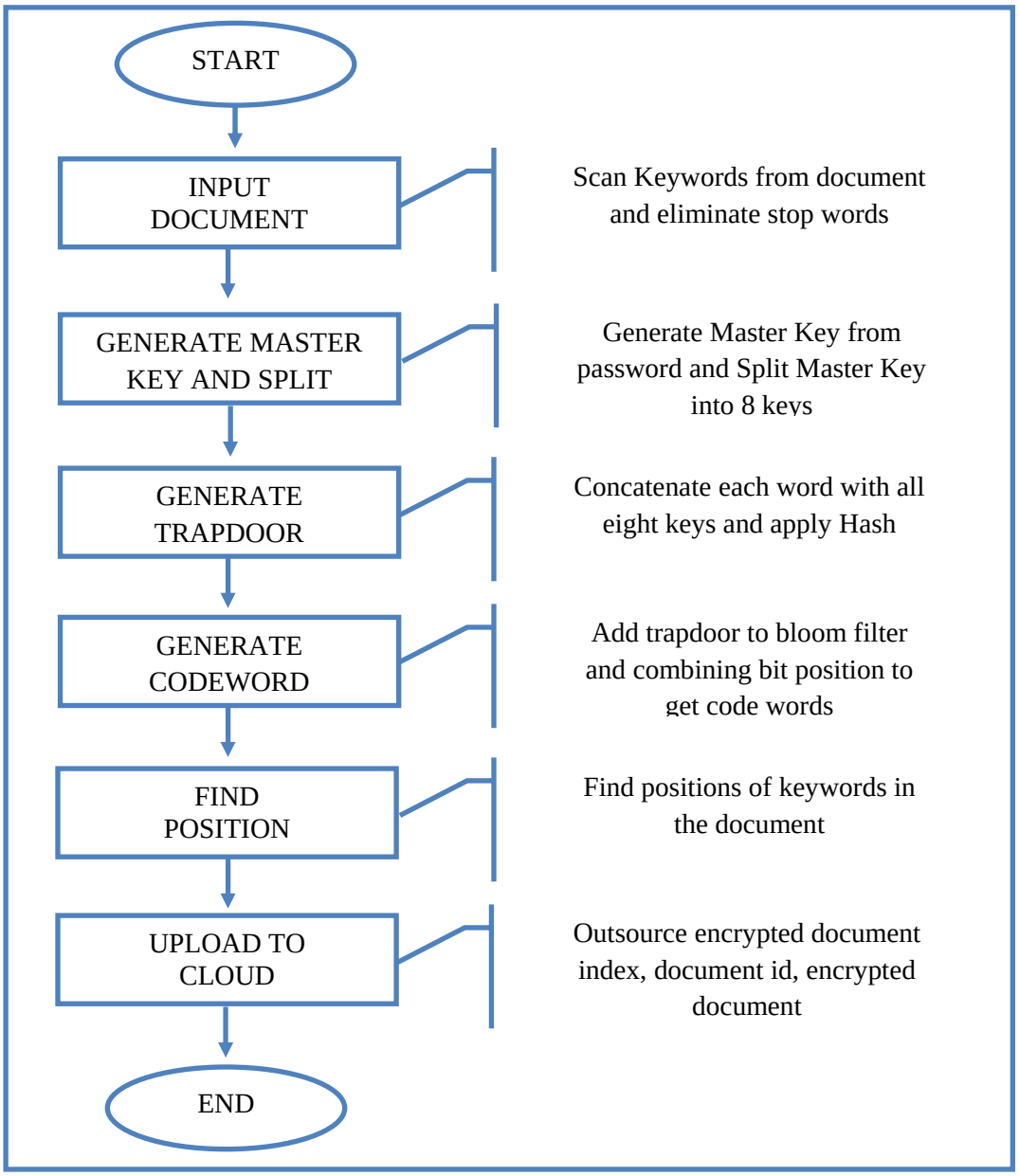

Figure 3.3: Graphical representation of Indexing

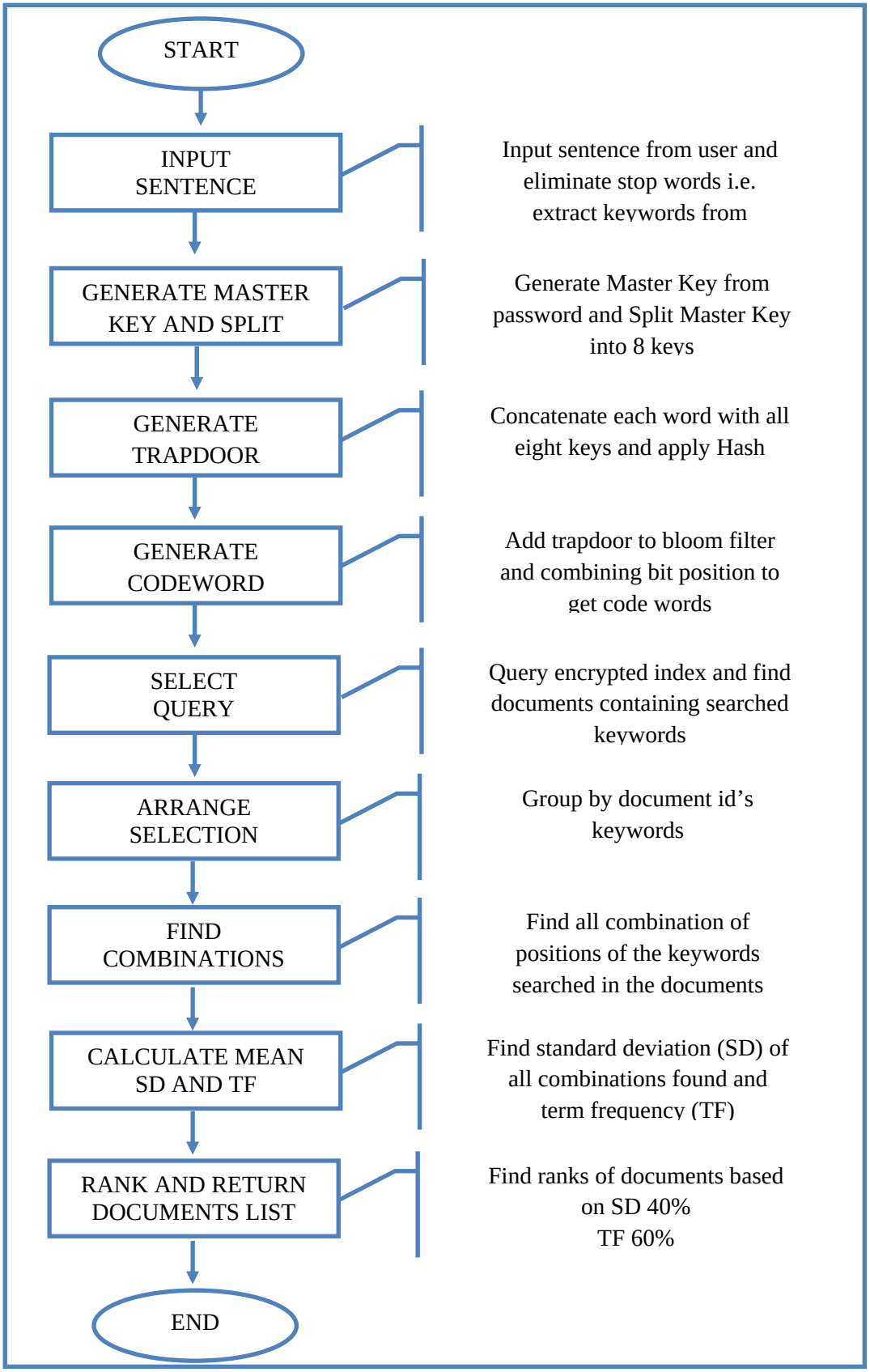

Figure 3.4: Graphical representation of Searching

3.5. ADVANTAGES OF PBSS

The advantages of PBSSare as follows:

1. The preprocessed index of documents makes the search efficient and accurate.
2. PBSS is a unique technique for sentence match for encrypted unstructured data in literature.
3. Exact sentence match and spread sentence match is supported in PBSS. If exact match is found the document is select. If search sentence keywords are spread in the document then the relevance of the document is calculated to select it for final results.
4. Natural Language Search is supported.
5. Punctuation marks are supported.
6. The sentence searching in PBSS is secure as the document indexes are encrypted by two security levels. The cloud or malicious users cannot predict the searched sentence, encrypted documents indexes and encrypted documents.
7. PBSS is an intelligent technique as it predicts most relevant document according to the searched sentence of basis of SD and TF.
8. PBSS works on pattern recognition algorithm to find combinations.

3.6. LIMITATIONS OF PBSS

The limitations of the proposed technique are as follows:

1. Case insensitivity has not been supported.
2. PBSS lack regular expression search.

Chapter 4: Implementation

This chapter briefly explains implementation steps of Indexing and Searching in detail. Results of all implementation steps of Indexing and Searching have been shown in tables.

4.1. IMPLEMENTATION ENVIRONMENT DETAILS

All the required software and hardware requirement which are needed to implement any software system are called environment details. For implementation of PBSS following hardware and software specifications are used.

4.1.1. SYSTEM SPECIFICATIONS

System specificationconstitutes hardware and software specification. Software specificationshave been shown in Table 4.1.

