بسم الله الرحمن الرحيم

SUDAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

COLLEGE OF GRADUATE STUDIES

MASTER OF INFORMATION TECHNOLOGY PROGRAM

Search on Encrypted Data: Outsource Personal Database

البحث في البيانات المشفرة في قاعدة بيانات طرف ثالث

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Information Technology

By:

Mawada Adam Mohammed Moaied

Supervision:

Dr.Faisal Mohammed Abdalla

May 2018

الآيـــــــــــة

بسم الله الرحمن الرحيم

قال تعالى:"قَالُواْ سُبْحَانَكَ لاَ عِلْمَ لَنَا
إِلاَّ مَا عَلَّمْتَنَا إِنَّكَ أَنتَ الْعَلِيمُ الْحَكِيمُ
"

سورة البقرة (32)

Dedication

“I don’t know what your destiny will be, but one thing ;the only ones among you who will be really happy are those who have sought and have found how to serve .”

Faisl Mohammed Abdallah

For those who answer the call in middle of the day or night. For those who answer the call from near and from far. For those who answer the call for help with no expectation of personal gain.

This work is dedicated to my sister Siham Ahmed God's mercy on her and ALL the volunteers who serve.

Stay calm .

Acknowledgment

I would first like to thank my supervisor, Dr.Faisal Mohammed Abdallah for all his support throughout the course of the project, as well as Changyu Dong for his input and help during meetings.

I would also like to thank my parents, and my uncle who have always been prepared to go out of their way to give their utmost support during my time at university.

Finally, I would like to thank my friends, in particular Aml Mohammed Ahmed, for friendship and help during my time knowing her.

Abstract

Data security is one of the fundamental Security requirements for outsourced databases in a cloud computing environment. Existing solutions usually suffer from problems such as information leakage, key management, and user authentication .In this thesis to overcome these security challenges, A proposed scheme is introduced to allows users to offload search queries to the cloud. The cloud is then responsible for returning the encrypted files (result) that match the search queries; server simply cannot make a plaintext keyword search on encrypted data. To maintain user's data confidentiality, the keyword search functionality should be able to perform over encrypted cloud data and additionally it should not leak any information about the searched keyword or the retrieved document. The main idea behind Searchable encryption data is to be able to perform an encrypted query without having to download the whole encrypted data. To solve this problem, RC4 algorithm is used for encrypt/decrypt keyword index . and also using the n-gram for approximate keywords search.

The proposed scheme enables a user to store data securely in the cloud by encrypting it before outsourcing and also provides user capability to search over the encrypted data without downloading it and revealing any information about the data or the query.

المستخلص

تأمين البيانات هو أحد متطلبات الأمان الأساسية لقواعد البيانات الخارجية في بيئة الحوسبة السحابية .وتواجه الحلول الحالية عدة مشاكل مثل تسرب المعلومات ، وإدارة المفاتيح ، ومصادقة المستخدم. في هذه الأطروحة للتغلب على هذه التحديات الأمنية ، يسمح النظام المقترح للمستخدمين بالبحث في البيانات المشفرة الموجودة في الخادم. ليكون الخادم مسؤول عن استرجاع الملفات المشفرة التي تتطابق مع استعلام البحث .للحفاظ على سرية بيانات المستخدم ، يجب أن تكون وظيفة البحث عن الكلمات الرئيسية قادرة على الاستعلام في أكثر من مستند مشفر ، بالإضافة إلى عدم تسريب أي معلومات حول الكلمة الرئيسية البحثية (keyword)أو المستند المسترجع. تتمثل الفكرة الرئيسية من بيانات التشفير القابلة للبحث في إمكانية إجراء استعلام مشفر دون الحاجة إلى تنزيل البيانات المشفرة بالكامل .لحل هذه المشكلة ، تم إختيار خوارزمية (RC4) لغرض التشفير / فك التشفير.فهارس الكلمات الرئيسية وكذلك استخدام (n-gram) لتقريب عملية البحث عن الكلمات الرئيسية. يتيح النظام للمستخدم تخزين البيانات بشكل آمن في السحابة(cloud) عن طريق تشفيرها في جانب العميل (client side)قبل وصولها للخادم كما يوفر قدرة المستخدم على البحث عن البيانات المشفرة دون تنزيلها و الكشف عن أي معلومات حول البيانات أو الاستعلام.

