Indexing of Unstructured Data for Searchable Encryption in Cloud Environment

By

Madiha Waris

2011-NUST-MSPHD-CSE (E)-15

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.Muhammad Zaman Fakhar**during the implementation of this research.

DEDICATION

To my parents and teachers

ABSTRACT

Due to the wide use of cloud services the amount of unstructured data being transferred to the clouds is increasing exponentially. The security threats to the cloud data and probability of data vulnerabilities is also increasing. To minimize the data hacking, misuse or unauthorized use different data encryption techniques have been adopted by the cloud services providers and cloud clients for the data security. With these encryption techniques the data searching becomes difficult. The purpose of this research is to perform indexing on the unstructured data using the original document. The technique will reduce the overhead of decryption before searches. The preprocessed index will be encrypted which will not reveal the identity of the documents or the words present in the document. The decryption is only needed when the relevant documents are returned to the user.The proposed technique achieves the requirement of high security, efficiency and accurate search. It can be used as a secure data retrieval technique by data owners and organizations to minimize security threat and to have efficient search across the cloud server.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This chapter presents the detailed introduction of the thesis. The basic purpose is to introduce the main concepts and description of the terms included in the research.

With the rise in demand of Information Technology (IT) services the petition for enhancing IT architecture, organization and infrastructure is also increasing. A lot of expenditure and time is needed in order to fulfill the enhancement of IT architecture, organization and infrastructure. Cloud computing architecture was established to overcome these business issues such that the all the IT services are outsourced to the customers in comparatively less time and minimum expenditure [1].

1.1. CLOUD COMPUTING

In Cloud Computing the management and facilitation of applications, software, resources, and information as services is done on the internet. 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 [2]. Cloud computing is an internet based model which delivers services based on the internet for all kind of market users comprising government organizations and financial health care markets [3]. The major advantages of the cloud computing model are storage management, minimal cost overhead on software and hardware usage, personnel maintenance, and data access universally without any geographical location restriction [4].

The online services such as email, information accommodation and data storage are becoming important factors in cloud services.Cloud concept advancement for application development is being popular now-a-days for renting various infrastructure services [5]. The cost saving issue is growing to be an important factor in cloud computing domain. According to a Global Information Security Survey 2010, Ernst and Young examined that there were organizations which needed to minimize IT expenditures without sacrificing technological advantages, stating that‘Their interest lies in computing services that require significantly less initial investment, fewer skilled internal IT resources and lower operating costs. As a result cloud computing services are gaining greater adoption[6].

1.2. CLOUD SECURITY

Now-a-days people store a lot of personal and potential data on their computers which is now being shifted to the cloud. The lack of security is actually a barrier for cloud adoption [5]. The information which is stored on the cloud is seemed to be important to the malicious third party and is needed to be secured against this intent. Therefore security measures are needed on both cloud provider and cloud customer level. A provider has more resources to secure the data over the cloud. By subscribing to the cloud, the information or dataon the cloud can be illegally used and destroyed by hackers [7]. The security issues in cloud platform include infrastructure security, data security, application service security and virtualization security [8], [9]. Also privileged access to data, regulatory compliance, data location, data segregation, data recovery, investigative support, long term viability and data availability are the major security issues which may arise while organizations deal with the cloud [10].Therefore there is a need for security implication, as it is the major issue in cloud computing for data confidentialityand data privacy [11]. Some standards for practicing security management in the cloud are ISO/IEC 27001/27002, Information Technology Infrastructure Library (ITIL) andOpen Virtualization Format (OVF) [12].A quantitative security analysis of cloud environment is performed in [13].

1.3. MOTIVATION

With the growth of data size, the need for efficient search also increases. From companywide email systems and large scale data collections, the data search is needed to be efficient and accurate. Majority companywide email systems and large databases are distributed in nature and they may be outsourced to the cloud in the form of data centers. Therefore there is a need to keep this outsourced data secure when distributed.

Along with the data volume increment due to large number of users, the reliability and security of cloud data and services is at stake. Therefore such cryptographic systems are needed through which data may be hidden from the hackers. One of the methods is through searchable encryption to secure potential and critical data on the cloud. The unstructured data on cloud is difficult to search with even searchable encryption techniquesbecause it will be difficult to search huge number of documents containing unstructured data. To provide efficient and accurate search to a large number of users in cloud environmentfor unstructured data become challenging.

1.4. SCOPE OF THE RESEARCH

Before outsourcing of sensitive data to the cloud, it has to be secured by applying encryption technique but on the other hand search andother operative process become difficult to perform. Traditional searchable encryption techniques facilitate users to search over encrypted data set on the cloud servers but these search techniques are not efficient and accurate. Moreover the searchable encryption techniques lack the ability of ranking the documents. These techniques will only search the matches of the keywords in documents.Documents will be returned in the order the matches found.

This research solves the problem of exact word match by indexing the documents before outsourcing them to the cloud server. The search is ranking based and retrieves documents in descending order of their ranks. The documents ranking is a concept of finding the frequency of occurrence of the keywords searched in the document. This is termed as Term Frequency (TF). The code words of the documents words are generated by following different hashing algorithms and security enhancements steps. This result in enhancing the security of data placed on the cloud by making data secure and privacy preserved.

1.5. PROBLEM DEFINITION

The encrypted documentscontaining unstructured data placed on cloud server are difficult tosearch for their contents. To clarify the statement suppose a user ‘A’ savespotential data documents in encrypted form on an unreliable cloud server. If User ‘A’ wants to retrieve all those documents which contain the word ‘cloud’ then in order to retrieve these the user ‘A’ has to first decrypt all the documents and then search the word ‘cloud’ from all decrypted documents. In this case the user has to share the secret key of encryption with the cloud which makes the documents unsecure.If the user does not shares the secret key then all documents needs to be downloaded first and then decrypted to perform search. This process is time consuming and unsecure for searches on cloud server which makes this solution inefficient to search the encrypted documents.

The ideal solution to the above problem and the problem which is to be solved in this research is that the data should be preprocessed and an encrypted index of documents should be created. The index should be secure and efficient for search operations. The encrypted index and the encrypted documents should be uploaded to the cloud server.The encrypted index should contain

distinct words from the documents and the words ranks indicating the frequency of occurrence of the word in some particular document. All searches should be performed on the encrypted index.If match of the keywords are found, only those documents which contains matched keywords should be returned to the user and decrypted on user's demand. During searches the cloud server will have no knowledge of searched keywordsor the documents contents.

1.6. RESEARCH CONTRIBUTION

The contribution to this research is briefly described as follows:

1. Unique distinct words from each document are indexed.
2. The ranking based keyword search technique is proposed.
3. The enhancement in security is done by the use of bloom filters for indexing unique words in the document.
4. Exact word match with punctuation marks is provided.
5. Case insensitivity can also be achieved with minor changes in the implementation.
6. The experimental results proved efficient and accurateresults of the proposed solution.
7. The comparison of proposed indexed search with linear search proved that proposed search is more efficient and accurate.

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.

1.7. BASIC CONCEPTS

1.7.1. CLOUD COMPUTING SERVICES

Various cloud services are available for the customers to run applications. Among service providers, Google is the biggest one and it provides services like Gmail, Google Documents and Google Calendar [14].

Cloud computing services are categorized into three types which are as follows [3], [10], [12],[14], [15], [16],[17]:

1. Software as a Service (SaaS)

2. Platform as a Service (PaaS)

3. Infrastructure as a Service (IaaS)

These categorized services are provided on demand of customer over the internet. These services are described below.

1.7.1.1. Software as a Service

In SaaS, the services and applications to the customers are rented i.e. ready to use applications instead of installing those applications on their computers. The application availability is done over the network by SaaS provider. SaaS applications are shared to multiple customers at a time keeping the fact that each application being shared is unique. The customer's information is kept safe by the SaaS provider. Examples of SaaS applications include web content delivery services, and online word processing tools. Google and salesforce.com act as SaaS service providers [12], [15], [17].

1.7.1.2. Platform as a Service

In PaaS, the services to the customers are offered as a platform for development environment to run their applications. This development environment is actually an Application Programming Interface (API) which is remotely configured. By using PaaS the customers lacking administrative skills can also run their applications because PaaS provides such a deployment platform which involves configuration management and development tools. The customers can easily make their own web applications without installing any tools on their computers [15]. Google App Engine is an example of PaaS by using which Python and Java based applications can be deployed [16], [17].

1.7.1.3. Infrastructure as a Service

In IaaS, abstracted hardware and operating systems as well as virtual machines are offered over the network. The customers are updated when using the IaaS because IaaS provides latest infrastructure technology. GoGrid, Flexiscale and Amazon are the examples which offer this service [12], [17].

The basic cloud environment architecture has been shown in Figure 1.1.

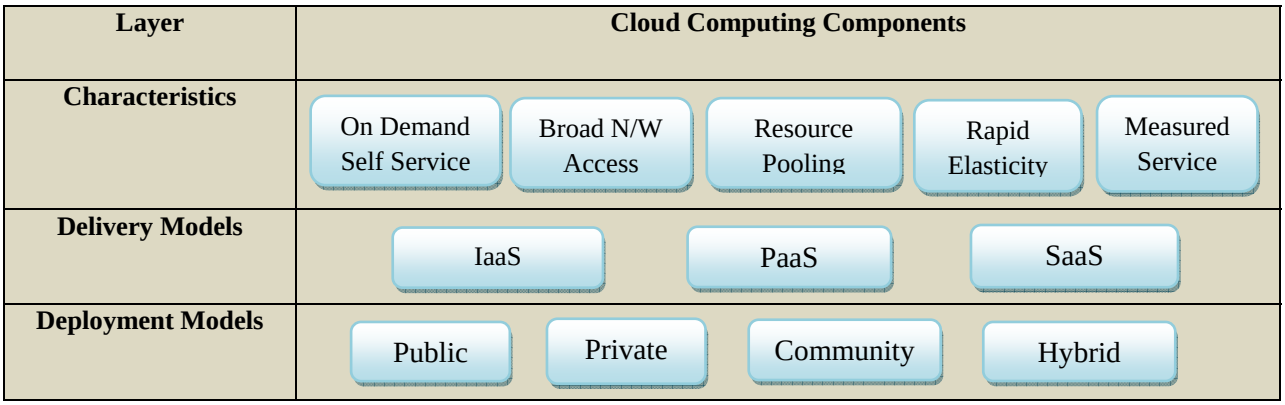

Figure 1.1: Cloud Environment Architecture derived from [18]

1.7.2. CRYPTOGRAPHIC FUNDAMENTALS

Cryptography refers to the art of secret writing. To provide security to the cloud data cryptography isused to transform the data to make it unreadable by the illegal users [19]. It can only be retrieved if the users have a secret key to access that transformed data. All cryptographic algorithms are classified into symmetric and asymmetric algorithms [20]. The main objectives of cryptography are integrity, confidentiality, authentication and non-repudiation.

1.7.2.1. Symmetric Cryptography

In a symmetric key cryptography both parties use the same key for encryption and decryption. This means that the encryption key must be shared between the two parties before any messages can be decrypted. Symmetric systems are also known as shared secret systems or private key systems. Symmetric ciphers are faster than asymmetric ciphers but the requirements for key exchange make them difficult to use [19], [21].

1.7.2.2. Asymmetric Cryptography

Asymmetric Cryptography is also known public key cryptography in which both communicating parties have a set of two related keys [19]. The public key is shared publicly. The private key should never be shared with anyone and is known by the owner only [21].

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.

1.7.2.3. Stream Cipher

Stream cipher is a pseudo random generator which is seeded on a private key. XOR is performed on stream of bits and plaintext input stream. The cipher text is combined bit-by-bit with the

plaintext stream.Therefore if any change occurs in plaintext it will change only the corresponding cipher text bits [21].

1.7.2.4. Block Cipher

Block ciphers convert a fixed sized block of plaintext into a fixed size block of cipher text. For generating cipher textthe block cipher uses all the bits of the input plain text block. Therefore the entire cipher text will be differentiated due to the change in even a single bit [21].

1.7.2.5. Hashing

In hashing 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 will be produced using different hash algorithms. Examples of hash algorithms are SHA-1 (varies from SHA-256 and SHA-512), Tiger and WHIRLPOOL [21].

1.7.3. 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 [22].There are a number of techniques for searching a document which are as follows:

1.7.3.1. Word Sub-Match Search

A substring is searched from the document. For example, when a user searches for a word ‘implement’ it should retrieve the word ‘implementation’ as well.

1.7.3.2. Exact Match Search

This search is the most common form of search in which a document which is searched for the keyword ‘X’ provides the exact matched document which contains the same word ‘X’. In this form of search all kinds of punctuation marks are restricted.

1.7.3.3. Regular Expression Search

This search enables finding a pattern from a given set of documents. Regular expressions are implemented as a typical finite state machine having input symbol from the document to be searched in one at a time. If the goal is achieved by the finite state machine it means the regular expression is satisfied and a match is obtained otherwise not.

1.7.3.4. Case Insensitivity Search

This search can be added in the exact match search by providing the facility of checking case sensitivity that both the searched word and the word in the document are having same character sensitivity.