Table 4.1: Software specifications used for testing of PBSS

Software Specifications	
Apache Server Version	Apache/2.2.21 (Win64) PHP/5.3.8
Php Version	PHP/5.3.8
Php My Admin Version	3.4.5
MySQL version	5.5.16-log
Browser	Firefox/Chrome

Hardware specificationshave been shown in Table 4.2.

Table 4.2: Hardware specificationused for testing of PBSS

Hardware Specifications	
Processor	Intel ® Core ™ i5-2430M CPU @ 2.40 GHz
Installed Memory (RAM)	4.00 GB
System Type	64-bit Machine

4.2. IMPLEMENTATION OF PBSS

The implementation of PBSS has two steps as described in proposed scheme Indexing and Searching. All step in Indexing and Searching will be explained with examples in this chapter.

4.2.1. INDEXING

4.2.1.1. Input Document

The data owner input the document for indexing. The document is scanned to extract useful keywords. The stop words are removed from these keywords. The document words are separated on space and a collection of keywords is obtained in an array after removal of stop words. The indexes of this array obtained are positions of the keywords in the document. The results of this step are shown in Table 4.3.

Table 4.3: Input Document

Input Document	
Input	Document 'Amount of searchable encryption techniques proposed in literature.'
Output	A set of distinct words in an array with their positions
Results	Array:([1] =>Amount [2] =>searchable [3] =>encryption [4] =>techniques [5] =>proposed [6] =>literature.)

4.2.1.2. Generate Master Key and Split

A secret password is taken as input and SHA512 is applied on it to get a master key. The master key generated is split into eight equal size distinct keys. The eight keys generated will be used in generate trapdoor step. The master key and eight split keys generation is shown in Table 4.4.

The eight split keys obtained in this step can also be generated with eight different passwords to ensure more secure trapdoors. The customized hash algorithm used can be to generate eight keys with eight different passwords

Table 4.4: Generate Master Key and Split

Generate Master Key and Split	
Input	Secret password
Output	Eight split keys
Results	Array ([0] => e6c83b282aeb2e02, [1] => 2844595721cc00bb [2] => da47cb24537c1779 , [3] => f9bb84f04039e167 [4] => 6e6ba8573e588da1, [5] => 052510e3aa0a32a9 [6] => e55879ae22b0c2d6, [7] => 2136fc0a3e85f8bb)

4.2.1.3. Generate Trapdoor

Generate trapdoor is the first security level of document keywords. The eight keys generated in Table 4.4 are concatenated with the document keywords generated in Table 4.3. The keyword is concatenated with eight keys and hash is applied on each concatenated values. The hash applied is SHA1. The eight hashed values are concatenated with coma inside to generate a trapdoor. The generate trapdoor step is explained with experimental results in Table 4.5.

Table 4.5: Generate Trapdoor

Generate Trapdoor	
Input	word, eight keys, SHA1 hash algorithm
Output	Trapdoor
Results	a104f3e24f622acbdb11b1480c21677b19eacf92,544967dc88058fe18c3e0ba135a6647216fadf82,958452b98248835b769cf0846dfd0b3790a943ad,3a08c7f1b1009ff3fe0ebdddd381c243875ef00a,296708c2666e5e1502403194885f8648719b4467,5e6f0bb55d70f70d0ae30e9b6ff1e0e745e0a408,794cc63d32fb86d6e20fd08f563a8756106b3ccf,394a339521ebc2dea91730501860e7ca02bbbe33

4.2.1.4. Generate Codeword

Generate codeword is second security level for documents keywords. The trapdoor generated in Table 4.5 is added to bloom filter. Bloom filter stores the trapdoor on five bit positions. In bloom filter crc32 hash algorithm is applied five times on the trapdoor after concatenation of five uniform random numbers to get five distinct bit positions for a trapdoor in the bloom filter. The five distinct positions are then concatenated to get a final codeword. The use of uniform random numbers helps to generate the same codewords for the same trapdoor so that during search the

codewords of the searched sentence can be matched with the indexed codewords. The generate codeword step is explained with experimental results in Table 4.6.

Table 4.6: Generate Codeword

Generate Codeword	
Input	Trapdoor, crc32 hash algorithm
Output	Codeword
Results	Bits positions of trapdoor in bloom filter: Array: ([0] => 31632, [1] => 26292, [2] => 65886, [3] => 4588, [4] => 22148) Codewords: 316322629265886458822148

4.2.1.5. Find Positions

When codewords generation is complete position are assigned to codewords. The positions are same as assigned to keywords during indexing. The codeword of a particular keyword will have the same position value which its original keyword has in the document. The codewords with positions are shown in Table 4.7.

Table 4.7: Find Positions

Find Positions	
Input	Codeword
Output	Codeword with position
Results	Array:([1] =>5121651887716876777127128 [2] =>316322629265886458822148 [3] =>16554388826586679013626 3 [4] =>5101510908561006414377 6 [5] =>26407136861822859846730 [6] =>4277550061025457054984)

4.2.1.6. Upload to Cloud

Upload encrypted document index, encrypted document and documents id's to cloud server.The document index generated form the above steps is shown in as shown in Table 4.8.

Table 4.8: Upload to Cloud

<table>
<tr><th colspan="2">Upload to Cloud</th></tr>
<tr><td>Input</td><td>encrypted document index, encrypted document and documents id’s</td></tr>
<tr><td>Output</td><td>Database table rows containing index values</td></tr>
<tr><td>Results</td><td>
<table>
<tr><th>codeword</th><th>position</th><th>did</th><th>enc_doc_name</th></tr>
<tr><td>5121651887716876777127128</td><td>1</td><td>3586c170e3e4262f0eb95a0cc24c5ebb3de14504</td><td>hfyGhJXholF3gSTb4g==</td></tr>
<tr><td>3163226292658864 58822148</td><td>2</td><td>3586c170e3e4262f0eb95a0cc24c5ebb3de14504</td><td>hfyGhJXholF3gSTb4g==</td></tr>
<tr><td>1655438882658667 90136263</td><td>3</td><td>3586c170e3e4262f0eb95a0cc24c5ebb3de14504</td><td>hfyGhJXholF3gSTb4g==</td></tr>
<tr><td>5101510908561006 4143776</td><td>4</td><td>3586c170e3e4262f0eb95a0cc24c5ebb3de14504</td><td>hfyGhJXholF3gSTb4g==</td></tr>
<tr><td>2640713686182285 9846730</td><td>5</td><td>3586c170e3e4262f0eb95a0cc24c5ebb3de14504</td><td>hfyGhJXholF3gSTb4g==</td></tr>
<tr><td>4277550061025457 054984</td><td>6</td><td>3586c170e3e4262f0eb95a0cc24c5ebb3de14504</td><td>hfyGhJXholF3gSTb4g==</td></tr>
</table>
</td></tr>
</table>

4.2.2. SEARCHING

4.2.2.1. Input Sentence

The data user inputs the sentence for searching. The stop words are removed from it. The keywords obtained from sentence are used to match sentence in document in PBSS. The input sentence step is same as explained in section 4.2.1.1.

4.2.2.2. Generate Master Key and Split

Master key is generated from the same secure password which was used in section 4.2.1.2. The master key is then split in to eight equal size keys to be used in trapdoor generation step.

4.2.2.3. Generate Trapdoor

Generate Trapdoor in searching is same as explained in section 4.2.1.3.

4.2.2.4. Generate Codeword

Generate Codeword in searching is same as explained in section 4.2.1.4.

4.2.2.5. Select Query

The searched sentence keywords are checked in the document indexes. A single database query selects the document which contains these search sentence keywords. The database query and results of this step are shown in Table 4.9.

Table 4.9: Select Query

<table>
<tr><th colspan="2">Select Query</th></tr>
<tr><td>Input</td><td>Codewords of searched sentence keywords</td></tr>
<tr><td>Query</td><td>SELECT * FROM codewords
WHERECodeword="5121651887716876777127128" or
codeword="316322629265886458822148" or
codeword="165543888265866790136263" or
codeword="263445748736207330840281" order by did</td></tr>
<tr><td>Output</td><td>Document id’s, codewords, positions, enc_doc_name</td></tr>
<tr><td>Results</td><td>
<table>
<tr><th>Codeword</th><th>position</th><th>Did</th><th>enc_doc_name</th></tr>
<tr><td>512165188771687677
7127128</td><td>1</td><td>3586c170e3e4262f0eb95a0cc2
4c5ebb3de14504</td><td>hfyGhJXholF3gS
Tb4g==</td></tr>
<tr><td>316322629265886458
822148</td><td>2</td><td>3586c170e3e4262f0eb95a0cc2
4c5ebb3de14504</td><td>hfyGhJXholF3gS
Tb4g==</td></tr>
<tr><td>165543888265866790
136263</td><td>3</td><td>3586c170e3e4262f0eb95a0cc2
4c5ebb3de14504</td><td>hfyGhJXholF3gS
Tb4g==</td></tr>
<tr><td>316322629265886458
822148</td><td>2</td><td>3586c170e3e4262f0eb95a0cc2
4c5ebb3de14504</td><td>jf7yI5rholGc3cDZ
8w==</td></tr>
<tr><td>165543888265866790
136263</td><td>3</td><td>3586c170e3e4262f0eb95a0cc2
4c5ebb3de14504</td><td>jf7yI5rholGc3cDZ
8w==</td></tr>
</table>
</td></tr>
</table>

4.2.2.6. Arrange Selection

The selected data is arranged in such a ways that all codewords are grouped according to documents which contain then. The arranged selection is shown in Table 4.10. The algorithm used for arrangement discards those documents which contain less than two codewords due to the reason that a sentence search minimum needs two codewords to be a portion of the sentence search.