Table of Contents

Contents

Table of Contents VII

List of Figures X

List of Tables XI

List of Abbreviations XII

CHAPTER ONE 13

INTRODUCTION 13

Introduction 14

1.1 Background 14

1.2 Problem statement 15

1.3 Objectives 15

1.5 Significant of research 15

1.6 Methodology 16

Figure (1.1) Search over encrypted data 16

1.7 Thesis organization 16

CHAPTER TWO 18

Literature review&related work 19

2.1 Overview 19

2.2 State of the art 19

2.2 .1 Search in Outsourced Personal Database 20

2.2 .2 Email Routing Service 20

2.2 .3 Matching in Internet-based PHR Systems 21

2.3 Outsourced Database Model (ODB) 22

2.3.1 System model:- 22

2.3.2 Objectives: 23
2.4 key requirements of security aspects in outsourcing 24
2.4.1 Confidentiality 24
2.4.2 Integrity 25
2.4.3 Availability 25
2.4.4 Authenticity 25
2.5 Advantages and Disadvantages of Outsourcing 25
2.5.1 The advantages of outsourcing 26
2.5.2 The disadvantages of outsourcing 26
2.6 Methods of searching on Encrypted Data 27
2.6.1 Symmetric Searchable Encryption (SSE) 27
2.6.2. Asymmetric searchable encryption 28
2.6.3. PIR : Private Information Retrieval 29
2.6 Related works 29
CHAPTER THREE 31
METHODOLOGY 31
Methodology 32
3.1 Introduction .... **Error! Bookmark not defined.**
3.2 System design 32
3.3 Frame work 32
3.3.1 flowchart 32
3.3.2 RC4 algorithm 33
3.3.3 Gram – Based Technique 34
3.3.4 Jaccard Technique 34
3.4 Implementation details 35
3.4.1 Encryption: 35
3.4.2 Search: 36

3.4.3 Decryption : 36

3.5 Architecture: 37

3.5.1 admin (data owner): 37

3.5.2 user : 38

3.5.3 Server 39

3.6 Security requirements 40

CHAPTER FOUR 41

IMPLEMENTATION 42

4.1 Introduction 42

4.2 System implementation 42

4.2.1 Client side procedures 42

4.2.2 Server side procedures 45

4.3 Result 46

4.4 Security analysis 48

4.4.1 Query Privacy 48

4.4.2 Data Files 48

4.4.3 Searchable Index 49

CHAPTER FIVE 50

CONCLUSION & RECOMMENDATION 50

Conclusion and Recommendation 51

5.1 Conclusion 51

5.2 Recommendation 51

REFERENCE 52

Reference 53

List of Figures

Figure (1.1) Search over encrypted data 16
Figure (2.1) An Illustration of Outsourced PHRs 21
Figure (2.2) Outsourced Database Model 23
Figure (3.1) flowchart for the document and index encryption process 33
Figure (3.2) General architecture and components of a search over encrypted data. 37
Figure (3.3) Functionality of data owner and stored data into the server 38
Figure (3.4) search in encrypted data scenario........ 39
Figure (4.1): System screen 43
Figure (4.2) uploaded file procedure........ 43
Figure (4.3) Searching results 44
Figure (4.4) download screen........ 44
Figure (4.5) data owner login........ 45
Figure (4.6) file uploaded table........ 45
Figure (4.8) Jaccard calculation 47
Figure (4.9) time search function........ 48

List of Tables

Table 2.1: Related work 30

List of Abbreviations

Abbreviation	Stand for
ODB	Outsource database model
PHR	Personal healthcare record
PIR	Private information retrieval
PPC	Partition plaintext ciphertext
SED	Search in encrypted data
SSE	Symmetric searchable encryption