1.7.3.5. Proximity Based Queries Search

This search provides the approximation results checking whether the searched word say ‘X’ is appearing before or after ‘Y’. If it is present then the exact match is returned based on the approximation.

1.7.3.6. Natural Language Search

This search is based on the concept of stemming. In this search the stop words i.e. ‘the’, ‘and’, and ‘a’ are removed. For Example if the search is done for the statement ‘How do I get Admission’, then this search algorithm will search those documents which include the words ‘get’ and ‘Admission’.

1.7.3.7. Linear Search

For a given query every document is searched from the beginning to the end of the document. For a large number of documents this search is not reliable and is much expensive. The benefit of this search is that there is no initial preparation time which is needed by the document as compared to the Pre-Processed Indexing scheme.

1.7.3.8. Pre-Processed Search

Each unique word of each document is indexed. Whenever the search is performed for a given word the index is checked and if matched then the document is added to the output result document set. The index construction is based on the hashing mechanism [22].

1.7.4. BLOOM FILTERS

From web aching to Peer-to-Peer networks bloom filters are being adopted for IP traceback [23], [24]. Bloom filter is usually located in the main memory for giving fast query response [25]. Certain number of has functions are applied on each member of the bloom filter. The member query speed can be enhanced by minimizing the number of hash functions. Therefore, the rate of false positives increase until the space of efficiency is sacrificed [26].

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 [27].

1.8. THESIS OUTLINE

Chapter 2 presents the literature background related to searchable encryption techniques proposed by various authors. Chapter 3 articulates the proposed technique in detail. 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.

2. LITERATURE REVIEW

This chapter briefly describes the different techniques of searchable encryption by different authors. The main idea achieved in this chapter is to perform analysis on the different techniques discussed in literature. Two main categories of searchable encryption techniques have been discussed following their analysis. The analysis highlights the advantages and disadvantages of the discussed techniques and their implementation details.

Before outsourcing of sensitive data to the cloud it has to be secured by applying encryption techniques. Therefore in this chapter the various techniques have been discussed for data protection, data searching and data indexing.This will be helpful for carrying out current research.

2.1. LINEAR SEARCH TECHNIQUES OVER ENCRYPTED CLOUD DATA

In this section the techniques by different authors forlinear search over encrypted data have been analyzed. This analysis will be helpful in showing the difference between the linear search and the indexed search.

Song et.al [28] described various techniques to solve the problem of searching on encrypted data. The set of techniques by Song et.alprovide a linear search for all documents. The algorithms consumeminimum space. Thesetechniques for searching are secure and provide query isolation.

The aim of the authors is to provide a search mechanism without any loss of data confidentiality. There is a possibilitythat the cloud server attempts to learn the content of data on the cloud server.Song's techniques can be used to support search queries when data users do not want to reveal the contents of data to the cloud server.

Song et.al initially performed sequential scan of a document following a basic scheme for controlled and hidden searches. The technique discussed constitutes two steps:(1) Encryption and (2) Search. In the first step encryption is performed where the numbers of documents to be uploaded are encrypted by the algorithm described below:

S1: Generate three keys: k',k'' and k''' based on a master key.

S2: Encrypt each word of a document using $k''$with a standard block cipher.

$$X_i = E_{k''}(W_i)$$ **Equation 2-1**

S3: Take$x$ bits from the stream cipher based on k''', and$x$ must be less than the length of encrypted word, which will remain consistent throughout the system. These x bits are referred as S_i.

S4: Split the encrypted word in step S2 i.e. S_i into left and right half where length of $L_i$is $x$and length of $R_i$is$X_i - x$.

$$X_i =< L_i, R_i >$$ **Equation 2-2**

S5: Build a word specific key i.e. k_i built by the combination of left half of the encrypted word and key k' before hashing.

$$k_i = f_{k'}(L_i)$$ **Equation 2-3**

S6: Combine $k_i$with S_i to produce bits equal to R_i before hashing.

$$F_{k_i}(S_i)$$ **Equation 2-4**

S7: Obtain cipher text using following formula:

$$< L_i, R_i > \oplus < S_i, F_{k_i}(S_i) >$$ **Equation 2-5**

In the second step Search is performed. To perform search of certain word the first five steps are same as discussed in the first step i.e. inencryption. After that the client has the following information of X and k:

$$X = E_{k''}(W_i)$$ **Equation 2-6**

$$k = f_{k'}(L_i)$$ **Equation 2-7**

After getting the value of $X$and $k$through client by encryption process the server have to perform following steps to carry out complete search:

S1: Encrypt each word of the document with the encrypted search word i.e.X resulting in the following pair:

$$S_i, F_{k_i}(S_i)$$ **Equation 2-8**

S2: Retrieve the bits S_i from the step S3 in encryption process which will depict the actual length of x.

S3: After finding S_i and k, hash them as in step S6 in encryption process and compare using the right part of the pair i.e. $F_k(S_i)$ and check if it matches or not. If it matches return this document to the client.

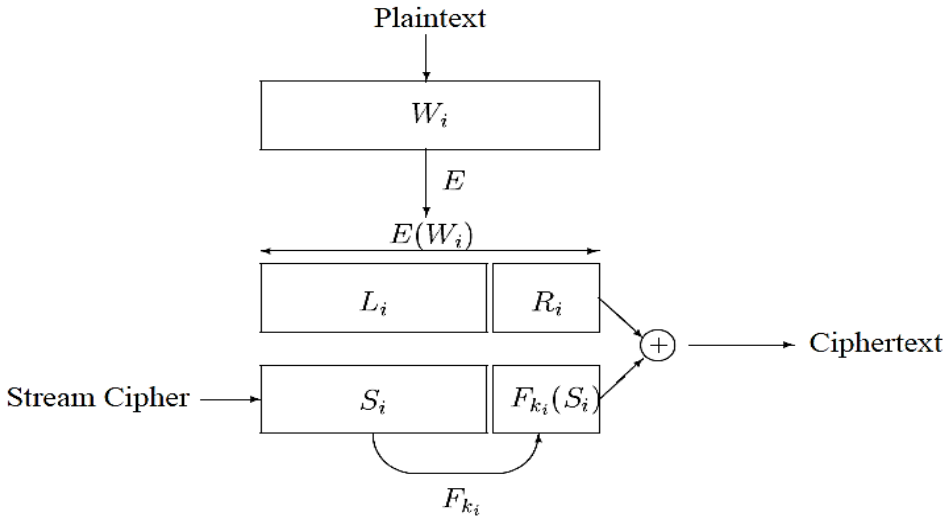

Figure 2.1: Linear Keyword Search derived from [28]

The flow chart of the technique is shown in Figure 2.1. These algorithms described in this technique only need $O(n)$ time for encryption and search. There is no extra space overhead in this technique. Despite of these advantages there are also some limitations i.e. with a large amount of data set this scheme is so much time expensive due to linear search of whole document for each search word. This technique lacks punctuation marks search, case insensitivity, regular expression match and sub matches.

D. Boneh et.al introduced the concept of public key encryption with keyword search for searching keywords from encrypted data placed on cloud [29]. The keyword search proposed by the author is based on linear encryption search and constitutes four basic functions which are $KeyGen$, Tag, $Trapdoor$ and $Test$ function which fulfill the requirement of keyword searching intwo phases. In the first phase the message is encrypted by the sender and the sender generates

some tags for the message by running the Tag function. The cipher text and tags are saved then on the server.

The steps included in first phase are:

- **S1:** $KeyGen(k)$: This step is performed by the receiver. This algorithm takes a security parameter k as input and generates a public/private key pair (A_{pub}, A_{priv}). In addition, the receiver generates the public keyword set$W$.
- **S2:** $Tag(A_{pub}, W)$: This step is performed by a sender. It takes A_{pub} and a keyword W as input and outputs a tag S_w.

In the second phase$Trapdoor$ function is run by the receiver for generating trapdoors for each keyword.The trapdoors are then sent to the server where the function $Test$ is executed to search from the tags attached to each encrypted message.

The steps included in second phase are:

- **S1:** $Trapdoor(A_{priv}, W)$: This step is performed by the receiver. It takesA_{priv} and a keyword Was input and outputs a trapdoorT_w.
- **S2:** $Test(A_{pub}, S_w, T_{W0})$: This algorithm is run by the server. It takes $A_{pub}, S_w \; and \; T_{W0}$ as input, and outputs 1 if $W = W_0$and 0 otherwise.

The proposed technique by D. Boneh et.al is only for tagging messages with a few keywords that can be searched. It does not scale to searches over the entire document. The performance factor is a problem in this research as the public and private keys are generated from a set of other number of different input words. For the generation of usable keys a large number of prime numbers have to be calculated in public key algorithms. Hence this process requires a lot of time during public key generation.Only exact match search has been provided in this scheme. Case-insensitivity, natural language search, sub-match search could also be added.

Koletka and Hutchison [30] described a solution which enables the users to store the data securely on public cloud. The proposed solution facilitates searching of the system through user's encrypted data input. Data confidentiality, data integrity and file sharing has been maintained by the proposed solution. The system contains two components which are a Client

side application and the Server application. The data security operations are performed by the Client side and the Server application handles the retrieval of data from the storage service and encrypted queries.

The system satisfies the requirements of confidentiality of data storage, integrity of data, creation of file sharing, key revocation allowance, searching, recovering from compromised key pair and access control to the server.

The basic architecture of this technique is comprised of two components: a Client application through which security operations are handled and a Server application through which keyword searching is handled. The system has been defined by these steps: Client Application, Server Application, Authentication Protocol, File System Operations and Searchibility operation.

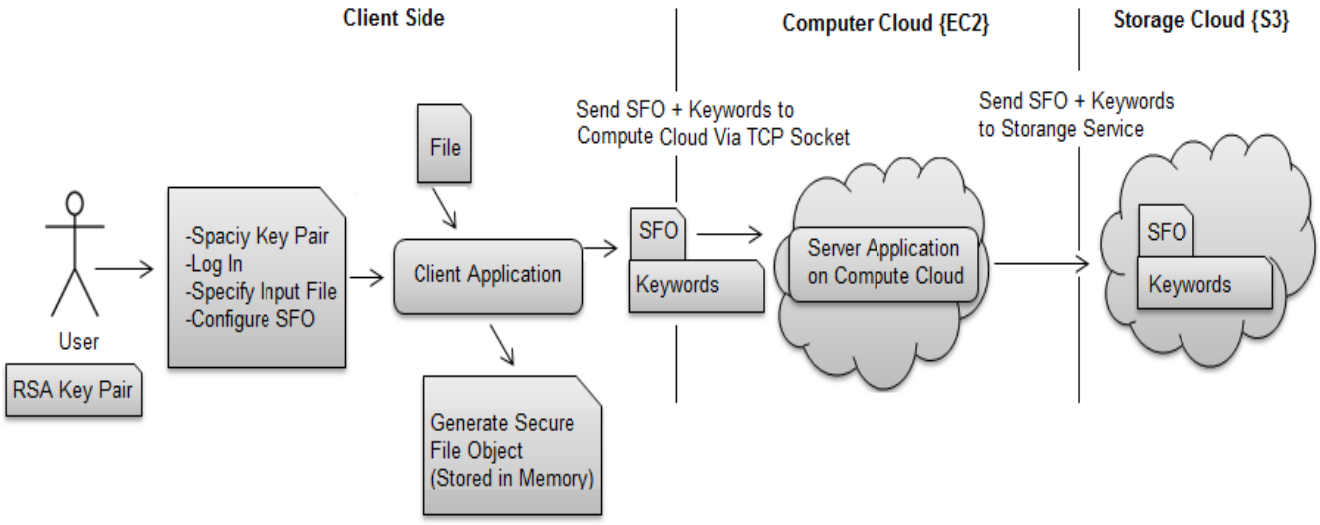

Figure 2.2: Searchibility Operation Architecturederived from [30]

Among all the operations in the architecture as shown in Figure 2.2the Searchibility operation is themost important step and is defined as follows:

In the Searchibility operation the user at first submits the encrypted keywords to the server on which the cryptographic functions are performed by the server.The results are returned to the user if the query is matched. The cryptographic functions are performed over the files which are stored in the storage service. The Secure File Object (SFO) contains all the data i.e.File name,

owner id, modified user id, read list, public key list, encrypted file encryption key, encrypted keywords, encrypted data digest and secure file object digest.

After the generation of searchable keyword which cannot be retrieved originallyan encrypted keywords field has been attached to each SFO. The encrypted keyword field is encrypted using symmetric algorithm with the file owner's secret key due to which only the searchable keywords are visible to file owner. The searchcomprises two steps: (1) Secure keyword generation and (2) Keyword Search.