Table 4.10: Arrange Selection

Arrange Selection	
Input	Document id's, codewords, positions, enc_doc_name
Output	Group codewords by document id
Results	Array ([kf92barholHbfHOHoA==] => Array ([1] => 512165188771687677712712 8 [2] => 512165188771687677712712 8 [3] => 316322629265886458822148 [4] => 165543888265866790136263) [hfyGhJXholF3gSTb4g==] => Array ([2] => 512165188771687677712712 8 [3] => 316322629265886458822148 [4] => 165543888265866790136263 [5] => 5101510908561006414377 6 [20] => 512165188771687677712712 8 [21] => 316322629265886458822148 [22] => 165543888265866790136263 [23] => 5101510908561006414377 6 [24] => 26407136861822859846730) [jf7yI5rholGc3cDZ8w==] => Array ([2] => 512165188771687677712712 8 [5] => 316322629265886458822148 [7] => 165543888265866790136263 [20] => 5101510908561006414377 6 [23] => 26407136861822859846730) [Iv1qqaXholGjZ5QS4g==] => Array ([1] => 512165188771687677712712 8 [2] => 512165188771687677712712 8 [3] => 316322629265886458822148 [4] => 165543888265866790136263 [5] => 5101510908561006414377 6 [6] => 26407136861822859846730) [t/4+f5/holF5sW47Ag==] => Array ([2] => 512165188771687677712712 8 [31] => 165543888265866790136263 [36] => 5101510908561006414377 6 [39] => 26407136861822859846730 [40] => 512165188771687677712712 8 [44] => 316322629265886458822148 [47] => 165543888265866790136263 [52] => 5101510908561006414377 6 [55] => 26407136861822859846730))

4.2.2.7. Find Combination

Different combinations of the searched sentence are checked in the arranged selected data. If there is a possibility of sentence match then the document is selected. All possible combinations of the sentence match are noted. The best combination is where all the keywords of the searched sentence are present in the document at consecutive positions. The possible combinations are shown in Table 4.11.

Table 4.11: Find Combinations

Find Combination	
Input	Group codewords by document id
Output	Possible sentence combinations
Results	[kf92barholHbfHOHoA==] => Array ([1] => Array ([3] => 316322629265886458822148 [4] => 16554388826586679013626 3 [5] => 5101510908561006414 3776 [6] => 26407136861822859846730) [1] => Array ([4] => 16554388826586679013626 3 [5] => 5101510908561006414 3776 [6] => 26407136861822859846730) [2] => Array ([5] => 5101510908561006414 3776 [6] => 26407136861822859846730)) [hfyGhJXholF3gSTb4g==] => Array ([0] => Array ([2] => 5121651887716876777127128 [3] => 316322629265886458822148 [4] => 16554388826586679013626 3 [5] => 5101510908561006414 3776 [6] => 26407136861822859846730) [1] => Array ([3] => 316322629265886458822148 [4] => 16554388826586679013626 3 [5] => 5101510908561006414 3776 [6] => 26407136861822859846730) [2] => Array ([3] => 316322629265886458822148

```
        [4] => 165543888265866790136263
        [5] => 5101510908561006414377 6
        [6] => 26407136861822859846730
)
```

4.2.2.8. Calculate Mean SD and TF

Calculate SD of all possible combinations resulted in section 4.2.2.7 and take mean. Find TF of all searched sentence keywords in the document which can be calculated from section 4.2.2.6 as the total codewords matched in particular documents are listed under the document.The total number of array values of the document in section 4.2.2.6 is TF of the searched sentence keywords in the document.

4.2.2.9. Rank and Return Document List

Documents are arranged in ascending order of their ranks. The ranks are calculated by following formula. SD and TF are calculated in section 4.2.2.8.

$$Rank = 0.4 \times SD - 0.6 \times TF$$ **Equation 4-1**

Chapter 5: Results and Analysis

In order to achieve the final goal of study experimental work has been done. This chapter will depict these experimental results.

5.1. SOFTWARE TESTING

Software Testing involves verification and validation of the software build for the requirements specification. For the implementation of the PBSS a software testing examines the time constraints for indexing and searching process. The testing of PBSS proves the results as expected. The testing algorithms are explained in section 5.2.

5.2. TESTING MEASURES

The testing measures for PBSS are listed below and an experimental evaluation has been performed as follows:

1. Indexing time ofPBSS
2. Searching time of PBSS
3. Indexing time of ranked keyword search
4. Searching time of ranked keyword search

Comparison of indexing and searching time for PBSS and ranked keyword search has been experimentally evaluated in this section. The graphs and experimental statistical evaluation proved the usefulness of the PBSS.

The comparison of indexing time for both techniques shows that the indexing time of PBSS is slightly more than ranked keyword search. The reason of this time increase in the number of indexing queries to database has increased in PBSS. Whereas the searching time of PBSS is less than ranked keyword search as the algorithm of searching in PBSS is efficient, faster and accurate then ranked keyword search.

5.3. EXPERIMENTAL EVALUATION

5.3.1. PBSS EVALUATION ON BENCHMARK DATA

5.3.1.1. Benchmark Data1

Table 5.1: Benchmark Data 1

Benchmark Data	
Text 1:	"Lightweight multithreading is an integral part of our programming model."
Text 2:	"Lightweight multithreading is an integral part of our programming model. Lightweight multithreading is an integral part of our programming model. Lightweight multithreading is an integral part of our programming model."
Text 3:	"Lightweight lorum multithreading ipsum is an loum integral ipsum part of lorum ipsum our programming lorum model. Lightweight lorum multithreading ipsm is an lorm integral ipsum part of lorum ipsum our programming lorum model."
Text 4:	"Lightweight loum multithreading ipsum is an lorum integral ipsum part of lorum ipsum our programming lorum model. Lightweight loum multithreading ipsum is an lorum integral ipsum part of lorum ipsum our programming lorum model. Lightweight loum multithreading ipsm is an lorum integral ipsum part of lorum ipsum our programming lorum model."
Text 5:	"Lightweight multithreading lorum is an integral part of our programming model. Lightweight lorum multithreading is an integral ipsum part of our programming lorum model. Lightweight multithreading is lorum an integral part of our programming lorum model."

Results		
Search Sentence	"Lightweight multithreading is an integral part of our programming model."	
Rank	***Document name***	***Reason***
1	2.txt	less SD and more TF
2	5.txt	more SD and more TF
3	4.txt	more SD and more TF
4	3.txt	more SD and less TF
5	1.txt	less SD and less TF

5.3.1.2. Benchmark Data 2

Table 5.2: Benchmark Data 2

Benchmark Data	
Text 1:	"Chronic lorum hepatitis B develops ipsum more commonly lorum ipsum in people who are infected with the virus at an early age (often at birth)."
Text 2:	"Chronic lorum hepatitis B develops more ipsum commonly in people who are infected with the virus at an early age (often at birth)."
Text 3:	"Chronic hepatitis B develops more commonly in people who are infected with the virus at an early age (often at birth). Chronic hepatitis B develops more commonly in people who are infected with the virus at an early age (often at birth). Chronic hepatitis B develops more commonly in people who are infected with the virus at an early age (often at birth)."
Text 4:	"Chronic lorum hepatitis B develops ipsum more commonly lorum ipsum in people who are infected with the virus at an early age (often at birth). Chronic lorum hepatitis B develops ipsum more commonly lorum ipsum in people who are infected with the virus at an early age (often at birth). Chronic lorum hepatitis B develops ipsum more commonly lorum ipsum in people who are infected with the virus at an early age (often at birth)."
Text 5:	"Chronic lorum ipsum hepatitis lorum ipsum B develops more lorum ipsum commonly lorum in people ipsum lorum ipsum who are infected with the virus lorum ipsum lorum ipsum lorum ipsum at an early age (often at birth)."

Results	
Search Sentence	"Chronic hepatitis B develops more commonly in people who are infected with the virus at an early age (often at birth)."

Rank	***Document name***	***Reason***
1	3.txt	less SD and more TF
2	4.txt	more SD and more TF
3	2.txt	less SD and less TF
4	1.txt	more SD and less TF
5	5.txt	more SD and less TF

5.3.1.3. Benchmark Data 3

Table 5.3: Benchmark Data 3

Benchmark Data	
Text 1:	"amount of searchable encryption schemes have been proposed amount lorum of searchable ipsum encryption schemes have lorum been proposed amount of searchable encryption schemes have been proposed amount of searchable encryption schemes have been proposed"
Text 2:	"amount important of cloud searchable value encryption schemes have been implemented proposed amount lorum of ipsum searchable lorum encryption ipsum lorum schemes ipsum have been proposed"
Text 3:	"amount important of cloud searchable value encryption schemes have been implemented proposed amount lorum of ipsum searchable lorum encryption lorum lorum ipsum schemes lorum lorum have been proposed amount ipsum of ipsum searchable ipsum lorum ipsm encryption ipsm lrum ipsm schemes ipsm have been proposed amount lorum of ipsm loum searchable ipsm lrum encryption ipsm lrum ipsm lorm schemes ipsm have been proposed"
Text 4:	"amount of searchable encryption schemes have been proposed"
Text 5:	"amount of searchable encryption schemes have been proposed amount lorum of searcha1ble lorum encryption schemes have ipsum been proposed"

Results		
Search Sentence	"amount of searchable encryption schemes"	
Rank	***Document name***	***Reason***
1	1.txt	less SD and more TF
2	3.txt	more SD and more TF
3	5.txt	more SD and more TF
4	2.txt	more SD and less TF
5	4.txt	less SD and less TF

5.3.2. INDEXING TIME OFPBSS

The time required to index a document is called indexing time and it is calculated by indexing the document. A set of 150 documents are taken for experimental evaluation but 20 documents are listed in the table below to make the concise and understandable graph as 150 files index time if plotted on graph makes it hard to understand. The time requiredto index number of distinct wordsin each document has been shown in Table 5.4.