CHAPTER ONE

INTRODUCTION

CHAPTER ONE

INTRODUCTION

1.1 Background

With the rapid development of cloud computing and mobile networking technologies, users tend to access their stored data from the remote cloud storage with mobile devices. The main advantage of cloud storage is its ubiquitous user accessibility and also its virtually unlimited data storage capabilities. Despite such benefits provided by the cloud, the major challenge that remains is the concern over the confidentiality and privacy of data while adopting the cloud storage services . For instance, unencrypted user data stored at the remote cloud server can be vulnerable to external attacks initiated by unauthorized outsiders and internal attacks initiated by the untrustworthy cloud service providers . There are several reports that confirm data breaches related to cloud servers, due to malicious attack, theft or internal errors . This raises concern for many users/organizations as the outsourced data might contain very sensitive personal organization/information.[1]

Several researches have addressed the issue of ensuring confidentiality and privacy of cloud data without compromising the user functionality. Confidentiality refers to the secrecy of the stored data so that only the client can read the contents of the stored data. To solve the problem of confidentiality, data encryption schemes can come in handy to provide the users with some control over the secrecy of their stored data. This has been adopted by many recent researches which allow users to encrypt their data before outsourcing to the cloud . However, standard encryption schemes will dampen users' searching ability over the stored data, since after encryption a user simply cannot use a plaintext keyword to perform a search anymore and therefore cannot retrieve the contents in an efficient way.[1]

1.2 Problem statement

How to search and retrieve for specific encrypted document on outsourced server without reveal any information ?

The problem arises when Alice want to retrieve document she must download the whole documents from Bob server and running decryption process which consume time , she can search for document in server using special keyword that may leak information to Bob also may cause overhead in query.

1.3 Objectives

i. To define formal security model and generic construction of searchable encryption scheme .

ii. To construct an efficient secure searchable encryption scheme in the model.

iii. To support the proposed scheme with security analysis to achieve privacy.

iv. To implement efficiency in respect to time and space in proposed scheme.

1.5 Significant of research

The primary goal is to lower the computation at the user end since he/she is paying for the services of the server. The server would take the majority of the work in terms of computation and performing the search. Another goal is to reduce the encrypted file size on the server. However, should be aware of the fact that as the encrypted file size decreases, the number of output bits from the cryptographic encryption scheme reduce, and therefore lowering the randomness and the security of the scheme. Moreover, an elaborate and full search capability might require word-by-word encryption, resulting in increased memory usage. Therefore, in order to develop an effective solution, one should take into consideration the tradeoffs between memory overhead, search capability and security when searching on encrypted data.

1.6 Methodology

This research aims to design and develop a privacy preserving data storage and retrieval system. The scopes involve the use of searchable encryption algorithms to search for specific keywords within an encrypted content, i.e., without requiring the user to download the database and decrypt its contents before searching can be performed. The proposed solution, in this research, introduce a solution for the problem of searching on encrypted data that optimally uses the n-gram based technique. The basic idea behind solution is to create a secure keyword list, encrypt keywords in the documents as a stream. Identify keywords in the documents and encrypt them separately, and make use of the keyword-based schemes' properties to perform a faster search with comparatively small overhead. Figure 1.1 shows the methodology of search over encrypted data