In Secure Keyword Generation, the keyword to be searched should have a $flag$and a random bit string 'r' stored with it in SFO. For each keyword the server performs the computation as shown in the equation below where $W_i$is $i-th$order from the list of keywords, $H_s$is a hash function which has been keyed with the secret$S$. $padding$is some random bits. $c_i$is the keyword which has been encrypted and used by the server to check which $c_i$matches the give search capability.

$$a_i = H_s(W_i), \quad b_i = H_{a_i}(r), \quad c_i = b_i \oplus (flag \,|padding) \qquad \textbf{Equation 2-9}$$

$c_i$is stored in the storage server withW_i. After the generation of encrypted file a list for a SFOsis uploaded on the cloud server.In the second step users may submit the search capabilities to receive search results. These search capabilities are generated only by the owner of a file because a secret key $S$is needed which is known by the authenticated user only. When user searches a keyword$W$, the owner generates it by following formula:

$$d_w = H_s(W) \qquad \textbf{Equation 2-10}$$

The user then gains this search capability from the owner and sends it to the server. The server performs certain operations on each file which are as follows:

$$p = H_{d_w}(r) \qquad \textbf{Equation 2-11}$$

where 'r' is saved with the encrypted keyword list.

For every $c_i$in the encrypted keyword list the computation done by server is:

$$x = p \oplus c_i \qquad \textbf{Equation 2-12}$$

The match occurs for a current file.When the first l bit of $x$matches the flag the file name is returned to the user as a search result. For every file this process is repeated in the storage service.

Since all the data is encrypted the user's data is secure enough to handle any kind of security breach on cloud. Illegal data access becomes very difficult on cloud servers. The analytical resultproves that the storage overheads continue to remain same even when the file size is increased.

2.1.1. ANALYSIS

A brief comparison of the above discussed techniques is given below for emphasizing on the effectiveness of the proposed techniques for linear search over encrypted cloud data.

The techniques proposed by Song et.al [28]solves the problem of searching over encrypted cloud data but as the amount of data set increases the search time increase. The reason for time increment is the linear scan of documents for searching. This technique does not support punctuation marks, sub-matches, caseinsensitivity and regular expressions.

D. Boneh et.al [29] has proposed the technique of public key encryption with keyword search for searching keywords from encrypted data placed on cloud. Only the tagged keywords are searched in this technique whereas search on complete documentsis not possible. The main problem identified in this technique is time consumption during public key generation. This technique lacks sub-match search, case-insensitivity, regular expressions search.

Koletka and Hutchison [30] have given a solution to the keyword search problem resulting data security on the cloud server.This technique can handle large number of files efficiently.But security overhead has been analyzed in this research. No performance evaluation for searching of a keyword has been done. This technique only provides single keyword search limiting search functionality.

2.2. INDEXING BASED SEARCHING TECHNIQUES OVER ENCRYPTED CLOUD DATA

In this section the techniques by different authors for indexed based search over encrypted data have been analyzed. This analysis will be helpful in finding the effectivenessof the indexed search.

N. Cao et.al [22] dealt with the problem of constructing a searchable encryption system by using secure ranked search. The searchable encryption system has been produced by facilitating the relevance ranking of the search results and achieving only the accurate retrieval of the results. For building secure searchable ranked index the statistical measure approach has been used i.e. relevance scoring and less information leakage about relevance scoring for keywords privacy.

The motivefor secure ranked keyword search is that traditional searchable encryption techniquesproposed in [28], [29], [31], [32], [33] allow keyword search through conventional Boolean search without checking the relevant documents during search. The drawback in those techniques is when they 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. And the document retrieval accuracy was a big issue in traditional methods.

The problem identified is that thedata owner hasall documentswhich need to be outsourced on the cloud server after encryption. Before outsourcing the data to cloud server secure searchable index need to be created from keywords taken from the data owner documents. 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 back those documents in which keyword is matched.

To solve the problem of ranked search an inverted index structure has been proposed. To find the most relevant documents against a keyword, the numerical score assignment has been done on the basis of ranking function. 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. To calculate the relevance scores of the documents TF and IDF rule is used. ‘TF’ is the term frequency which is a numerical value showing the number of times a keyword is appearing in a document. ‘IDF’ is the inverse document frequency which his

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 scheme proposed for ranked searchable encryption comprises two phases which are (1) Setup Phase and (2) Retrieval Phase. The Setup phase is completed by two algorithms i.e. Key Generation and Build Index.

S1: The completion of this phase is the responsibility of data owner who executes the 'Key Generation algorithm' and with this execution the public and private key parameters are initialized.

S2: The Build Index is run to extract unique wordsfrom the documentsfor the creation of searchable index.

The owner encrypts the documents and outsources them to the cloud server along with the indexes containing relevance scores in encrypted form. Indexesare encrypted using Order Preserving Mapping. Through broadcast encryption the owner also distribute the secret parameters to a group of authorized users.

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

S1: The user generates the trapdoor against a specific keyword using Trapdoor Generation algorithm and delivers it to the cloud server.

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

The positive aspect of this technique is that the ranked keyword search 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.

This research proved mathematicallythe proposed technique and no implementation proofs have been specified. No implementation scenarios have been given to help Data owner for implementationof the searchable encryption system.

Curtmola et.al [31] revised the previous work ofindex based searcheson cloud servers and proposed an improved search technique. The contributions are to review the existing security definitions and to highlight the shortcomings, to introduce two new adaptive and non-adaptive adversarial models for encryption, to present two new constructions to prove the security of new definitions and to allow the multi user setting instead of single user setting.

The proposed scheme by Curtmola et.al is constructed on the basis of combination of a lookup table (T) and array(A). A linked list is generated for saving the list of document identifiers in which the word is found. After creation of linked lists they are encrypted and before encryption a key is associated with each element in the list to encrypt the next element in the list. This techniquecomprises four steps which are (1) $Keygen$, (2) $BuildIndex$, (3) $Trapdoor$and (4)$Search$. These steps have been described as follows:

S1: For $ygen(k)$,three random keys are generated based on a security parameter. These keys combine to form a private key K.

$$K = (s, y, z) \qquad \textbf{Equation 2-13}$$

S2: For $BuildIndex(k, i)$ document index is built using a given key in this step. The distinct words are collected from the documents. An array (A) is constructed.For each unique worda list of documents is created containing that word.A key is generatedto encrypt the 0^{th} element in the documents. For each element in the documents which contains the word following operations are performed:

1. For this particular element a key is generated to encrypt the next node in the set.
2. A tuple containing document identifier, the key to next tuple and the index of the next element in the list is generated.
3. The tuple is inserted into array (A) before increment of the counter.
4. After array construction the lookup table is constructed this contains the information related to the value associated with every word in the document set which is as follows:

$$< addressof(A[N_0]), k_{i_0} > \quad \textbf{Equation 2-14}$$

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

$$value = < addressof(A[N_0]), k_{i_0} > \oplus f_y(w_i) \quad \textbf{Equation 2-15}$$

The above value is stored in the lookup table by the using a pseudo-random function$F$, combined with the key$z$.

$$T[F_z(w_i)] = value \quad \textbf{Equation 2-16}$$

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

$$T_w = (F_z(w), F_y(w)) \quad \textbf{Equation 2-17}$$

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

$$value = T[T_w\downarrow_1] \quad \textbf{Equation 2-18}$$

The XOR is performed on second part of the trapdoor and the above value to get the tuple having following information:

$$< addressof(A[N_0]), k_{i_0} > = value \oplus T[T_w\downarrow_2] \quad \textbf{Equation 2-19}$$

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

This technique is faster than the previous ones. Constant search time algorithm has been provided for a large dataset.Mainproblem for implementing this scheme is a need to update the array(A) and trapdoor$T_w$whenever the document is added or removed. The document index size increases linearly as the document size.

Park et.al [34] described the two approaches (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 definition and analysis of group search security and keyword index search security has been given. The efficient performance of proposed encrypted database has been calculated. The scheme for keyword index search security has been analyzed in this research to provide safe and secure search which is without re-encryption of all the documents.

The basic technique proposed in this research for having efficient searching in cloud environment comprises steps: $SysPara$, $KeyGen$, $IndGen$, $DocEnc$, $TrapGen$, $Retrival$ and Dec. These steps are explained as follows:

- **S1:** The$SysPara(1k)$ takes a security parameter $k$and outputs a system parameter 1. 1Indicates the elements to set out the encrypted database system.
- **S2:** The $KeyGen(1)$takes 1 as an input, and generates user's group session key set$\{gk\}$, index generation key set$\{ik\}$ and document encryption key set$\{dk\}$.
- **S3:** The $IndGen(ik\ , W)$takes input index generation key set $ik$and a keyword set$W$. Output is index list table.
- **S4:** The $DocEnc(dk, D)$ encryptsdocument using encryption key $dk$and a document$D$.
- **S5:** The $TrapGen(w, ik)$takes a keyword $w$and index generation keyik. Trapdoor T_w is obtained by the encryption of keyword $w$and index generation key ik.
- **S6:** The $Retrival(T_w)$takes trapdoor T_w as an input. If trapdoor is present in the index listthen the output is obtained in the form of encrypted documents that map the identifiers of the matching values in the index list table.
- **S7:** In $Dec(E(D), dk)$the document is decrypted by document encryption key $dk$and encrypted document$E(D)$.

In above technique the indexes are not secureand 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 in [35] proposed a scheme for searchable encryption by which user can share data with others securely by 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.

For search the user generates the trapdoor of keywords witha secret key. The decryption of data is performed only by the legitimate users.

The advantages of using this technique are data confidentiality, quick search speed, and efficiency in communication volume due to only one round check for communication process. On the other hand there is no technical and experimental proof for the validation for this technique.

2.2.1. ANALYSIS

A brief comparison of the above discussed techniques is given below for emphasizing on the effectiveness of the proposed techniques for indexingbased searching techniques.

N. Cao et.al [22] proposed the solution to keyword search problem based on secure ranking. The proposed technique lacks implementation details as only mathematical proofs have been given for finding results.

Curtmola et.al [31] revised the previous work ofindex based searcheson cloud servers and proposed an improved search technique. No results have been discussed in this research. Only the theoretical details have been given. No analysis is performed on security and performance of keyword to be searched.

Park et.al [34] defined the processes of efficiency and searching in 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 [35] gave a searchable encryption technique by generating index. There is no experimental evaluation of the results obtained by the implementation of this technique. Only the theoretical evaluation has been done.

3. PROPOSED MODEL

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

The encrypted data placed on the cloud is difficult to search efficiently. Search is performedafter first downloading all the documents from the cloud server and then decrypting. This is the case if secret key for encryption are not shared with the cloud server. Therefore to avoid huge search time consumptionsuch a technique is needed which provides fast data retrieval over the cloud server.The searching should be performed in such a way that the server does not know any information about the original data to be retrieved by the user.The questions to be answered in this research are as follows:

1. How to Index unstructured data for searchable encryption in cloud environment?
2. How Searchable Encryption can be achieved in an extremely efficient fashion i.e. fast search?
3. How the index based search is better than linear search on encrypted data?
4. 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 achieving fast, efficient, secure and accurate data retrieval in this research keyword based searching technique has been proposed so that the users can selectively retrieves the documents of their interest. For this purpose the data before sending it on the cloudindex generation is performed for all documents and then the documents are encrypted and sent to the cloud server.Ranking based search has been provided which returns the documents in descending order of their ranks i.e. the documents when retrieved are in ranked order depending on the frequency of occurrence of the words the document contains. The documents having words with high frequency are on top. The secureindexing technique is implemented by using the concept of

bloom filters and different hashing algorithms. The code words of the document words are generated by different hashing algorithms by following security enhancement steps.

During search the server has no knowledge of searched keywordsor the documents contents. The documents should not be decrypted during the search process.The purpose of this research is achieved by presenting a secure ranking based indexing scheme for enabling efficient search on encrypted data placed on cloud server without information leakage to the untrusted cloud server.

3.2.1. SEARCHING ENCRYPTED DATA USING BLOOM FILTERS

To do the searchable encryption for unstructured data the proposed technique uses the concept of bloom filters. The basic strategy of bloom filters is discussed in this section. Bloom filter is a data structure which is used for fast set membership test with the possible false positives. A Bloom filter is stored as an array of bits. All the bits are initialized to '0'. After the addition of an element different hash functions are performed on that element. The input to each hash function is the element to be added in the array and the output from each hash function is the index into the array which is different for each hash. After the calculation of the hash algorithms each bit in the filter at the indexes which are specified by the hash outputs is set to 1. As an example in the Figure 3.1 the bloom filter using three hash functions as shown.

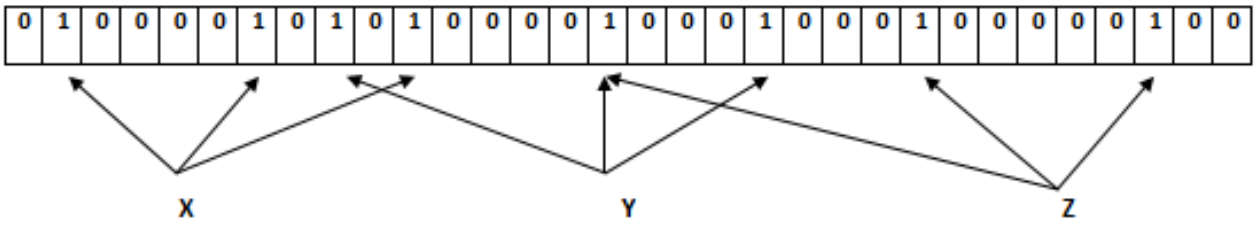

Figure 3.1: Bloom Filter using three Hash Functions

In Figure 3.1 three words X, Y and Z have been added as members to the array in such a way that three hash functions have been applied on each of the word.Two hash outputs may result in the same bit index and in that case the bit which is already set to 1 remains as it is e.g. in the figure the hash output of two different hashes from Y and Z are resulting in the same bit index.