Table 5.4: Indexing time ofPBSS

Document name	No. Of index word	Indexing Time (Sec)
How to search eu gin goh.txt	180	10
Cloud_computing_security_risk.txt	421	22
Optimizing security of cloud computing within the dod.txt	698	32
Deploying public key infrastructure as a cloud service.txt	1169	63
Service-oriented modeling and architecture.txt	1877	96
A wrapping approach and tool for migrating legacy components.txt	2105	111
Risk management in global software development process planning.txt	2375	120
Project management a case study.txt	2556	131
Data protection-aware design for cloud services.txt	2873	148
The implementation and deployment of an erp system an industrial case study.txt	3536	181
Database-agnostic transaction support for cloud infrastructures.txt	3631	184
Secure multidimensional range queries over outsourced data.txt	4696	227
What do software practitioners really think about project success a cross-cultural comparison.txt	4427	215
A method engineering based legacy to soa migration method.txt	5240	256
Authorized private keyword search over encrypted data cloud computingicdcs11.txt	6570	340

The statistical results obtained in Table 5.4 have been graphically shown in Figure 5.1.

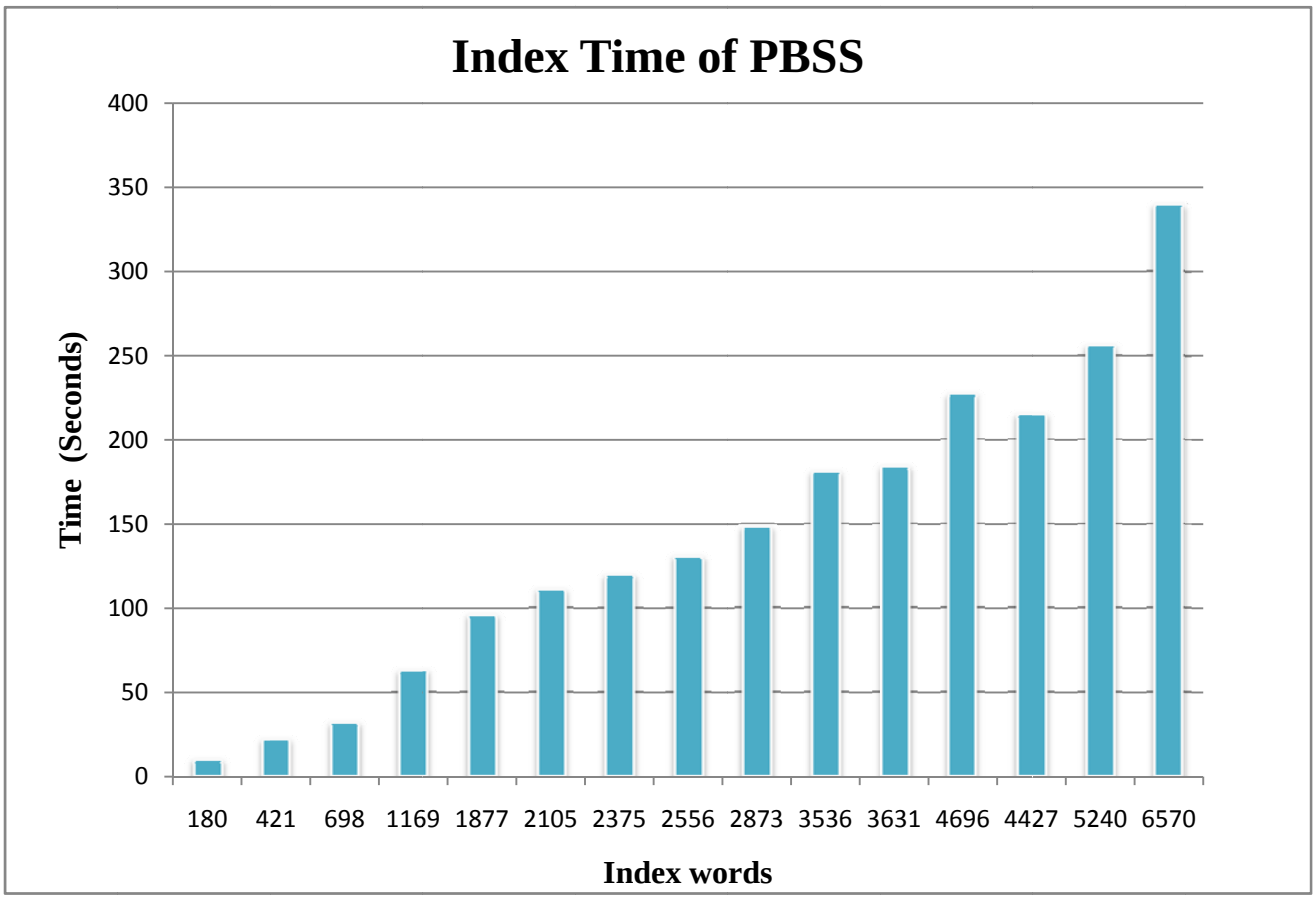

Figure 5.1: Graphical representation of index time ofPBSS

Figure 5.1 depicts that as the number of indexed words shown horizontallyincrease in the documents the indexing time shown vertically is also increasing. Hence it can be said that:

Indexing a document αNo. of index words

5.3.2.1. Index time for one keyword

From the results in Table 5.4the indexing time for one keyword is calculated as:

Total keywords = 42354

Total time consumed = 2136

One keyword index time = Total time / Total words = 2136/42354 = .0504 seconds

5.3.2.2. Words indexed in one second

The total number of keywords indexed per second can be calculated as follows:

Total keywords= 42354

Total time consumed = 2136

Keywords indexed in one sec = Total keywords / Total time consumed = 42354/2136 = 19.82 words

These results can be used for time estimation of time for indexing of specific number of keywords. The increase in the number of index words increase the indexing time. The reason PBSS takes more time to index documents is that each individual words has to be indexed with its position in the document instead in ranked search only distinct words are indexed with their ranks. The similar words only increment the rank.

5.3.3. SENTENCE SEARCH TIMEOFPBSS

When a sentence is searched its keyword's codewords are matched in the document indexes. After the matching of keyword's codewords the collection of data is arranged and the combinations of the sentence are found. Mean standard deviation of these combination and TF ranks the document with the formula given in section 4.2.2.9. The time required to search a sentence has been shown in Table 5.5. Twenty sentences form different indexed documents have been taken as a sample data set for finding search time.

Table 5.5: Search time of PBSS

Searched Sentences	Search Time (Sec)
Cloud Computing sees a technical and cultural shift of computing service	2.792
The problems and risks of poor architectural practices are well known	2.005
Cloud computing is fraught with security risks, according to analyst firm Gartner.	2.468
The Terminal Emulator manages the communications	1.505
Business Information Integration from XML and Relational Databases	2.059

Sources	
Deploying Public Key Infrastructure as a Cloud Service	2.023
2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science	2.617
IDENTIFIED SECURITY ATTRIBUTES IN CLOUD COMPUTING	1.593
Most large organizations are deeply mired in their information systems (IS) sins of the past.	2.682
Pregel: A System for Large-Scale Graph Processing	1.59
Access Control Policies	1.472
STATE- OF-THE-ART ON DW DESIGN APPROACHES	1.439
Certain Query (QS): that selects tuples that certainly qualify the conditions	1.706
Microfocus has developed a DLL object.	1.481
Visual studio .NET environment is extendable.	1.504
the Z-schemas and the notations we have used the Z-word tool.	1.636
Encrypted Processes for Oblivious Data Retrieval	1.615
Server-Side Protection: Virtual servers and applications,	1.637
OLAP (On Line Analytical Processing) data warehouse	1.887
Exploit facets' ability to reference an ontology	1.63

Figure 5.2 depicts that the vertical axis show the time required for searching specific keywords and on horizontal axis searched keywords have been shown. It has been observed that the search time remains in the range of 1.5 seconds to 3.5 seconds. The statistical results obtained in Table 5.5 have been graphically shown in Figure 5.2.