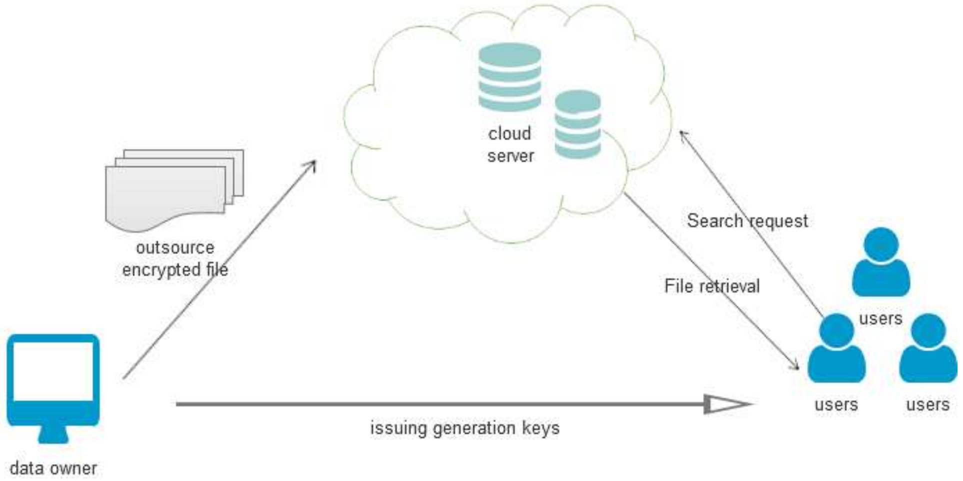


Figure (1.1) Search over encrypted data

1.7 Thesis organization

Chapter one: Introduction: In this section, the main points discussed are about the Overview, the Background of the project, the scopes and limitations of the project and the approach to research employed are discussed.

Chapter two: Literature Review: Definitions and overview about the different information security methods to gather knowledge on the existing theories of search in encrypted data review it for proposing an improvised system for providing the required security and discuss about different functionalities of algorithms used.

Chapter three: In this chapter will also discuss the technical information about the system, including the system and software design decisions taken. As well as the structure of the system, showing the various modules and database organization.

Chapter four: In this chapter, the results will be shown and the success of the project will be evaluated. And discuss the security analysis for proposed scheme.

Chapter five : finally in this chapter contain conclusion and recommendation.

CHAPTER TWO

LITERATURE REVIEW & RELATED WORK

CHAPTER TWO

LITERATURE REVIEW&RELATED WORK

2.1 Overview

Continued growth of the Internet and advances in networking technology have fueled a trend toward outsourcing data management and information technology needs to external Application Service Providers. By outsourcing, organizations can concentrate on their core tasks and operate other business applications via the Internet, rather than incurring substantial hardware, software and personnel costs involved in maintaining applications in house. Database outsourcing is a recent and important manifestation of this trend. In this model, the provider is responsible for offering adequate software, hardware and network resources to host the clients' databases as well as mechanisms for the client to efficiently create, update and access the outsourced data.

The database outsourcing paradigm poses numerous research challenges which influence the overall performance, usability and scalability. One of the foremost challenges is the security of stored data. A client stores its data (which is a critical asset) at an external, potentially untrusted, Database Service Provider site. It is essential to provide adequate security measures to protect the stored data from both malicious outsider attacks and the Database Service Provider itself. Security in most part, implies maintaining data integrity and guarding data privacy. Although some work has been done to protect data confidentiality while guaranteeing its continued availability to authorized users, the problem of providing efficient integrity in this model has not received much attention. [4]

2.2 State of the art

In This section, presents a brief summary of the searchable encryption schemes. Searchable encryption scheme can be designed based on:

2.2 .1 Search in Outsourced Personal Database

Suppose Alice is a frequent traveler and needs to access her database during her travel anytime and anywhere in the world. For this, Alice can outsource her personal database to a third-party service provider, such as Google or Dropbox. With this approach, Alice needs to reveal everything (the data and search criteria) to the third-party service provider, which makes the solution undesirable from the privacy perspective. To achieve a privacy-preserving solution, Alice can employ a SED scheme to encrypt her database and outsource the ciphertext. Later on, Alice can issue a search query (containing encrypted search criteria) to the service provider, which can then search in the database and return the encrypted documents which match the search criteria. As to this scenario, we distill the following security requirements.

– Only Alice can generate contents for the sourced database and decrypt the encrypted contents in the database, and only Alice can issue meaningful search queries.