For test case to check the membership of an element that whether it is present or notthe element is hashed using the same algorithms and the resultant output bits are joined by conjunction so

that the element is considered a member only if all the bit values from all hash outputs are resulted as ‘1’. If any of the resultant bits of hash output is ‘0’ it means the element is not stored in the bit array.When the saturation level is increasedfor elements being addedthe overlapping of elements start increasing. When all the bits specified by the hash outputs are set to 1 the false positives can occur even when the element wasn’t originally added to the filter.

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 [27].

After understanding the bloom filter concept the proposed framework is described in the next section 3.3.

3.3. PROPOSED FRAMEWORK

The basic scheme for proposed model constitutes two phases which are:

S1: Index Generation

S2: Keyword Search

The proposed framework constitutes three entities which are a data owner, data user and cloud server. In Index Generation the basic needs are that the data owner has all the documents which are needed to be outsourced on the cloud server in the encrypted form. Before outsourcing the data to cloud server a secure searchable index is made from distinct words taken from the documents. This process is done with the help of hashing algorithms and bloom filters.

In Keyword Search when a data user wants to search the document containing a specific keyword a trapdoor of the searched keyword is generated andtheir codewords are generated.Then a match is checked in the document index. If codewords are found in the documentsindexes then document id’s matching those codewords are returned. The id’s are in descending order of their documents ranks.

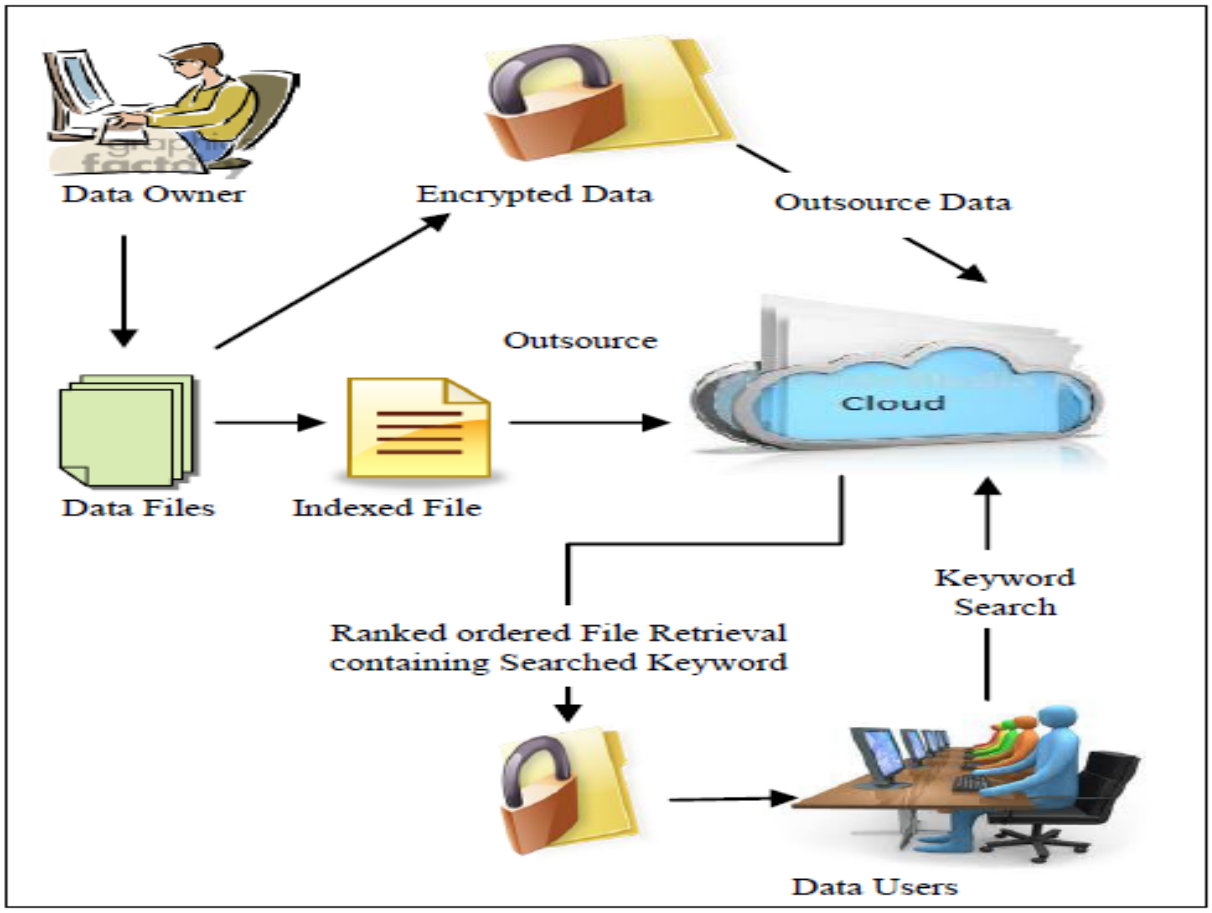

Figure 3.2: Proposed Framework

Figure 3.2 depicts the basic framework of the proposed system which shows the process flow to outsource and search encrypted documents over the cloud. The framework shows the basic idea behind the searchable encryption proposed in this research. It illustrates that the documentswhich are the authority of the data owner are indexed first and then encrypted. Then the encrypted index and encrypted document is sent to cloud server. During search the keywordsare searched in the encrypted index and if there is a match then the documents containing the search words are sent to the data user based on their ranks. The server is unaware of any kind of information placed by the owner or retrieved by the user from the cloud server achieving data security and confidentiality. Since the distinct keywords are preprocessed before sending them to the cloud server therefore a constant search time is achieved during search.

3.4. PROPOSED TECHNIQUE

3.4.1. WORKING SCENARIO OF PROPOSED TECHNIQUE

In the proposed techniquefor creating a document index trapdoors of the distinct words are created which are sent to the bloom filter which produces the codewords for trapdoors. Each word of a document is represented by a codeword. The proposed scheme consists of two phases which are Index Generation and Keyword Search.

3.4.1.1. Index Generation

The steps performed in Index Generation are as follows:

3.4.1.1.1. Collection of distinct words

When Data owner wants to outsource a document on the serverthe document is taken as an input and a set of distinct words are collected from the document.

3.4.1.1.2. Stop Words Elimination

During collection of distinct words from the document stop words are removed in the proposed technique. Stop words are those words which do not serve to add meaning to a sentence but they only help to complete a sentence. Examples are ‘the’, ‘a’, ‘an’, ‘because’, ‘must’, ‘my’, ‘not’, ‘of’, and ‘and’. By the removal of these words only meaningful words are indexed for searching.

3.4.1.1.3. Case Insensitivity

Case insensitivity can be achieved by minor changes in the proposed technique for which each word before indexing is first converted into lower case such that all the distinct words are stored in lower case on the cloud. When certain keywordsare searched from the user side these keywords are first converted into lower case and then matched. In this way the proposed technique can support case insensitivity.

3.4.1.1.4. Master Key Generation

A Master Key is generated to be used throughout the process of indexing and searching. Master Key generation process uses a security parameter i.e. a password. The key is generated by taking the Hash of given password i.e. $SHA512$of the security parameter.

$$M_k = SHA512(password) \quad \textbf{Equation 3-1}$$

$M_k$is then split into a set of eight keys which are used throughout the scheme for trapdoor generation.

$$M_k: k_1, k_2, k_3, k_3, k_4, k_5, k_6, k_7, k_8 \quad \textbf{Equation 3-2}$$

3.4.1.1.5. Trapdoor Generation

A trapdoor is a conversion of the term being searched for such that an untrusted server can find potential matches without gaining the knowledge of the plaintext. Trapdoor generation takes place by the concatenation of each of eight keys with the unique distinct word from each document and applying hash on it. This process repeats for all eight keys generated in 'Master Key Generation' step and results obtained are coma separated. The end result obtained is a Trapdoor.

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

3.4.1.1.6. Codeword Generation

The trapdoor value for each unique word from each document is sent to the bloom filter where crc32 algorithm is used to get bit positions where the trapdoor is to be saved.The whole point of a crc32 is to hash a stream of bytes with as few collisions as possible. Depending on the security requirements and false positive rate of the bloom filter the number of bit positions which combine to show a trapdoor can be varied. Currently we are using five different hash algorithms to apply on the trapdoor to have five bit positions to represent a single trapdoor codeword.Each trapdoor for a word is saved at five different bit positions and these bit positions are combined to form a codeword.

$$codeword = 5\ hash\ algos\ (Trapdoor) => 5\ bit\ postions \quad \textbf{Equation 3-4}$$

3.4.1.1.7. Data Outsourcing to Cloud Server

Before sending data to the cloud server the original documents are encrypted.For each encrypted document its encrypted name, document id, rank of each word and codewords for each wordsare sent to the cloud databasealong with encrypted document by giving document a new id.As a result the cloud database contains the encrypted indexof the document.

3.4.1.2. Keyword Search

The steps performed during search are as follows:

3.4.1.2.1. Keyword Trapdoor Generation

When a user searches a keyword it is at first concatenated with each of the eight keys and hash is applied on them such that the following operation is performed:

$$Trapdoor: f(k_1 + keyword), f(k_2 + keyword) \dots f(k_8 + keyword) \quad \textbf{Equation 3-5}$$

The eight keys used in above equation to be concatenated with the keyword are same which were produced during Key Splitting in Index Generation step. These keys remain the same throughout the process.

3.4.1.2.2. Codeword Generation of Keyword's Trapdoor

Codeword Generation is done from the trapdoor by the same method used in Index Generation phase at Owner level. It is done in such a way that the trapdoor obtained for searched keyword 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.

$$codeword = 5\ hash\ algos\ (Trapdoor\ for\ keyword) => 5\ bit\ postions \quad \textbf{Equation 3-6}$$

3.4.1.2.3. Codeword Outsourcing to Cloud

When the codeword generated for the keyword is sent to the cloud to fetch the relevant documents which contain this keyword it performs following steps:

S1: The codeword is searched in the document indexes placed in the cloud database.

S2: If there is a matchthe ranking of the search word is checked.

S3: The documents id's containing that search word are returned to the user in descending order of their rankings i.e. Top ranked comes first.

3.4.2. GRAPHICAL REPRESENTATION OF PROPOSED TECHNIQUE

In this section the flow of the proposed model has been shown graphically for both of the Index Generation andKeyword Search.

3.4.2.1. Graphical Model for Index Generation

The graphical model of the Index Generation step is shown in Figure 3.3. It shows the overall flow of the Index Generation step. At first the Key Generation process takes place. The values of the keys will then be used for concatenation with the words and hence generating Trapdoor

through hashing. The trapdoor is then sent to bloom filter where further hashes are applied to generate codewords which are then sent to the cloud server.