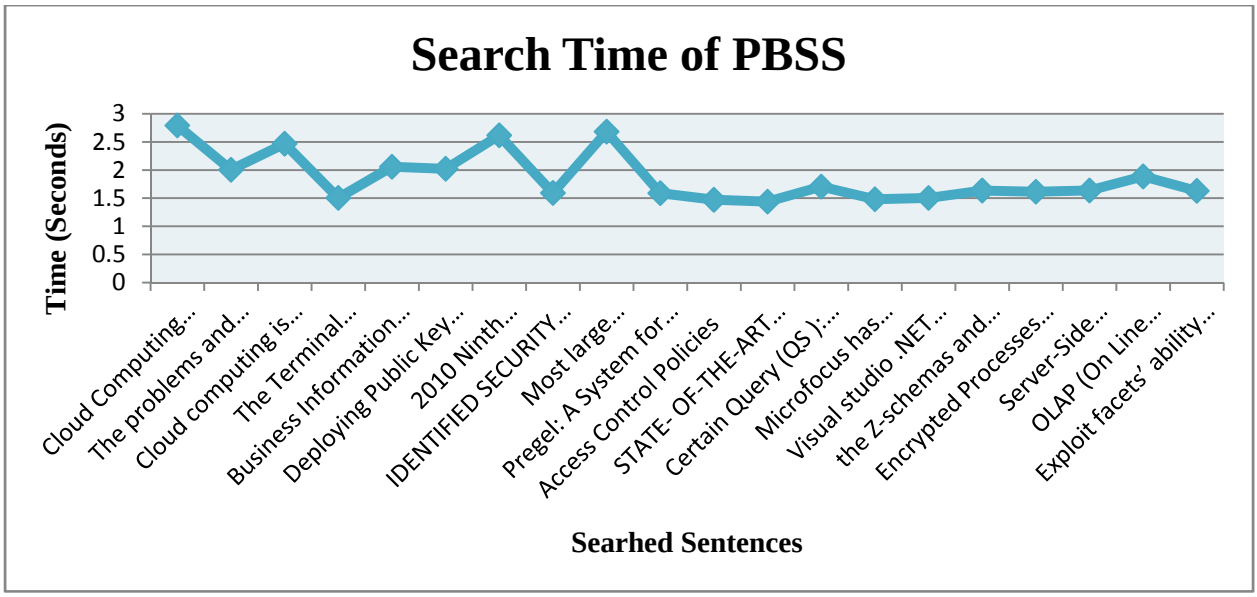

Figure 5.2: Graphical representation of search time of PBSS

5.3.4. INDEXING TIME OF RANKED KEYWORD SEARCH

Ranked keyword search technique proposed by in [50] is taken for comparison which is also a searching technique for encrypted unstructured data. The ranked keyword search technique discussed in [50] comprise of two steps: Index Generation and Search. The indexing time consumed in ranked keyword search technique for the same documents indexed with PBSS are shown in Table 5.6

Table 5.6: Indexing time of ranked keyword search

Document name	No. Of index word	Indexing Time (Sec)
How to search eu gin goh.txt	126	2
Cloud_computing_security_risk.txt	371	8
Optimizing security of cloud computing within the dod.txt	556	16
Deploying public key infrastructure as a cloud service.txt	878	25
Service-oriented modeling and architecture.txt	1085	29
A wrapping approach and tool for migrating legacy components.txt	1205	33
Risk management in global software development process planning.txt	1360	40
Project management a case study.txt	1463	42
Data protection-aware design for cloud services.txt	1763	49

The implementation and deployment of an erp system an industrial case study.txt	2020	58
Database-agnostic transaction support for cloud infrastructures.txt	2251	65
Secure multidimensional range queries over outsourced data.txt	2420	77
What do software practitioners really think about project success a cross-cultural comparison.txt	2762	82
A method engineering based legacy to soa migration method.txt	2933	85
Authorized private keyword search over encrypted data cloud computingicdcs11.txt	3305	102

The statistical results shown in Table 5.6 have been graphically shown in Figure 5.3.Figure 5.3 shows the indexing time with technique proposed in [50]. The indexing time of this technique is less than PBSS because in PBSS each word at its particular position is ranked while in the ranked indexing only rank of the commonkeywords is increased.

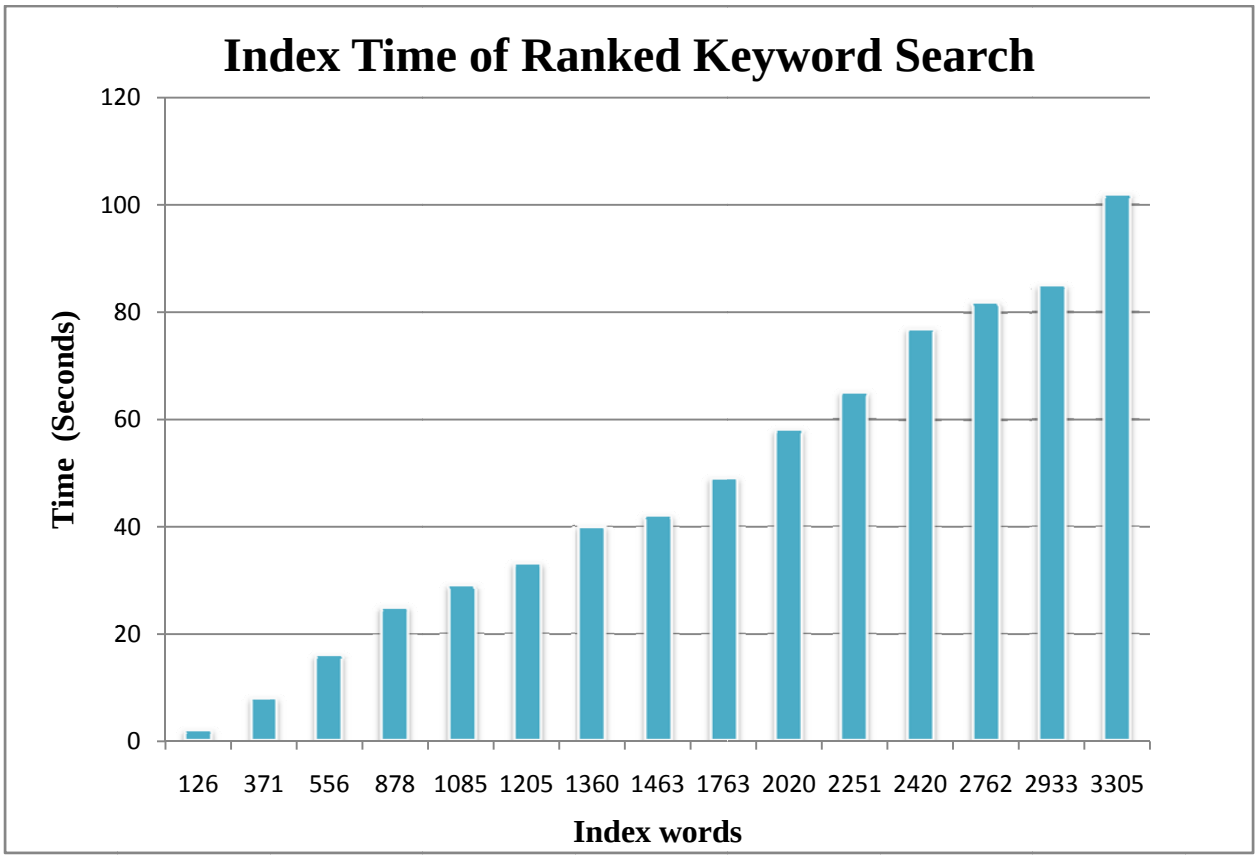

Figure 5.3: Graphical representation ofindex time of ranked keyword search

The total keywords to index are less in ranked indexing instead of PBSS. For example a 'Cloud Computing' is present in document three times. In ranked indexing it will be inserted to indexing database once. While in PBSS it will be inserted to database three times with three different positions. The implementation scenario of indexing in PBSS can be improved to get better results. The graph representing the indexing time of ranked search indexing is shown in Figure 5.3.

5.3.5. SEARCH TIME OF RANKED KEYWORD SEARCH

The search time of ranked keyword search technique is shown in Table 5.7 for the same searched sentences searched with PBSS. Same data set is being used to compare search time of ranked search and PBSS. The search time consumed with ranked keywords search of [50] is little bit more than PBSS. This proved that PBSS is more efficient with respect to search time then ranked keyword search. The statistical data obtained from experimentation is shown in Table 5.7.

Table 5.7: Search time of ranked keyword search

Searched Sentences	Search Time (Sec)
Cloud Computing sees a technical and cultural shift of computing service	4.634
The problems and risks of poor architectural practices are well known	2.676
Cloud computing is fraught with security risks, according to analyst firm Gartner.	4.01
The Terminal Emulator manages the communications	3.469
Business Information Integration from XML and Relational Databases Sources	4.101
Deploying Public Key Infrastructure as a Cloud Service	2.116
2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science	2.104
IDENTIFIED SECURITY ATTRIBUTES IN CLOUD COMPUTING	1.529
Most large organizations are deeply mired in their information systems (IS) sins of the past.	4.176
Pregel: A System for Large-Scale Graph Processing	3.374

Access Control Policies	2.708
STATE- OF-THE-ART ON DW DESIGN APPROACHES	1.76
Certain Query (QS): that selects tuples that certainly qualify the conditions	3.294
Microfocus has developed a DLL object.	3.584
Visual studio .NET environment is extendable.	3.221
the Z-schemas and the notations we have used the Z-word tool.	3.992
Encrypted Processes for Oblivious Data Retrieval	3.558
Server-Side Protection: Virtual servers and applications,	3.414
OLAP (On Line Analytical Processing) data warehouse	4.295
Exploit facets' ability to reference an ontology	3.26

The statistical results obtained in Table 5.7 have been graphically shown in Figure 5.4.Figure 5.4depicts that on horizontal axis searched sentences have been shown and on vertical axis the time required to search for a specific sentence has been shown. It has been observed that the search time remains almost constant around 1-5 seconds for 150 documents.