– No entity, including the server, should learn what Alice has searched for.[3]

2.2 .2 Email Routing Service

Among all our sourcing services, email may be one of the most well-known examples, where users' email data is stored and related services are managed by the email service providers. In email services, the service providers normally have access to all emails of their customers in plaintext so that a lot of privacy concerns exist (e.g. sensitive email messages and targeted advertisements). Now, suppose that there is an email service provider, which wants to provide secure email service and allow users to receive encrypted emails. In this situation, a user Alice can employ a SED scheme and have all her emails encrypted under her (public) key. Later on, Alice can ask the service provider to search in her encrypted emails and then selectively retrieve the interesting ones. For instance, during her vacation, Alice can simply retrieve the emails labeled as "urgent" through her smart phone, without being bothered by other emails. As to this scenario, we distill the following security requirements.

– Every entity should be able to generate encrypted emails for Alice, but only Alice can read her emails.

– Only Alice can issue meaningful email retrieval queries. In addition, Alice may also want the service provider to scan her encrypted emails, in particular the attachments, to detect viruses or malwares without learning unnecessary plaintext information

– No entity, including the server, should learn what Alice has searched for.

2.2 .3 Matching in Internet-based PHR Systems

An Internet-based PHR (personal healthcare record) system, such as Microsoft Health Vault 1, helps users store their PHRs and allow the information to be accessed and edited via a web browser or some APIs, and they may also help users find kindred spirits (i.e. build social networks) and share their information. Considering a user, say Alice, her PHR data can come from a lot of sources. For example, she can get prescription results from her doctor, treatments from a hospital, test results from a laboratory, and monitoring results from home-based sensors. Quite often, a lot of Alice’s PHR data may be directly sent to Alice’s account, while the rest will be input by Alice herself. Fig.2.1 shows a general picture of an Internet-based PHR system. In most existing Internet-based PHR systems, users will be provided privacy controls. However, there are a number of concerns which stop users from sharing their data. One concern is that the system providers say Microsoft, are always able to fully access the data. Although there will be some privacy agreement, But users may still worry about that these providers may abuse their data. The other concern is that, even if the service providers behave honestly, their databases may be compromised, in which case all data may be leaked. Since PHRs are sensitive

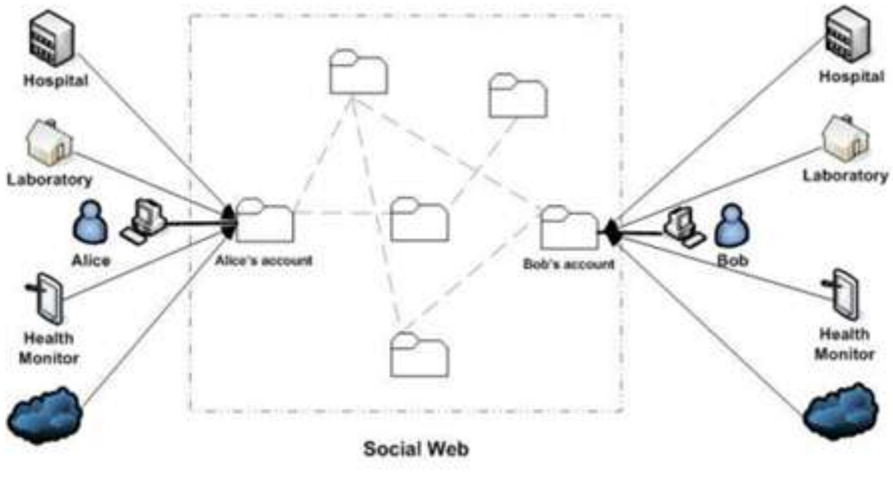


Figure (2.1) An Illustration of Outsourced PHRs [2] information to individuals, and an information leakage may cause undesirable consequences, such as being discriminated by the potential employer because of a disease. To solve the privacy problem, users can employ a SED scheme and have all their PHR data encrypted under their own (public) keys. Moreover, the users can authorize third-party server(s) to match their encrypted data without recovering the plaintext information. As to this scenario, we distill the following security requirements.