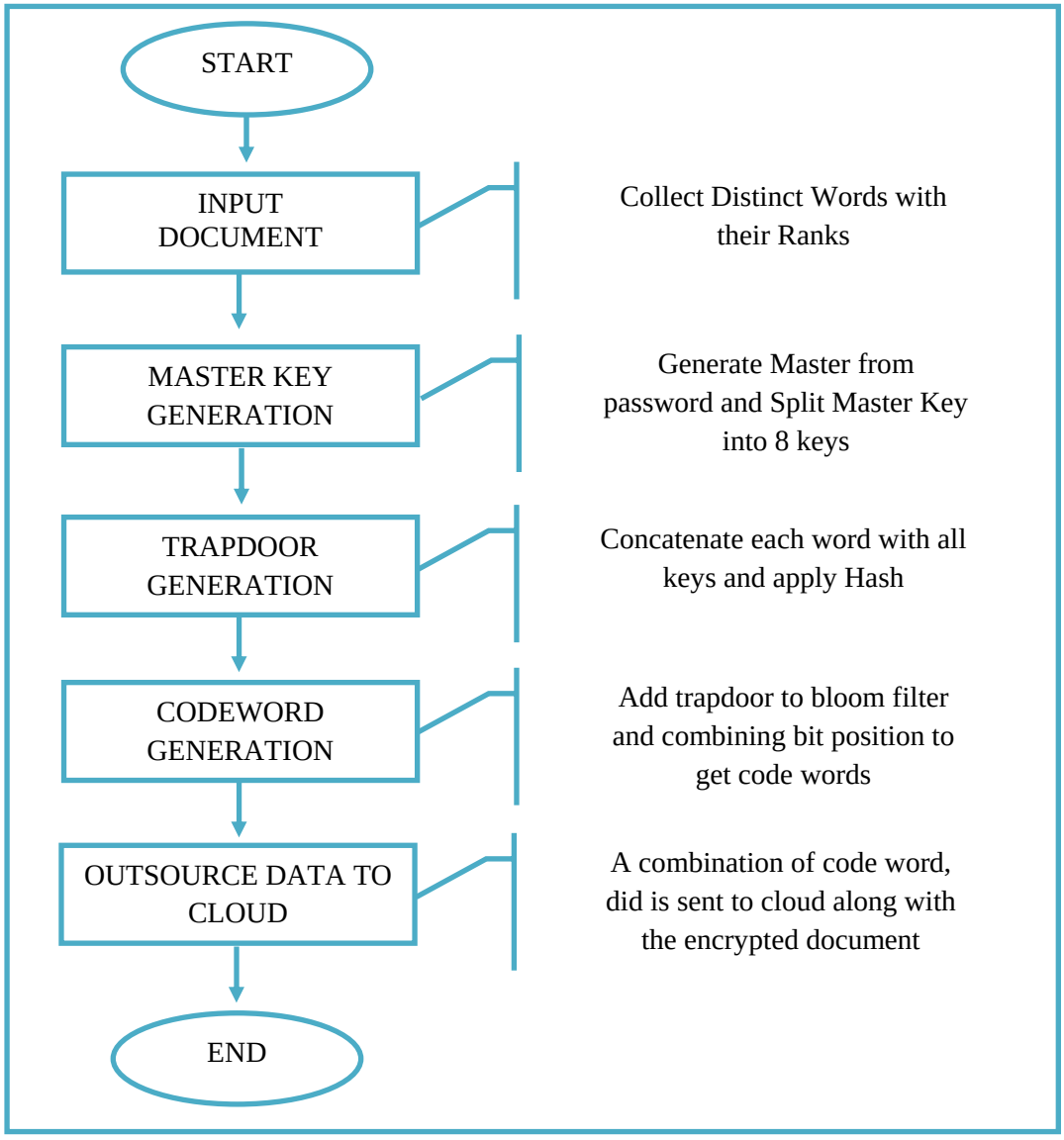

Figure 3.3: Graphical Representation of Index Generation

3.4.2.2. Graphical Model for Keyword Search

The graphical model ofKeyword Search is shown in Figure 3.4. It shows the overall flow of Keyword Search step.For search the process continuous up to ‘Generate Codeword’ step shown in Figure 3.3 and then further the searching is performed to check whether the codewords have a match or not in ‘Check Codeword Match’ step as shown in Figure 3.4.If there is a match the rank

of the keyword is checked in the whole data set in 'Rank Check' step. On the basis of keyword rankings the documents id's are returned to the user.

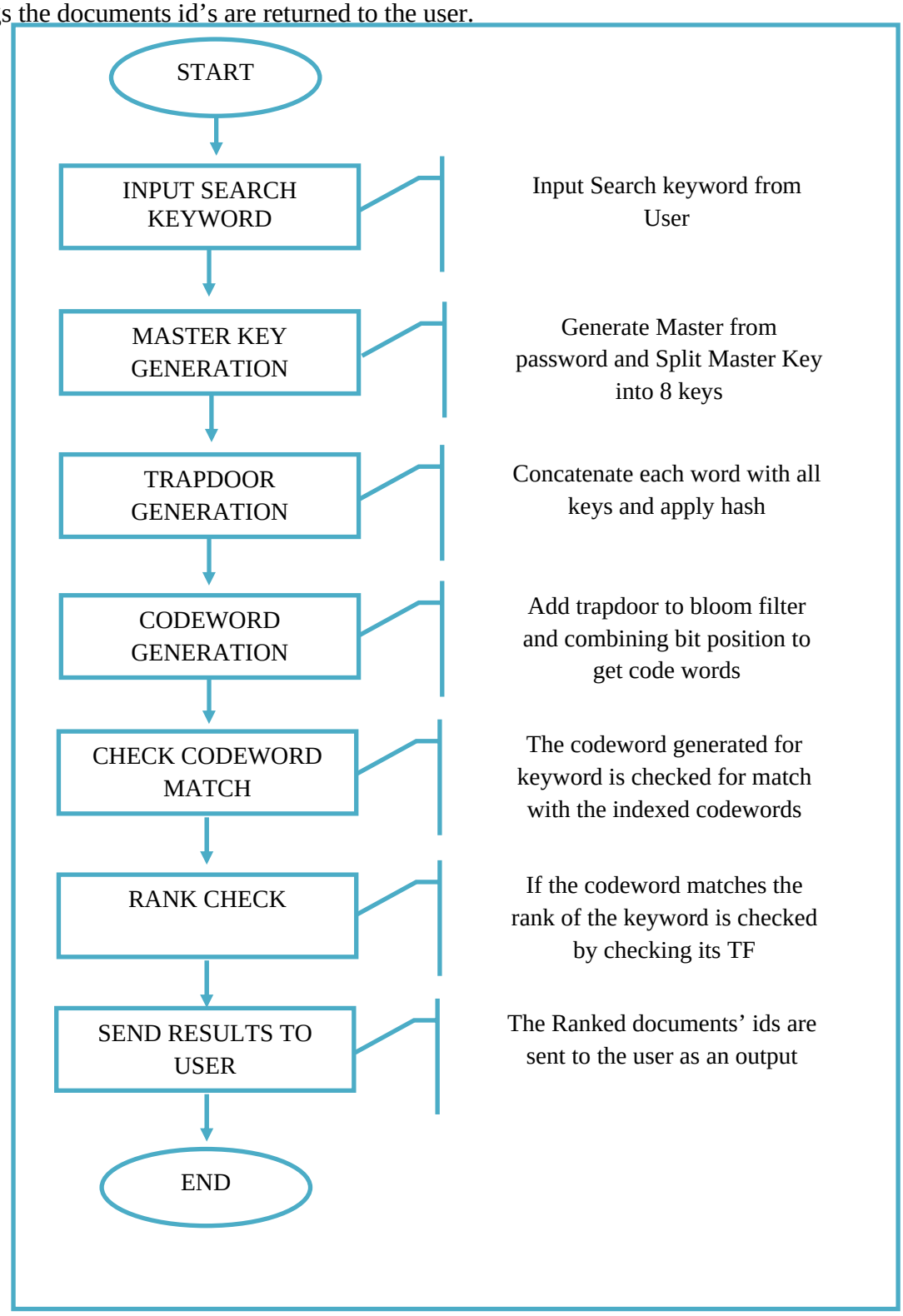

Figure 3.4: Graphical Representation of Keyword Search

3.5. ADVANTAGES OF PROPOSED TECHNIQUE

The advantages of the proposed technique are as follows:

1. The indexing of unstructured data before sending to the cloud server providesfaster search as compared to the sequential linear search of encrypted documents. For very large documents the search time reduces comparatively.

2. The proposed indexing technique for searching over encrypted cloud data is suitable for the applications where the number of users increase for searching data.

3. Exact Match search is supported.

4. Natural Language Search is supported.

5. The searched keyword document contentson the cloud server are secure. The cloud server is unaware of any information placed on server. This enhances the security of data and its confidentiality.

6. The index generation increases the size on disk to store index of the documents but this space is less as compared to the space required for encryptionand decryption ofdocuments to be searched bylinear search technique.

3.6. LIMITATIONS OF PROPOSED TECHNIQUE

The limitations of the proposed technique are as follows:

1. This scheme provides exact-match search but it is unable to handle sub-matches, sentence based searching and regular expressions.

2. The major drawback is that whenever change is required in the documents the updating of indexes is needed. Hence storing an updating the index is a overhead.

4. IMPLEMENTATION

Thischapter elaboratesthe implementation details and achievement of the requirements of the proposed technique.

4.1. IMPLEMENTATION ENVIRONMENT DETAILS

Environmental details constitute the system used for implementation of the real time working environment of the proposed technique including software and hardware specification, interfaces and data.

4.1.1. SYSTEM SPECIFICATIONS

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

Table 4.1: Software Specifications

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 Requirements for Proposed System

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 PROPOSED SYSTEM

The implementation of the indexing unstructured data for searchable encryption over the cloud includes following stages.

4.2.1. COLLECTION OF DISTINCT WORDS

To outsource a document on a cloud serverthe document is sent as an input and the output is obtained in the form of a set of distinct words which are stored in an array. The distinct words are ranked according to the frequency of their occurrences in the document.

Table 4.3: Collection of Distinct Words

Collection of Distinct Words	
Input	Document
Output	A set of distinct words in an array with their ranks
Results	Array: ([International] => 8, [Services] => 16, [what!] => 1, [what] => 1, [This] => 4)

In Table 4.3 the results obtained after applying operation on the input document have been shown. The array contains the useful words indexed along with their ranks e.g. the word ‘International’ occurs eight times in this document which is taken as an input to distinctively separate the unique words.

4.2.2. STOP WORDS ELIMINATION

For collecting distinct words from the document to be uploaded on the server the stop words from that document are removed. For each word it is checked if it is a stop word or not. If it is then it is eliminated. The list of stop words which have been eliminated is shown in Table 4.4.

Table 4.4: Stop Words Elimination

Stop Words Elimination	
Input	All distinct words from document
Output	Distinct words without stop words

Stop Words List	List of Stop Words eliminated= ('able, about, across, after, all, almost, also, am, among, an, and, any, are, as, at, be,been, but, by, can,could, did, do, does, either, else, ever, every, for, from, get, got, had, has, have, he, her, hers, him, his, how, however, i, if, in, into, is, it, its, just, least, let, like, likely, may, me, might, most, must, my, neither, no, nor, not, of, off, often, on, only, or, our, own, rather, said, say, says, she, should, since, so, some, than, that, the, their, them, then, there, these, they, this, tis, to, too, was, us, wants, was, we, were, what, when, where, which, while, who, whom, why, will, with, would, yet, you, your')

When any of the stop word from the list of stop words shown in Table 4.4 is searched by the user no result is shown in the list of retrieved documents.

4.2.3. CASE INSENSITIVITY

Case Insensitivity can be achieved with minor changes in the implementation. The words before indexing are converted into lower case and similarly during search operation the keywords searched are converted to lower case first and then searched.When user is searching for upper case letter it gives same result as in case of searching lower case letter. The input is an upper case letter and the output obtained is lower one as shown in Table 4.5.

Table 4.5: Lower Case conversion

Lower Case Conversion	
Input	The input word, Searched keyword
Output	Lower case input word, Lower case searched word
Results	[International]=> [international], [CLOUD]=> [cloud]

4.2.4. MASTER KEY GENERATION

For creation of master key a security parameter in the form of a password is taken as an input and the hash algorithm i.e. SHA512 is applied on the password. As a result a master key is obtained as shown in Table 4.6.

Table 4.6: Master Key Generation

Master Key Generation	
Input	A security parameter i.e. password
Output	Master Key
Results	b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5e07394c706a8bb980b1d7785e5976ec049b46df5f1326af5a2ea6d103fd07c95385ffab0cacbc86

4.2.5. SPLITTING OF MASTER KEY

The master key generated in Table 4.6 is split into set of eight keys which are saved as distinct keys in an array.

Table 4.7: Key Splitting

Key Splitting	
Input	Master Key
Output	Eight distinct split keys
Results	Array: ([0] => b109f3bbbc244eb8, [1] => 2441917ed06d618b, [2] => 9008dd09b3befd1b, [3] => 5e07394c706a8bb9, [4] => 80b1d7785e5976ec, [5] => 049b46df5f1326af, [6] => 5a2ea6d103fd07c9, [7] => 5385ffab0cacbc86)

In Table 4.7the master key split into eight keys has been shown as saved in eight index positions of an array. The keys have been divided equally.

4.2.6. TRAPDOOR GENERATION

The eight keys generated in Table 4.7a distinct word from the Table 3 and a hash function i.e. SHA1 serve as input for trapdoor generation. The output result has been shown in Table 4.8. The

process flows in such a way that each of the eight keys is concatenated with the specific distinct word and after concatenation hash is applied on it.

The result of word with eight different keys after hash are concatenated by putting coma between then to obtain final trapdoor of the word as shown in table below.

Table 4.8: Trapdoor Generation

Trapdoor Generation	
Input	word, keys and hash function (SHA1)
Output	Trapdoor
Results	Trapdoor = SHA1 (b109f3bbbc244eb8+word), SHA1 (2441917ed06d618b+word)... Trapdoor=7c901d046c7409aaa2177928d9c8fbd733d18953,88baa93ed3f4ebc1ad78c8d4bf779c00b8264a00,d7b93647b6f05461490d47a38be6aa00ed2da393,5d0a04f8c5702ba9b3041e01138f82500a185293,f81715524370eb5cca9e7b81becd64ddf21a8a9f,c206a5a371e0e0d834d9547c4983f097ad07ce2b,a11833fbda1eb3594270b57fdde52a883c94a444,a5c5ebffd16e20600b172e8169c7db2b9f377ac7

4.2.7. CODEWORDGENERATION

The trapdoor value for each distinct word and hash algorithm are served as an input for the creation of codeword. For this purpose the trapdoor value is sent to the bloom filter where five hash algorithms are applied on this trap value obtained as a result in Table 4.8.