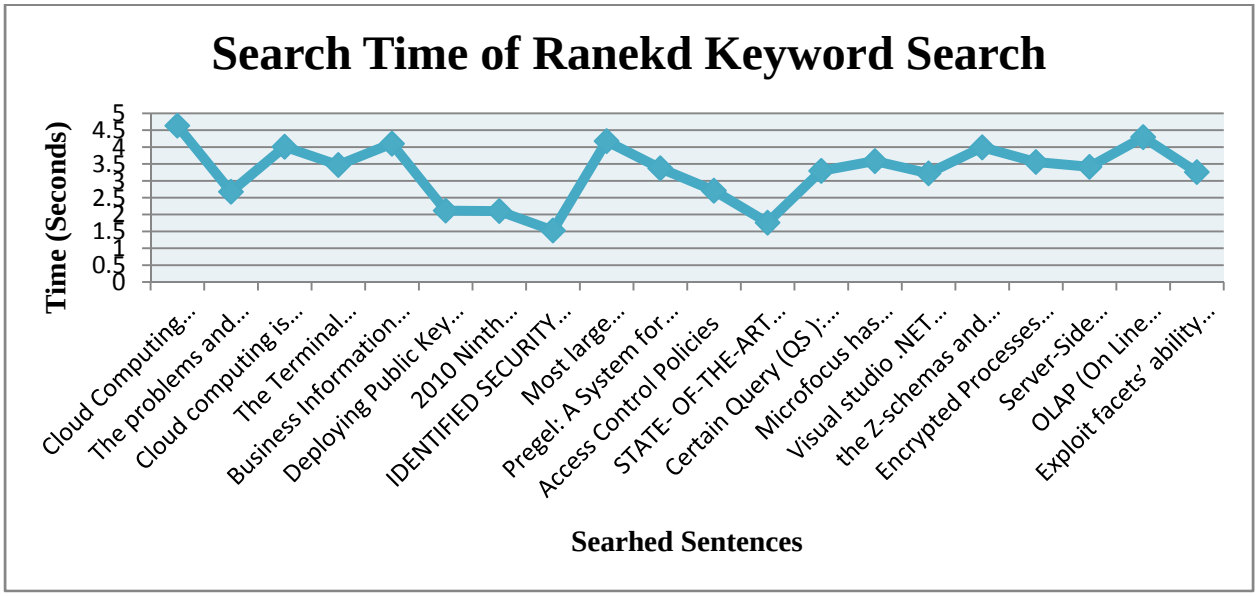

Figure 5.4: Graphical representation of search time ofranked keyword search

5.3.6. INDEX TIME COMPARISON OF PBSS AND RANKED KEYWORDSEARCH

The comparison of time taken to preprocess the document to generate the encrypted indexes in PBSS and ranked keyword search technique[50] is shown in Figure 5.5. It can be seen that indexing time of PBSS is more than ranked search due to more database queries for insertion of codewords to the database. The time of indexing in PBSS can be reduced with minimizing the database queries.

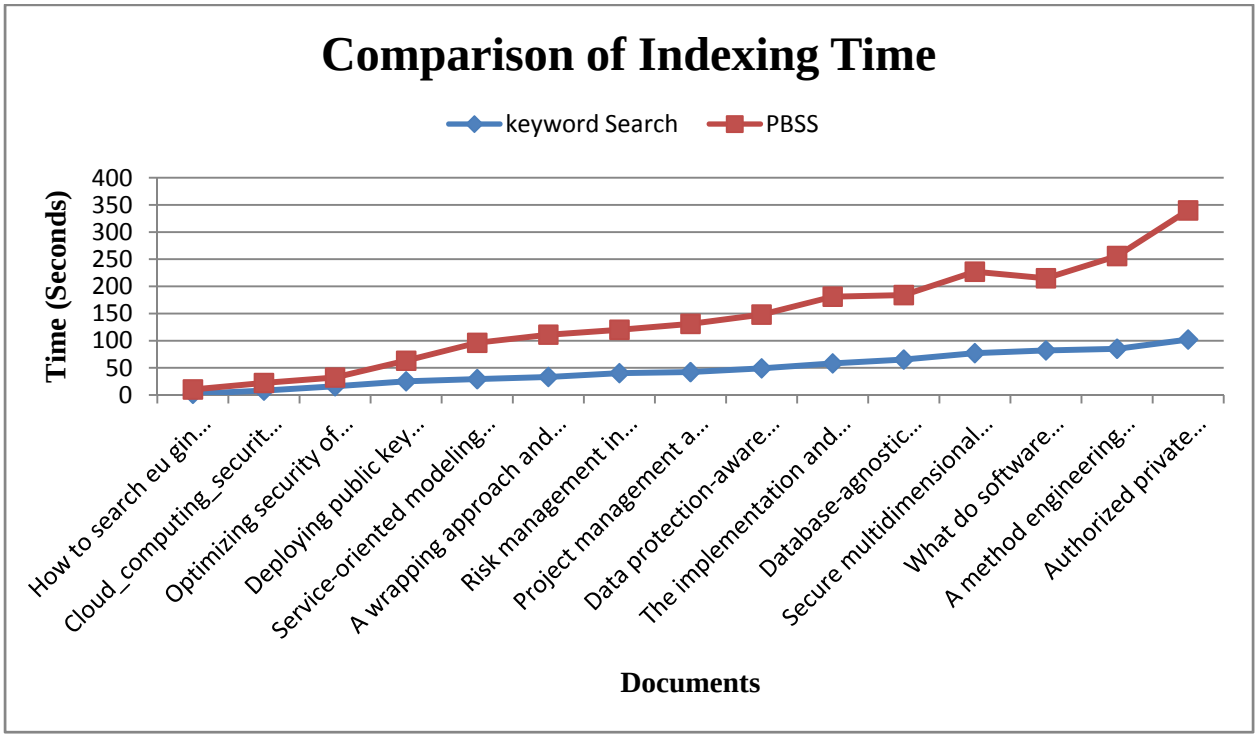

Figure 5.5: Graphical representation of comparison of indexing time

5.3.7. SEARCH TIME COMPARISON OF PBSS AND RANKED KEYWORD SEARCH

Search time comparison of PBSS and ranked keyword search technique [50] is shown in Figure 5.6. It can be seen that PBSS takes less time for sentence searching then ranked keywords search technique. The same data set is used to calculate the searching time of PBSS and ranked keyword search. The statistical analysis proved that search time of PBSS is less thanranked keyword search.