– A user, say Alice, should be able to allow multiple entities to generate contents for her, and only Alice should be able to decrypt the encrypted contents intended for her.

– Together with another user, Alice can authorize third-party server(s) to match their ciphertext. The authorization should support different levels of granularity, and third-party server(s) should not be able to learn any information more than the match result, such as "equal" or "not-equal" in exact matching.

This application scenario has motivated the SED schemes for joint databases in the asymmetric-setting [3]

2.3 Outsourced Database Model (ODB)

In the Outsourced Database Model (ODB), organizations outsource their data management needs to an external service provider. The service provider hosts client's databases and offers seamless mechanisms to create, store, update and access (query) their databases. This model introduces several research issues related to data security which we explore.

2.3.1 System model:-

The Outsourced Database Model show in figure (2.1) below consists of 3 entities: (1) the data owner(s), (2) the database service provider (server) and (3) the client(s) (also referred to querier (s)). The data owner creates, modifies and deletes the contents of the database. The server hosts the owner's database, i.e., the owner outsources its database to the server. The clients issue queries about the database to the server.

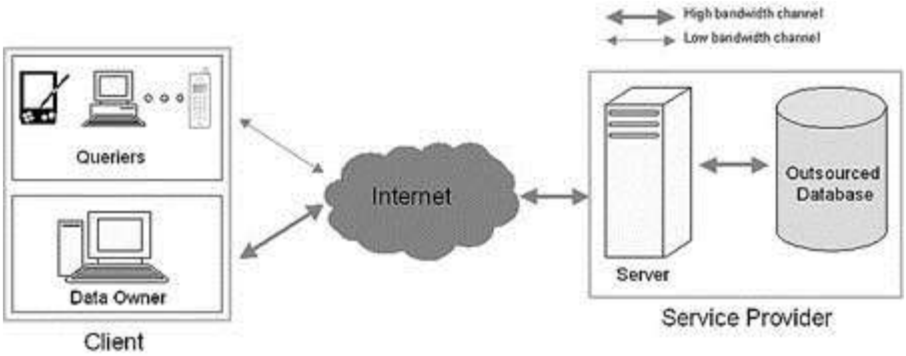


Figure (2.2) Outsourced Database Model [5]

Some of the parameters identifying a specific ODB include the number of owners and clients and the type of trust in the server. Is the server trusted with the data contents but not with integrity? Or do we not trust the database administrators and therefore need to employ encryption to provide data privacy?

2.3.2 Objectives:

To address various security issues that arise in the Outsourced Database Model. These range from providing data confidentiality, authenticity and integrity, to enabling an untrusted server to run queries over encrypted data. We also focus on the performance aspects of our solutions.

I. Authenticity and Integrity

Using signatures we provide mechanisms to allow the querier (client) to ensure that the records returned from the untrusted server originated from the data owner and have not been tampered with. Aim at minimizing the bandwidth and computation required to enable this verification. A new signature scheme, Condensed-RSA, is proposed and we compare its performance with an elliptic curve based signature scheme introduced by Boneh, et al.

II. Data Privacy

If the database server is fully untrusted, the measures need to be taken such as to protect the owner's data privacy. The goal is to hide the data contents from the server, by employing data encryption, while still allowing the server to operate the database. In other words the challenge can be formulated as: how to allow the server to perform queries over encrypted data.

III. Efficient Secure Storage Model in RDBMS

Several database vendors already offer integrated solutions that aim to provide data privacy within existing products. Treating security and privacy issues as an afterthought often results in inefficient implementations. Some notable RDBMS storage models (such as the N-ary Storage Model) suffer from this problem. We analyze issues in storage, looking at trade-offs between security and efficiency, and then propose a secure storage model, Partition Plaintext Ciphertext (PPC), which enables efficient cryptographic operations while maintaining a high level of security. [6]

2.4 key requirements of security aspects in outsourcing

In This section discussed the key requirements of security aspects in database outsourcing.