Table 4.9: Codeword Generation

Codeword Generation	
Input	trapdoor, hash algorithm (crc32)
Output	Codeword
Results	Bits positions of trapdoor: Array: ([0] => 51429, [1] => 71134, [2] => 67320, [3] => 30355, [4] => 73029) Codeword:514297113467320303357302 9

When five hash algorithms are applied on this trapdoor five different bit positions are obtained where this trapdoor is saved. By combining these five bit positions of trapdoor codeword is formed. The resultant bit positions are saved in an array as shown in Table 4.9.

4.2.8. INDEX INSERTION INTO DATABASE

The document to be sent to the cloud server is encrypted and its encrypted document name, the codeword produced in Table 4.9the rank of the word and the document id containing that word serves as an input to be sent to the database. As an output the cloud database contains the encrypted indexed values as shown in Table 4.10.

Table 4.10: Index insertion to cloud database

<table>
<tr><th colspan="2">Index insertion to cloud database</th></tr>
<tr><td>Input</td><td>codeword, rank, document id, encrypted document name</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>Rank</th><th>Did</th><th>enc_doc_name</th></tr>
<tr><td>6828268253237661362730541</td><td>11</td><td>d047bfb595eef24ee6fb5fa49ca2791ca2993655</td><td>qAKYKHFJO1FHr6xU+jJAZg==</td></tr>
<tr><td>1124933565118301443132242</td><td>5</td><td>M086hfb595eef24ee6fb5fa49ca2791ca2993655</td><td>E6SIhJO1Gd9frHPblifTeIDtSJRns=</td></tr>
</table>
</td></tr>
</table>

4.2.9. KEYWORD SEARCH

When a user searches a keyword from the database the above mentioned steps are performed first. Against the searched keyword it is checked if it a stop word or not. The case sensitivity for this keyword can be checked and it can be converted into lower case if case insensitivity has to be achieved. The keyword is then combined with all of the eight keys generated in section 4.2.5 and a Trapdoor is generated in the same way as in section 4.2.6. Using the trapdoor and five hash algorithms codeword is then generated.

Once the codeword of the searched keyword is generated it is sent to the cloud server for querying the indexes of the documents.

4.2.9.1. Codeword outsourcing to cloud

The searched keyword'scodeword is sent to the cloud server where it is matched with already built codewordspresent in the indexes of the documents which were generated during Index Generation step. If searched keyword's codeword is matched with the any of the document indexedcodewords it means this particular word has been matched and the result obtained is the document id and encrypted document name as shown in Table 4.11. The documents are arranged according to descending order of their ranks which are calculated by the frequency of occurrence of the searched words in the documents.

Table 4.11: Codeword Matching

<table>
<tr><th colspan="2">Codeword Matching</th></tr>
<tr><td>Input</td><td>Codeword for the searched keyword</td></tr>
<tr><td>Output</td><td>Document id, Encrypted document name</td></tr>
<tr><td>Results</td><td>Query: SELECT did,encrpyted_document_name , sum(rank) as ranking FROM codewords WHERE codeword='42965381556225265930214 76' or codeword='472065097419677335283085 5' GROUP BY did order by ranking desc

Result:

<table>
<tr><th>Did</th><th>enc_doc_name</th></tr>
<tr><td>d047bfb595eef24ee6fb5fa49ca2791ca2993655</td><td>qAKYKHFJO1FHr6xU+jJAZg==</td></tr>
<tr><td>daad78b1b89f9be6a28a97207d61615569bb4293</td><td>6vyeuB+YyXFECYj0u=yTO6Z</td></tr>
</table>
</td></tr>
</table>

The query in Table 4.11depicts that when codewords are matchedthe documents id's and encrypted document names are returned on the basis of ranking of the documents which is obtained based on the frequency of words in each document. The documents have been ordered

on the basis of ranking and ranking has been calculated by taking the sum of the ranking value for each word in each document.The documents id's containing that code word are returned to the user in descending order of their rankings i.e. Top ranked documents comes first.

5. ANALYSIS AND RESULTS

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

Anything newly built needs testing to see if it works. If it works then it needs to be tested to find out if it will do what you built it for. Software testing is every action intended at evaluating a characteristic of a program or system and to determine that it satisfies its required results. Testing is further to just debugging. Testing can be aimed for quality assurance, authentication and confirmation, or consistency evaluation. Testing can be used as a general metric as well. Two areas of testing are based on the correctness testing and reliability testing. Purpose of Software Testing is the execution of a program or system with the objective of finding errors.

5.2. TESTING OF ALGORITHMS

Testing of algorithms has been done extensively for the proposed system and to get the accurate result all syntax and logical errors have been removed. The algorithms which have been implemented and for which experimental evaluation has been performed are as follows:

- To find out the indexing time of the proposed technique
- To find out the searching time of the proposed technique
- To find out the data outsourcing time for linear encryption technique
- To find out the search time for linear encryption technique

Comparison of search time for a keyword for both proposed technique and linear encryption technique has been experimentally evaluated in this section. The graphs and experimental statistical evaluation proved the usefulness of the proposed technique.

5.3. EXPERIMENTAL EVALUATION

5.3.1. INDEXING TIME OF PROPOSED TECHNIQUE

The indexing time for the document to be outsourced on the cloud server has been calculated by creating the indexes from the original document. There were 150 documents which have been

indexed to be outsourced on the cloud. Out of 150 a sample data of 15 documents has been picked and experimental evaluation has been done. The time requiredto index number of distinct wordsin each document has been shown in Table 5.1.

Table 5.1: Indexing Time for Proposed Technique

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 obtained in Table 5.1 have been graphically shown in Figure 5.1.

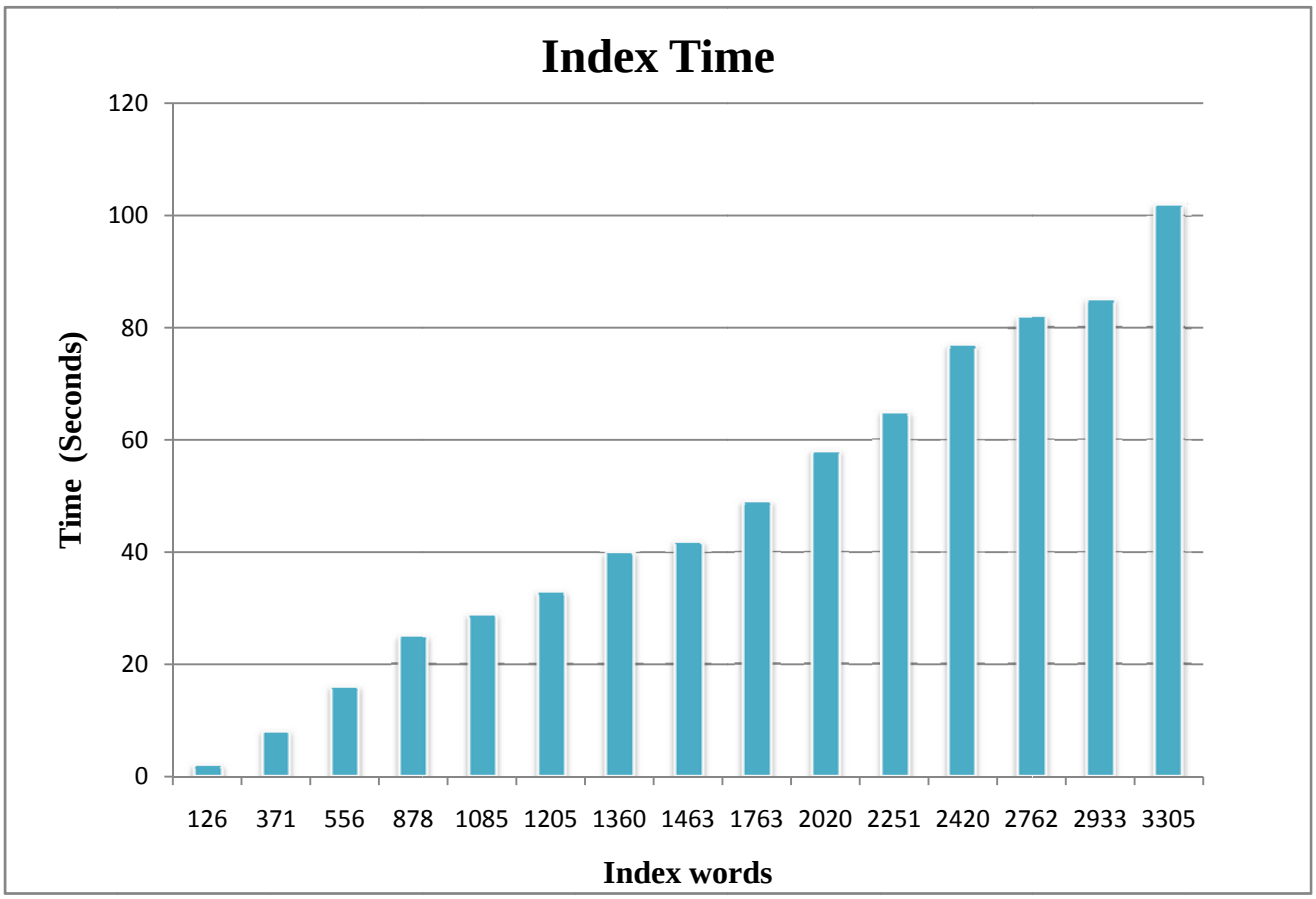

Figure 5.1: Graphical Representation of Index Time for proposed technique

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:

Time required forindexing a document αNo. of Wordsto Index in the document

5.3.1.1. Index Time for One Keyword

Following the results obtained in Table 5.1 the index time for one word to be stored on the cloud server may be calculated as:

Total words = 24498

Total time consumed = 713

One word index time = Total time / Total words = 713/24498 = .0291 seconds

5.3.1.2. Words indexed in One Second

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

Total words= 24498

Total time consumed = 713

Words indexed in one sec = Total words / Total time consumed = 24498/713 = 34.34 words

After the evaluation of the time requirement for indexing words it can be concluded that the time requirement can be estimated for any document by counting words which need to be indexed.

5.3.2. KEYWORD SEARCH TIME FOR PROPOSED TECHNIQUE

When a keyword is searchedit is matchedin the indexes of the documentspresent on the cloud server. A data sample of 20 keywords has been taken. These keywords have been picked from different documents and have been searched. On searching these keywords from the cloud server the time required to search a particular keyword has been calculated.The time required to search a specific keyword has been shown in Table 5.2.

Table 5.2: Search Time for proposed technique

Searched Keywords	Search Time (Sec)
Architecture	3.417
black-box migration	2.427
migration strategy	3.93
Characteristics of Cloud Computing	3.519
enterprise resource planning	3.731
Emulator	1.538
Granularity	2.104
Anti-Malware	1.529
Non Deterministic Finite State Automaton	2.378
challenging research work in cloud computing	2.322
microscopy slides	1.672

image segmentation	1.869
HasPolygon property	2.245
data-intensive	3.417
service-level agreements	2.427
Cultural difference	1.714
Demographic: respondent's personal characteristics	3.142
Mann–Whitney	1.542
BUILDING BLOCKS	1.63
fuzzy keyword	2.07

The statistical results obtained in Table 5.2 have been graphically shown in Figure 5.2.

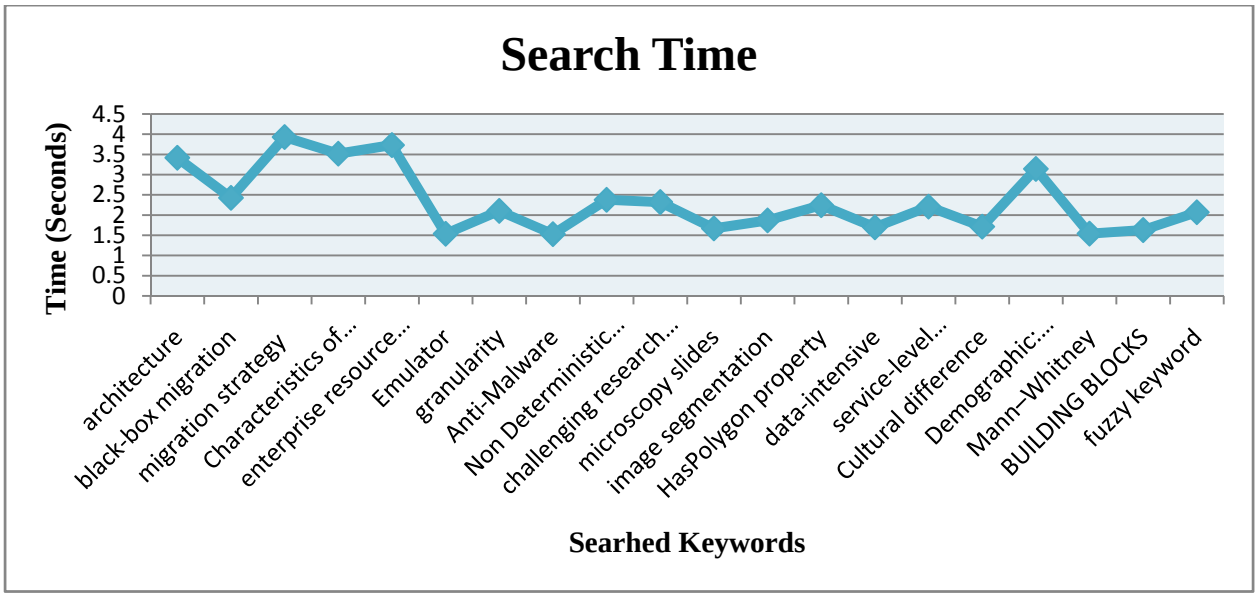

Figure 5.2: Graphical Representation for search time for proposed technique

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.