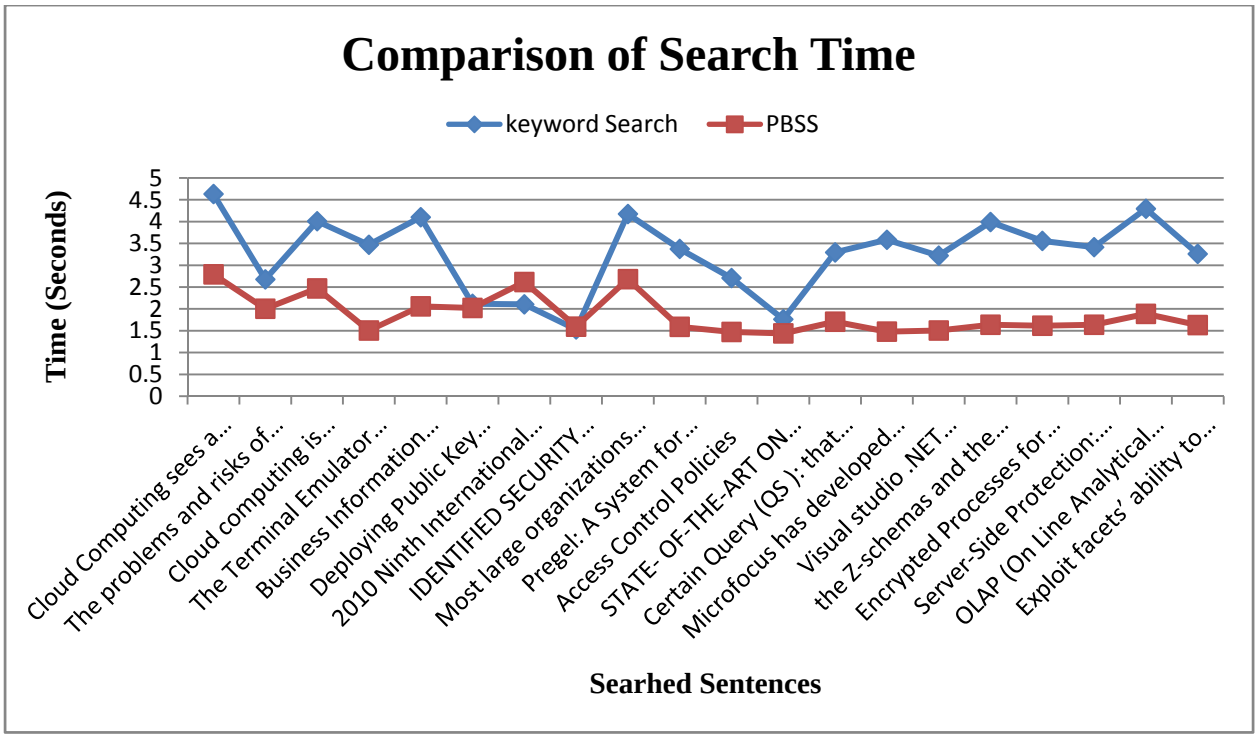

Figure 5.6: Graphical representation of comparison of search time

5.4. SECURITY ACHIEVEMENT IN PBSS

5.4.1. SECURITY OF DOCUMENT INDEX

The basic problem to propose PBSS and implement is to provide secure and efficient searching for encrypted unstructured data. The encrypted unstructured data is difficult to search with traditional encryption techniques. Therefore secure indexing based technique PBSS is proposed and proved to be efficient and accurate for sentence searches in encrypted unstructured data. The index generated with PBSS includes two security measures to ensure security of the indexes. The first security measure is trapdoor and second is codewords. The algorithms and steps of both these steps have been briefly explained in section 3.4.1.3 and 3.4.1.4 respectively. These two security level achieve top security needed for cloud environments as the data passed through these security levels is difficult to hack and unpredictable. The use of eight distinct keys to generate a trapdoor can be individually generated with eight different passwords which provide customizability. The hash algorithms used in indexing are customizable as data owner can use other hash algorithms available or can make custom hash function according to needs.

5.4.2. SECURITYOFSEARCHING

The searching of sentence also includes two security measures as discussed in indexing. The searched sentence keywords are first converted to trapdoors and then to codewords. These two levels of security fulfill the required need of security for cloud environments and data passed through these two security levels cannot be hacked or predicted by cloud. The searched sentence keywords are directed towards data owner where they are converted to codewords and sent to cloud server to fetch matching documents in ranked order the most relevant document on top. The network security layer can also be included as required like secure certificates to ensure only authorized user can send sentence of searching towards the data owner.

Chapter 6: Conclusion and Future Work

This chapter concludes the research work and contributions of the research work. Future directions to extend this research work have been given. The summarized security achievements have been given at end of this chapter.

6.1. CONCLUSION

Security considerations in cloud environment are most critical. The main hindrance in adaption of cloud service is security threat to the confidential and private data of data owners and data user. Cloud computing provide cost effective and efficient resource at a risk of security. The enterprises which have confidential and private data are reluctant to adapt cloud services due to the security threat so the main focus of cloud service providers is towards security threats of the cloud. In recent surveys about the cloud services risks security risks are at top.

Due to security threats in cloud services the data owner and data user have to store encrypted data on the cloud server. Different encryption techniques have been proposed by researcherfor the security of cloud data at different levels. With the use of encryption technique on the data searching becomes difficult. The data users have to face lot of difficulties in searching from encrypted data. Different searchable encryption techniques have been proposed by researcher for encrypted unstructured data but these techniques are not efficient and accurate. Keyword based searching technique for encrypted unstructured data have been proposed in literature but no implementation details are provided. No sentence searching technique has been proposed in literature so there is a need of sentence searching technique for encrypted unstructured data. PBSS is proposed as a sentence searching technique for encrypted unstructured data. PBSS preprocess the document and generates the encrypted document index which will be used for search operations. All searching operations will be performed on the encrypted index of the document. The encrypted index generation follows two security levels to ensure data security and privacy in the cloud environment. The index is encrypted so that cloud server or malicious users will not be able to deduce any information about searched data or content of the documents.

For achieving the research objectives following workflow have been adopted. In Chapter 1 brief introduction of cloud environment and cloud security threats are discussed. Basic terminologies of cloud environment have been discussed in detail. Research scope explains in detail the scope of the research. Motivation and problem definition are addressing why this research was needed and what were the problems which need to be solved. Chapter 2 highlights different searchable encryption techniques proposed in literature and the advantages and disadvantages. Complete analyses of proposed techniques in literature have been provided for the better understanding of the motivation of this research and problems identified in literature. Framework of PBSS and its working is discussed in Chapter 3. Detailed explanation of steps involved in PBSS is provided in this chapter. The implementation stages have been elaborated in Chapter 4. Each step of PBSS implementation with result of actual implementation is shown in tables. The analysis of the results and PBSS has been demonstrated in Chapter 5.

The research objectives have been obtained successfully. The documents of data owner are indexed before outsourcing to cloud server for searching. The index is made secure with two levels of security to ensure privacy and confidentiality. Secure sentence search is provided with the same security level as used during indexing. The search process is efficient and accurate.

Decryption of documents before searching is not required as in existing techniques. The documents are decrypted on retrieval by data user. The cloud server or malicious users cannot deduce searched content or document content. The searched documents are most relevant according to the searched sentence. The ranking of document is based on SD and TF and all possibilities of sentence match are considered in PBSS. PBSS achieves high security, efficiency and accuracy.

6.1. CONTRIBUTIONS

The contributions of this research are as follows:

1. Encrypted index of the document is created.
2. Encrypted index is secure for cloud environment and provide secure search.
3. Two level of security have been introduced in PBSS.
4. Sentence search is provided with most relevance criteria.

5. The retrieved searched documents are in the ranked order based on term frequency and standard deviation of keywords positions.
6. Overhead of decryption before searches is reduced.
7. The search using PBSS has been proved secure and efficient.
8. A new technique for sentence search is proposed in this research.

PBSS provides efficient and accurate sentence search as compared to the keyword based search.

6.1.1. SECURITY ACHIEVEMENTS

The security achievements of PBSS are shown in Table 6.1.

Table 6.1: Security achievements

Salient Features	Achievement
Hash algorithms used are not fixed	Customizability
More than eight keys can be used to generate trapdoors	Security
Two security levels implemented for indexing	Security (Trapdoor & Codeword)
Two security levels implemented for searching	Security (Trapdoor & Codeword)
Any encryption technique can be used instead of AES 256 bit	Customizability
PBSS search mechanism is invisible to users	Data privacy and Confidentiality
Cloud server is unaware of encrypted index and encrypted searching	Data privacy and Confidentiality
Cloud server is unaware of the documents contents	Data privacy and Confidentiality

6.1.2. AREAS OF APPLICATIONS

PBSS is proposed for cloud environment where a large amount of unstructured data is stored in encrypted form. PBSS is not restricted to cloud environment only. This can be used on individual server with in an organization for data security. PBSS technique is an easily deployable working technique which can be customized according to the users need. PBSS can be implemented in banking systems, defense systems and other areas where security achievement is major concern

with efficient and accurate search on encrypted unstructured data. PBSS can be used by organization like IEEE and Springer for better search on the encrypted unstructured data. The organizations are dealing with huge amount of unstructured data in form of research papers stored in their database. The use of PBSS is not restricted to the areas mentioned above. It is open to all areas where encrypted unstructured data need to be searched efficiently and accurately.

6.2. FUTURE WORK

Current research is based on position based sentence search. The ranking mechanism parameters introduced in this research are SD and TF. The research can be extended in following areas:

6.2.1. RANKING MECHANISM

PBSS uses SD and RF for ranking the documents i.e. two parameter for ranking a document. Ranking formula can be improved and other parameter need to be considered like document length and number of index words.

6.2.2. CASE INSENSITIVITY & SUBMATCH SEARCH

PBSS does not support case insensitivity and sub matches. PBSS can be extended to enable case insensitivity and sub match searches. Case insensitivity can be achieved with minor change but sub match search need a careful attention.

6.2.3. SECURE DATABASE

The data base used for storing documents indexes can be made secure using encrypted database and encrypted queries. Database security can be achieved by technique discussed in literature i.e. Order Preserving encryption techniques. Database security can be achieved by techniques and standards proposed in [37], [38]. A hybrid approach may also be implemented which facilitates all above features.

REFERENCES

[1]. Al-Aqrabi, H.; Lu Liu; Jie Xu; Hill, R.; Antonopoulos, N.; Yongzhao Zhan, "Investigation of IT Security and Compliance Challenges in Security-as-a-Service for Cloud Computing." Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW), 2012 15th IEEE International Symposium on , vol., no., pp.124,129, 11-11 April 2012

[2]. Gurudatt Kulkarni, Ramesh Sutar, Jayant Gambhir, "Cloud Computing-Storage as Service." International Journal of Engineering Research and Applications (IJERA), Vol. 2, Issue 1,Jan-Feb 2012, pp.945-950

[3]. P. Mell and T. Grance, "Draft Nist Working Definition of Cloud Computing." http://csrc.nist.gov/groups/SNS/cloudcomputing/ index.html, Jan. 2010.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[4]. A. Tripathi and A. Mishra, "Cloud Computing Security Considerations." 2011 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), 14-16 Sept., Xi'an-Chaina, pp 1-5

[5]. The NIST Definition of Cloud Computing, version 15, by Peter Mell and Tim Grance, October 7, 2009, National Institute of Standards and Technology (NIST), Information Technology Laboratory

[6]. Sabahi, Farzad, "Cloud computing security threats and responses." In Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on, pp. 245-249. IEEE, 2011.

[7]. Kalagiakos, Panagiotis, and Panagiotis Karampelas, "Cloud computing learning." In Application of Information and Communication Technologies (AICT), 2011 5th International Conference on, pp. 1-4. IEEE, 2011.