5.3.3. DATA OUTSOURCING TIME FOR LINEAR ENCRYPTION

Linear encryption does not require any index time for data outsourcing on the cloud but it only requires encryption time for the document to be placed on the server. Table 5.3 shows the documents which are saved on the server in encrypted form along with the time required by each document to be placed on the server. The size in kilobytes has also been shown for each original document. A data sample of 15 documents has been picked up for the experimental evaluation.

Table 5.3: Data Outsourcing Time for Linear Encryption

Document name	Size in KBs	Outsourcing Time (Sec)
How to search eu gin goh.txt	1.90	1
Cloud_computing_security_risk.txt	4.81	1
Optimizing security of cloud computing within the dod.txt	14	2
Deploying public key infrastructure as a cloud service.txt	19.4	2
Service-oriented modeling and architecture.txt	24	2
A wrapping approach and tool for migrating legacy components.txt	34.7	6
Risk management in global software development process planning.txt	37.3	4
Project management a case study.txt	38.5	6
Data protection-aware design for cloud services.txt	31.4	5
The implementation and deployment of an erp system an industrial case study.txt	49	5
Database-agnostic transaction support for cloud infrastructures.txt	74.5	10
Secure multidimensional range queries over outsourced data.txt	72	9
What do software practitioners really think about project success a cross-cultural comparison.txt	117	15
A method engineering based legacy to soa migration method.txt	79.1	11
Authorized private keyword search over encrypted data cloud computingicdcs11.txt	92.5	14

The statistical results obtained in Table 5.3 have been graphically shown in Figure 5.3. This figure shows the document names along the horizontal axis and shows the time required to encrypt a document along with the size of that document. The measure of time has been taken along y-axis.

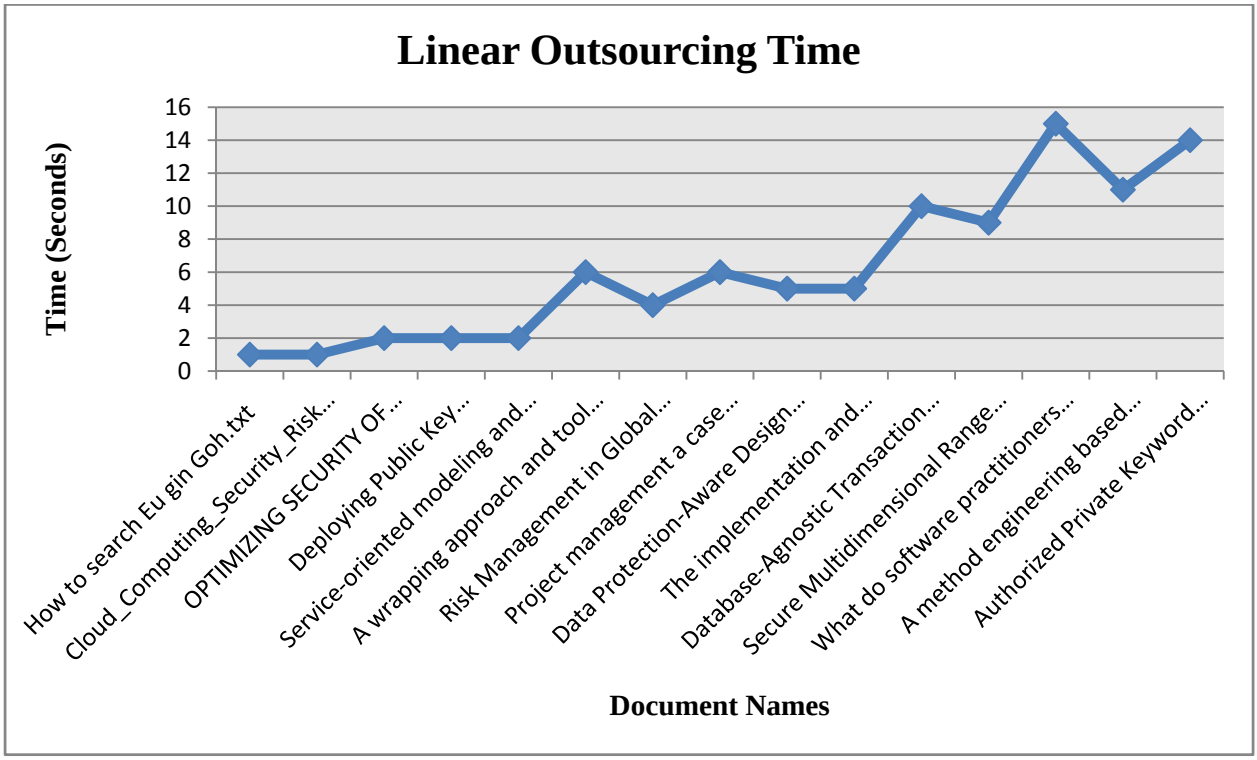

Figure 5.3: Graphical Representation of Outsourcing Time for Linear Encryption

5.3.4. LINEAR SEARCH TIME

The time required for searching a keyword by linear search in encrypted documents has been calculated and shown in Table 5.4. The data set which has been taken for this experimental evaluation is the same data sample which was used during searching the keyword through proposed technique. The data sample contains 20 words and for each word the searching time has been noticed through linear search on encrypted data.

Table 5.4: Linear Search Time

Searched Keywords	Searching Time (Sec)
Architecture	668
black-box migration	772
migration strategy	620
Characteristics of Cloud Computing	693
enterprise resource planning	668
Emulator	613
Granularity	647
Anti-Malware	688
Non Deterministic Finite State Automaton	653
challenging research work in cloud computing	652
microscopy slides	613
image segmentation	629
HasPolygon property	630
data-intensive	628
service-level agreements	689
Cultural difference	667
Demographic: respondent's personal characteristics	676
Mann–Whitney	760
BUILDING BLOCKS	683
fuzzy keyword	624

The statistical results obtained in Table 5.4 have been graphically shown in Figure 5.4.Figure 5.4depicts that on horizontal axis searched keywords have been shown and on vertical axis the time required to search for a specific keyword has been shown. It has been observed that the search time remains almost constant around 600-670 seconds i.e. 10 minutes for 150 documents.

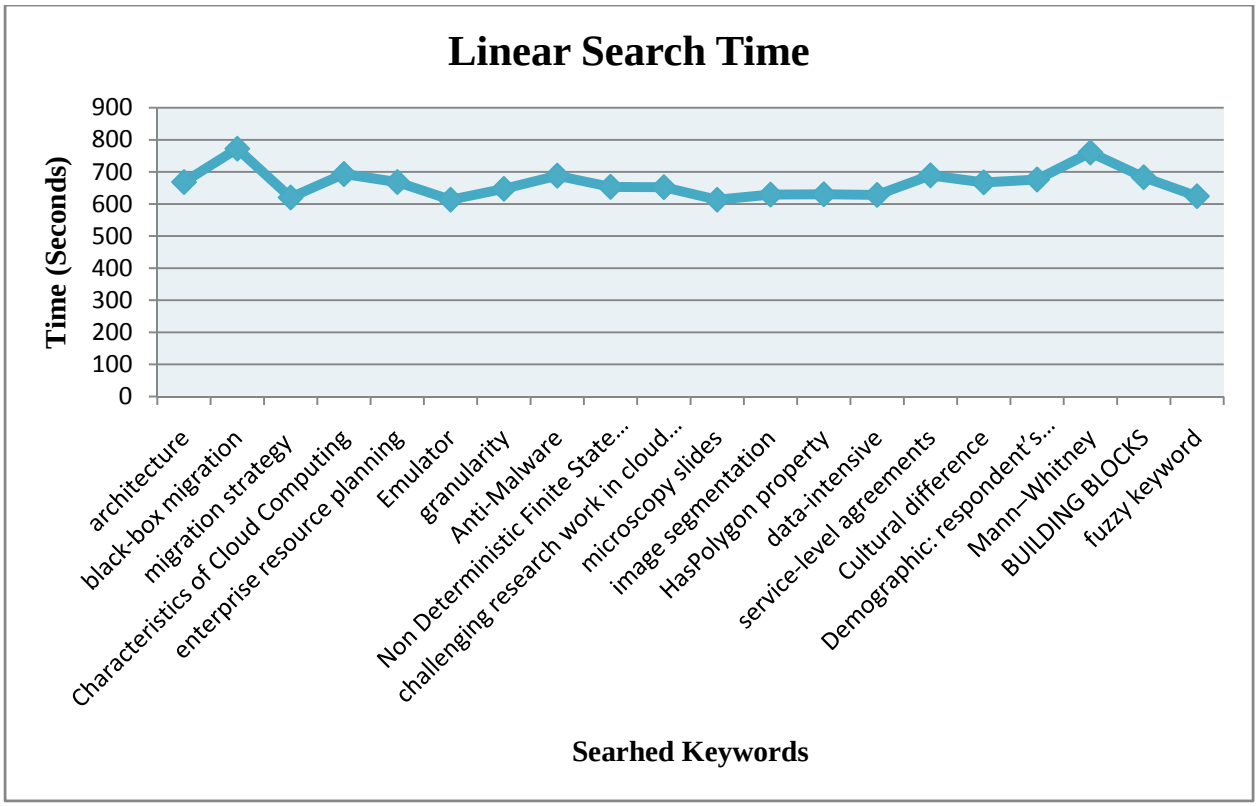

Figure 5.4: Graphical Representation for Linear Search Time

5.3.5. SPACE COMPARISON FOR PROPOSED TECHINQUE AND LINEAR ENCRYPTION

Linear and proposed systems have equal space utilization for encryption where as an additional cost for index space is required in case of proposed technique. Index space overhead for 236982 words is 48.58MB calculated by sample data set.

5.3.6. TIME COMPARISON OF PROPOSED TECHNIQUESEARCH AND LINEAR SEARCH

When a keyword is searched it has been calculated for both proposed technique and linear search. In linear search the documents are first downloaded, decrypted and then searched. In proposed technique keywords are searched from indexes of the documents.The time required to search keywords from both the techniques has been recorded and results have been shown in Table 5.5.

Table 5.5: SearchTime Comparison for Linear and Proposed Technique

Searched Keywords	Time of Proposed Technique (Sec)	Time of Linear Search (Sec)
Architecture	3.417	668
black-box migration	2.427	772
migration strategy	3.93	620
Characteristics of Cloud Computing	3.519	693
enterprise resource planning	3.731	668
Emulator	1.538	613
Granularity	2.104	647
Anti-Malware	1.529	688
Non Deterministic Finite State Automaton	2.378	653
challenging research work in cloud computing	2.322	652
microscopy slides	1.672	613
image segmentation	1.869	629
HasPolygon property	2.245	630
data-intensive	3.417	628
service-level agreements	2.427	689
Cultural difference	1.714	667
Demographic: respondent's personal characteristics	3.142	676
Mann–Whitney	1.542	760
BUILDING BLOCKS	1.63	683
fuzzy keyword	2.07	624

In Table 5.5search time for 20 keywords for both of the proposed technique and linear search has been calculated.

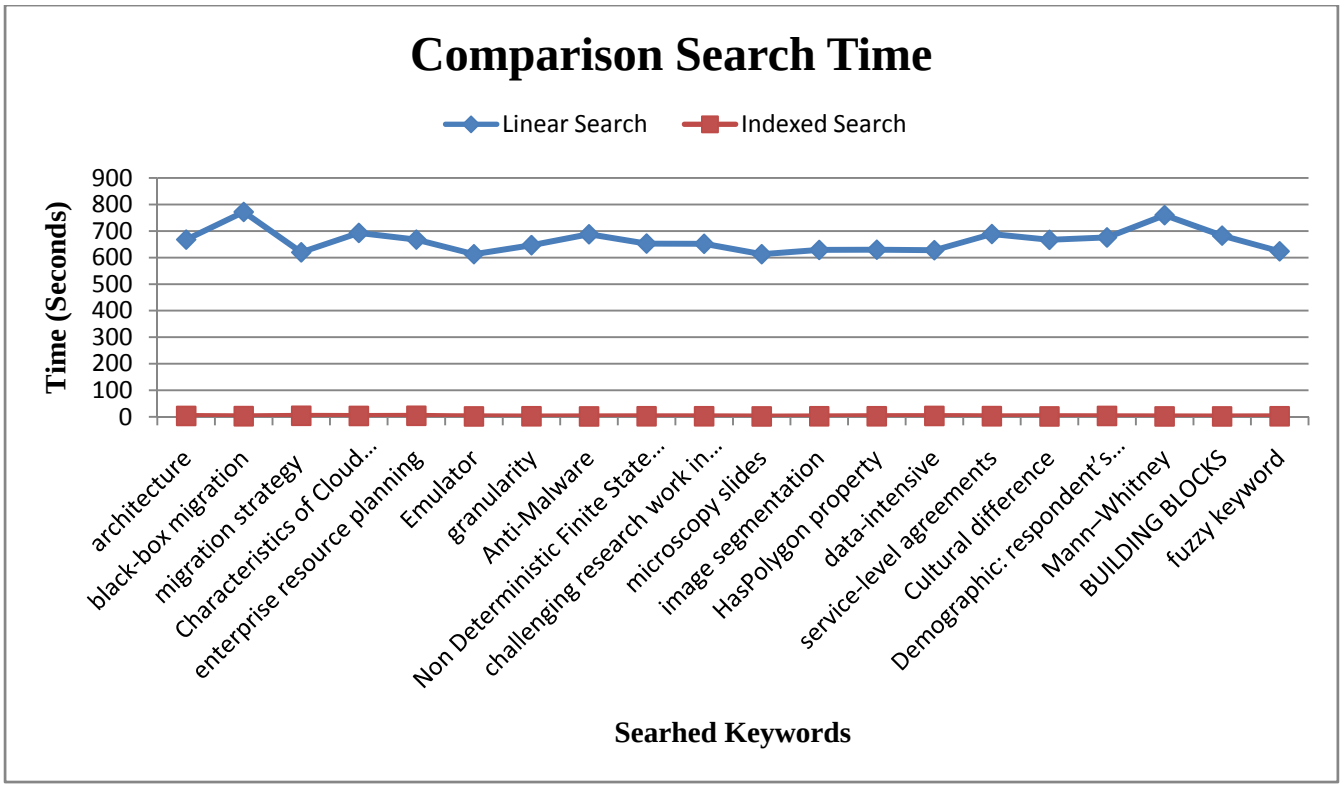

Figure 5.5: Graphical Representation of Search Time Comparison for Linear and Proposed Technique

In Figure 5.5the search time calculated in Table 5.5 has been graphically represented. On x-axis keywords which have been searched are shown whereas on y-axis time acquired by searching process has been shown. The graphical representation shows that the time required to search the keywords using proposed technique is comparatively less than the linear search. Thereforeproposed technique is more efficient as compared to the linear search.

5.4. SECURITY ANALYSIS OF PROPOSED TECHNIQUE

5.4.1. INDEX SECURITY

In this implementation the index generated by the index generation step is secure as the index generation step follows different security levels. Initially a primary key is generated by using SHA512 algorithm which is split into eight equal sized keys. Depending on the security threats the keys can be individually generated by using eight different passwords which makes it more secure. The trapdoors generated by using these eight keys and the hash algorithm used in this technique SHA1 are completely secure. This is because it is very difficult to find the eight

unique keys which are used to generate trapdoor and the hashing algorithm. In this implementation publically available hashing algorithms like SHA1, SHA512 have been used but more secure and efficient hashing algorithms can be used by the data owners.

Trapdoor generation is first step of security enhancement. The codeword generation is second step of security enhancement. A trapdoor is converted to secure codeword by using bloom filters. Each trapdoor is stored in the bloom filter on five bit positions. The bit positions are obtained by applying crc32 checksum algorithm five times on the trapdoor with concatenation of five uniform random numbers with the trapdoor. The use of five uniform random numbers is to get five different bit positions for a trapdoor. The bit positions for the same trapdoor will always be same as needed to generate consistent codeword of a particular trapdoor as codewords need to be matched during the search process. Combining the five bit positions together generates a codeword which is secure and unpredictable as the codewords are numerical numbers. It is very difficult to predict what these numerical numbers represent. This codeword generation step makes another security level which cannot be broken or predicted.

Finally the original document needs to be encrypted in order to be saved on the cloud server. In this implementation a standard AES 256 bit encryption is used. The proposed technique of indexing and search is independent of the encryption technique used. The encryption technique discussed in [36] can also be used. After generation of index of the document with the proposed scheme any best encryption techniquecan be used which can fulfill all the security requirements of the cloud server. In proposed technique symmetric encryption has been used. The data owner according to requirements of security can use symmetric or asymmetric encryption techniques available publically or can create personalized encryption technique for cloud servers.

5.4.2. SEARCH SECURITY

The data user only needs to search document which are most relevant to the searched keywords. The data user does not need to have any knowledge of the implementation details or how the search keywords are converted to codewords and data is retrieved. The data user will send the keywords to the data owner who knows how to convert the keywords to codewords to be searched in the documents index. Other implementation schemes may allow user to send keywords to the cloud server where after authorization keywords will be converted to codewords and matched results will be returned to the data users.

The conversion of the searched keywords to codewords follows the same security levels and security enhancement steps as discussed in Index Security. This proves that Index generation and Search both have same level of security. It also proves that the search of the proposed technique is secure and unpredictable.

The searched keywords when converted to codewords do not reveal any information that what has been searched on the cloud server or which documents have been searched. The documents are placed on the server with new identities which do not reveal the name of the document or its purpose or any other related information. The documents data and their identities are already made secure and unpredictable by the use of secure encryption techniques.

6. CONCLUSION AND FUTURE WORK

In order to summarize the whole research this chapter includes the concluding remarks. The future directions have also be been suggested.

6.1. CONCLUSION

In IT industry Cloud Security is an evolving and emerging matter of interest. For fulfilling enterprise needs a lot of IT industries and organizations need cloud security to be a cost effective option. Organizations store their sensitive data on the cloud considering it to be a trustworthy location. Therefore an environment is needed to be built which provides extremely negligible data leakage over the cloud and hence providing secure data retrieval if required. This study has addressed the issue of information security and secure information retrieval over the cloud.

There are the exiting approaches which claim to propose a secure architecture for efficient data search over the cloud. The problem is that no existing technique has provided the implementation details to achieve secure and efficient search. The aim of this research is to provide secure search in cloud environment using indexing approach for unstructured data. The objective of this research is to have a secure searchable encryption with outflow of secured stored data and efficient search. The technique proposed in this research is based on indexing for encrypted search. The proposed scheme preprocesses the document data to make secure index which is easily searchable in cloud environment without revealing the identity of the document or the index.The purpose of this technique is to make the index and document both secure to be stored on the cloud server as the security threats in the cloud environment are the most critical.

For attaining the purpose of the research the workflow pattern which was adopted is as follows: In Chapter 1 complete introduction of this research including the description of major concepts has been given. The motivation to this research and major contributions to be achieved through this study has been mentioned. The next step was to collect related data through literature survey. Analysis of existing techniques from the literature has been done in Chapter 2. On the basis of existing techniques a new technique for secure data search has been proposed. The proposed idea was implemented and proved to show fast data retrieval whenever search is performed on the cloud server. For this purpose indexing of unstructured data at data owner level has been done. The complete processing and basic scheme of proposed technique has been briefly described in

Chapter 3. The implementation stages have been elaborated in Chapter 4. The analysis of the results has been demonstrated in Chapter 5 and a comparative analysis is performed between the proposed technique and linear search technique.For verifying the implementation of the proposed technique the evaluation method which has been used for checking it is a comparative method. The comparative analysis of the proposed technique has been done with the linear search technique over encrypteddata. Statistical evaluation of both of the techniques has been graphically presented in Chapter 5.

The major contributions which were aimed to achieve have been fulfilled. The unique distinct words from each document which the data owner wants to outsource to the cloud server have been indexed. The ranked keyword search has been implemented. The results obtained at the end are in the form of ranked set of documents. Bloom filters have been used in order to check the exact match of the searched keyword. Exact match search with punctuation marks has been provided. Case insensitivity can be accomplished attaining accurate data retrieval with minor changes in implementation. Stop words elimination has been acquired so that only meaningful words become the part of the keyword search minimizing the search time.

This research has reduced the overhead of decryption before searches. The proposed technique only takes a preprocessing time in start but search is efficient and accurate. The encrypted preprocessed index has been usedwhich does not reveal the identity of the documents or the words present in the documents and hence achieving the secure storage.

Theproposed technique achieves the requirement of high security, efficiency and accurate search. Itcan be used as a secure data retrieval technique by data owners and organizations to minimize security threat and to have efficient search across the cloud server. The limitation of this research work is that whenever data owner wants to change the documents the updating of indexes is needed. Hence updating the index is an overhead.

6.2. CONTRIBUTIONS

The achievementsof this research are as follows:

1. Unique distinct words from each document are indexed.
2. The ranking based keyword search technique is proposed to provide efficient and accurate search.

3. Security is achievedusingbloom filters during indexing unique words of the document.
4. Exact word match with punctuation marks is provided.
5. Case insensitivity can also be achieved with minor changes in the implementation.
6. The experimental results proved efficient and accurate results of the proposed solution.
7. The comparison of proposed search with linear search proved that proposed search is more efficient and accurate.

The proposed technique has reduced the overhead of decryption before searches and the search time on a considerable scale. The preprocessed index has been encrypted due to which identity of the documents or the document contents can be revealed. The decryption is only needed when the relevant documents are returned to the user.

6.2.1. SECURITY ACHIEVEMENTS AT A QUICK GLANCE

The security analysis highlights distinguished features of the proposed technique as shown in Table 5.6.

Table 6.1: Security Analysis for Proposed Technique

Salient Features	Achievement
Any hash algorithms can be used according to needs	Customizability
Eight keys can be generated with eight different passwords	Security
Two distinct levels of security achievement	Security (Trapdoor →Codeword)
Proposed scheme is independent of encryption technique used	Customizability
Search process hold same security level	Security (Trapdoor → Codeword)
Data user or cloud server will not have any knowledge how search is performed	Data privacy and Confidentiality
Cloud server is unaware of searched keywords and which documents have been searched.	Data privacy and Confidentiality
Cloud server is unaware of the documents, their identities and any related information.	Data privacy and Confidentiality

6.2.2. AREAS OF APPLICATIONS

The proposed techniqueis for cloud environments where huge amount of unstructured data is transferred to the cloud servers. This technique secures unstructured data form hacking, unauthorized access and other security vulnerabilities. The proposed technique can be used in different areas where the data security is critical and the amount of data is huge. This technique can be used for securing sensitive medical data of patients and for military data. The usage of encrypted indexing technique is not restricted to the above mentioned applications.

6.3. FUTURE WORK

Current research is based on exact match search for multiple keywords. The proposed research may be extended for following research directions in future.

6.3.1. SENTENCE BASED SEARCH

Multiple keyword based search can be extended to enable user to find complete sentencesbased on positions of the keywords in a sentence.The same basic security technique can be used for making index secure. Position based search can be implemented using auto correlation or cross correlation concepts.

6.3.2. SUB MATCH SEARCH

This scheme provides exact match search but it is unable to handle sub matches therefore in future the sub match property may be added. If it is added it enables a user to search for a part of a sentence. For example on searching the sub word ‘complete’ the word ‘completely’ should also be retrieved.

6.3.3. DATABASE SECURITY

The database queries and table headings should also be made secure so that the information contained in the table rows and columns is not revealed by hackers. This security feature may also be added in future to secure database. 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] K. Jeffery, and B. Neidecker-Lutz, "The Future of Cloud Computing Opportunities for European Cloud Computing Beyond 2010, European Commission Information Society and Media

[2] 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

[3] 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

[4] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing", Technical Report UCB-EECS-2009-28, Univ. of California, Berkeley, Feb. 2009

[5] 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

[6] Ken E. Stavinoha, 2010, What is Cloud Computing and Why Do We Need It?http://isacahouston.org/documents/WhatisCloudComputingandWhyDoWeNeedIt.pdf

[7] 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)

[8] 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

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

[10] 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

[11] 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)

[12] 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)

[13] Gonzalez, N.; Miers, C.; Redigolo, F.; Carvalho, T.; Simplicio, M.; de Sousa, G.T.; Pourzandi, M., "A Quantitative Analysis of Current Security Concerns and Solutions for Cloud Computing", Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on , vol., no., pp.231,238, Nov. 29 2011-Dec. 1 2011

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

[15] 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)

[16] 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

[17] 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

[18] 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

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

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

[21] W. Harrower, “Searching encrypted data,” Department of Computing, Imperial College London, Tech. Rep, 2009

[22] 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

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

[24] 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

[25] 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

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

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

[28] 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

[29] 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

[30] R. Koletka, A. Hutchison, “An Architecture for Secure Searchable Cloud Storage”, Information Security South Africa (ISSA), 2011, 15-17 Aug. 2011, pp. 1-7

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

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

[33] 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

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

[35] 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

[36] Burr, W.E., “Selecting the Advanced Encryption Standard”, Security & Privacy, IEEE , vol.1, no.2, pp.43,52, Mar-Apr 2003

[37] Yan Zhu; Di Ma; Shanbiao Wang, “Secure Data Retrieval of Outsourced Data with Complex Query Support”, Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on Pp.481,490, 18-21 June 2012

[38] Sun S.Chung; Ozsoyoglu, G., “Anti-Tamper Databases: Processing Aggregate Queries over Encrypted Databases”, Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on, vol., no., pp.98,98, 2006