[8]. Almorsy, Mohemed, John Grundy, and Amani S. Ibrahim, "Collaboration-based cloud computing security management framework." In Cloud Computing (CLOUD), 2011 IEEE International Conference on, pp. 364-371. IEEE, 2011.

[9]. Sengupta, S.; Kaulgud, V.; Sharma, V.S., "Cloud Computing Security--Trends and Research Directions." [J]. Services (SERVICES) , 2011 IEEE World Congress on 2011, Page (s): 524- 531

[10]. Kantarcioglu, Murat, Alain Bensoussan, and SingRu Hoe, “Impact of security risks on cloud computing adoption.” In Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on, pp. 670-674. IEEE, 2011.

[11]. Huth, Alexa and Cebula, James. “The Basics of Cloud Computing.” 2011. Available from: http://www.us-cert.gov/reading_room/USCERT-CloudComputingHuthCebula.pdf (accessed March 25, 2013

[12]. Kresimir Popovic , Zeljko Hocenski, “Cloud computing security issues and challenges.” in The Third International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services, 2010, pp. 344-349

[13]. Deyan Chen; Hong Zhao,” Data Security and Privacy Protection Issues in Cloud Computing.” International conference on computer science and engineering, Vol 3 March 2012

[14]. Ramgovind, S. Eloff and M.M. Smith, E.,“The management of security in Cloud computing.” In Information Security for South Asia (ISSA), 2010, pp. 1-7

[15]. J. Heiser and M. Nicolett, “Assessing the security risks of cloud computing.” Gartner Report, 2009. http://www.gartner.com/DisplayDocument? id=685308 (accessed March 26, 2013)

[16]. Youssef Gahi, Mouhcine Guennoun, Zouhair Guennoun, Khalil El-Khatib, “Encrypted Processes for Oblivious Data Retrieval.” 6th International Conference on Internet Technology and Secured Transactions, 11-14 December 2011, Abu Dhabi, United Arab Emirates.

[17]. Barnatt, C. (2010) A Brief Guide to Cloud Computing, London: Constable & Robinson Ltd

[18]. Zhang, Shaomin, Xiaoqiang Li, and Baoyi Wang, “Study on the Protection Method of Data Privacy Based on Cloud Storage.” International Journal of Information and Computer Science 1, no. 2 (2012)

[19]. Putri, N.R., Mganga, M.C. 2011, “Enhancing Information Security in Cloud Computing Services using SLA Based Metrics.” Master’s thesis: Blekinge Institute of Technology. Available from:http://netlearning2002.org/fou/cuppsats.nsf/all/780daa1ef3027f82c1257864001c2d87/$file/MCS-2011-03.pdf (accessed March 26, 2013)

[20]. Jensen, M., Schwenk, J. O., Gruschka, N. and Iacono, L. L. 2009,"On Technical Security Issues in Cloud Computing." In IEEE International Conference on Cloud Computing (CLOUD-II 2009), Bangalore, India, September 2009, 109-116

[21]. Anu Rathi, Yogesh Kumar Anish Talwar, "Aspects of Security in cloud computing." International Journal Of Engineering And Computer Science, Volume 2 Issue 4 April, 2013 Page No. 1361-1363

[22]. Mohammed A. AlZain, Eric Pardede, Ben Soh, James A. Thom, "Cloud Computing Security: From Single to Multi-clouds." hicss, pp.5490-5499, 2012 45th Hawaii International Conference on System Sciences, 2012

[23]. Ayushi, "A Symmetric Key Cryptographic Algorithm." 2010 International Journal of Computer Applications (0975 - 8887), Volume 1 – No. 15, Pages: 1-4

[24]. Forouzan, B.A. (2007) Data Communications and Networking, 4th edition, New York: McGraw-Hill

[25]. W. Harrower, "Searching encrypted data." Department of Computing, Imperial CollegeLondon, Tech. Rep, 2009

[26]. C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data." IEEE Transactions on Parallel And Distributed Systems, VOL. 23, NO. 8, AUGUST 2012

[27]. A. Broder and M. Mitzenmacher, "Network applications of bloom filters: A survey." Internet Mathematics, vol. 1, no. 4, pp. 485–509, 2004

[28]. S. Dharmapurikar, P. Krishnamurthy, and D. E. Taylor, "Longest prefix matching using bloom filters." IEEE/ACM Transactions on Networking, vol. 14, no. 2, pp. 397–409, 2006

[29]. Chen, Yang, Abhishek Kumar, and Jun Xu, "A new design of Bloom filter for packet inspection speedup." In Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE, pp. 1-5. IEEE, 2007

[30]. M. Mitzenmacher, "Compressed bloom filters". IEEE/ACM Transactions on Networking, vol. 10, no. 5, pp. 604–612, October 2002

[31]. C.Antognini.: Bloom Filters, http://antognini.ch/papers/BloomFilters20080620.pdf (accessed April 4, 2013)

[32]. N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, “Privacy-preserving multi-keyword ranked search over encrypted cloud data,” in Proc. of INFOCOM, 2011, on 10-15 April, pp 829-837

[33]. Dawn Xiaodong Song, David Wagner, and Adrian Perrig, “Practical Techniques for Searches on Encrypted Data.” In proceedings of IEEE Symposium on Security and Privacy, May 2000

[34]. D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, “Public Key Encryption with Keyword Search.” In C. Cachin and J. Camenisch, editors, Advances in Cryptology EUROCRYPT 2004, volume 3027 of LNCS, pages 506–522. Springer, 2004

[35]. Eun-Kyung Ryu; Takagi, T., “Efficient Conjunctive Keyword-Searchable Encryption.” Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on, vol.1, no., pp.409,414, 21-23 May 2007

[36]. E.-J. Goh, Secure Indexes, Technical Report 2003/216, Cryptology Print Archive, http://eprint.iacr.org/, 2003

[37]. Y.-C. Chang and M. Mitzenmacher, “Privacy Preserving Keyword Searches on Remote Encrypted Data.” Proc. Int‘l Conf. Applied Cryptography and Network Security (ACNS ‘05), 2005

[38]. Reza Curtmola, Juan Garay, Seny Kamara, and Rafail Ostrovsky, “Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions”, 2006

[39]. Park et al. “PKIS: practical keyword index search on cloud datacenter.” EURASIP Journal on Wireless Communications and Networking 2011 2011:64.

[40]. Sun-Ho Lee and Im-Yeong Lee, “Secure Index Management Scheme on Cloud Storage Environment”. International Journal of Security and Its Applications, Vol. 6, No. 3, Pages: 75-82, July, 2012

[41]. Li, Jin, Qian Wang, Cong Wang, Ning Cao, Kui Ren, and Wenjing Lou., “Fuzzy keyword search over encrypted data in cloud computing.” In INFOCOM, 2010 Proceedings IEEE, pp. 1-5. IEEE, 2010.

[42]. Nikijahan Tamboli, Bharat Savani, Ruchita Choudhari, Nikesh Shah & ApekshaGadkar, “Fuzzy Keyword Search Over Encrypted Data.” International Journal of Computer Science and Electrical Engineering (IJCSEE) ISSN No. 2315-4209, Vol-1 Iss-1, 2012

[43]. Cao Ning, Zhenyu Yang, Cong Wang, Kui Ren, and Wenjing Lou., "Privacy-preserving query over encrypted graph-structured data in cloud computing." In Distributed Computing Systems (ICDCS), 2011 31st International Conference on, pp. 393-402. IEEE, 2011.

[44]. Narayan, Shivaramakrishnan, Martin Gagné, and Reihaneh Safavi-Naini., "Privacy preserving EHR system using attribute-based infrastructure." In Proceedings of the 2010 ACM workshop on Cloud computing security workshop, pp. 47-52. ACM, 2010.

[45]. Li, Ming, Shucheng Yu, Ning Cao, and Wenjing Lou., "Authorized private keyword search over encrypted data in cloud computing." In Distributed Computing Systems (ICDCS), 2011 31st International Conference on, pp. 383-392. IEEE, 2011.

[46]. Vincent, Treesa Maria, and Mrs J. Sakunthala, "Encrypted Data Storage in Cloud Environment." Internationl Journal of Advanced Research in Computer Science and Software Engineering Vol. 3, Issue 3, March(2013), pp 498-506.

[47]. Bedi, Rajeev, Mohit Marwaha, Tajinder Singh, Harwinder Singh, and Amritpal Singh, "Analysis of Different Privacy Preserving Cloud Storage Frameworks." International Jorunal of Computer Science and Information Technology Vol 3, No. 6, Dec 2011

[48]. Li, Ming, Shucheng Yu, Wenjing Lou, and Y. Thomas Hou., "Toward Privacy-Assured Cloud Data Services with Flexible Search Functionalities." In Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on, pp. 466-470. IEEE, 2012.

[49]. Swaminathan, Ashwin, Yinian Mao, Guan-Ming Su, Hongmei Gou, Avinash L. Varna, Shan He, Min Wu, and Douglas W. Oard, "Confidentiality-preserving rank-ordered search." In Proceedings of the 2007 ACM workshop on Storage security and survivability, pp. 7-12. ACM, 2007.

[50]. Madiha Waris, (2013), "Indexing of unstructured data for searchable encryption in cloud environment." E&ME College, NUST, Pakistan.