2.4.1 Confidentiality

Assures that only the authorized and intended users or systems are given consent to access the data. Data confidentiality refers to keep the data concealed from unauthorized access when it is stored and also when the data is in transit state. While ensuring the confidentiality, some additional dimensions are considered viz. user privacy and access privacy. Privacy is one of the primary requirements to achieve the security. User privacy conceals the user identity when he fetches or manipulates the data. Access privacy is assured when the access pattern of database and intended database records for a particular user are kept secret.[7]

2.4.2 Integrity

Assures that the data stored in database or in transmission state are not modified or manipulated except by trusted persons or processes. Completeness and correctness are two important dimensions of integrity Completeness means that query results obtained by fetching all records from the database and no any record containing the predicate in the query is excluded. It ensures that entire results are obtained when a query is fired on the database. Correctness means that the query results obtained from server are tamperproof and generated by original server. To verify that integrity is maintained, query assurance mechanism needs to be incorporated which ensures that query fired against the database is correctly and completely executed by service provider or process including all matched predicates in query.[7]

2.4.3 Availability

Ensures that data are available to the trusted users and systems when they access database in an authorized manner. It is degree or extent to which database is in operable state which is calculated in terms of reliability. It is recommended for service providers to provide always-on availability of database to their valued and authorized users. To provide high level of availability, database system’s down-time should be kept low. [7]

2.4.4 Authenticity

Implies that contracts, query transactions and communication are genuine and identities of the involved entities (users and system) are verified and known. To provide the authenticity, the digital signatures are used. [7]

2.5 Advantages and Disadvantages of Outsourcing

Outsourcing most commonly known as offshoring has pros and cons to it. Most of the time, the advantages of outsourcing overshadow the disadvantages of outsourcing.

2.5.1 The advantages of outsourcing

i. **Swiftness and Expertise:**

Most of the times tasks are outsourced to vendors who specialize in their field. The outsourced vendors also have specific equipment and technical expertise, most of the times better than the ones at the outsourcing organization. Effectively the tasks can be completed faster and with better quality output. [8]

ii. Concentrating on core process rather than the supporting ones:

Outsourcing the supporting processes gives the organization more time to strengthen their core business process

iii. **Risk-sharing:**

One of the most crucial factors determining the outcome of a campaign is risk-analysis. Outsourcing certain components of your business process helps the organization to shift certain responsibilities to the outsourced vendor. Since the outsourced vendor is a specialist, they plan your risk-mitigating factors better [8]

iv. **Reduced Operational and Recruitment costs:**

Outsourcing eludes the need to hire individuals in-house; hence recruitment and operational costs can be minimized to a great extent. This is one of the prime advantages of offshore outsourcing.[8]

2.5.2 The disadvantages of outsourcing

i. **Risk of exposing confidential data:** When an organization outsources HR, Payroll and Recruitment services, it involves a risk if exposing confidential company information to a third-party

ii. **Synchronizing the deliverables:** In case you do not choose a right partner for outsourcing, some of the common problem areas include stretched delivery time frames, sub-standard quality output and inappropriate categorization of responsibilities. At times it is easier to regulate these factors inside an organization rather than with an outsourced partner

iii. **Hidden costs:** Although outsourcing most of the times is cost-effective at times the hidden costs involved in signing a contract while signing a contract across international boundaries may pose a serious threat

iv. **Lack of customer focus:** An outsourced vendor may be catering to the expertise-needs of multiple organizations at a time. In such situations vendors may lack complete focus on your organization’s tasks. [8]

With all these pros and cons of outsourcing to be considered before actually approaching a service provider, it is always advisable to specifically determine the importance of the tasks which are to be outsourced. It is always beneficial for an organization to consider the advantages and disadvantages of offshoring before actually outsourcing it.[8]

2.6 Methods of searching on Encrypted Data

With the rise of cloud computing and storage in the last decade or so, many people have raised concerns about the security issues of outsourced data. And to match this, cryptographers have come up with many proposals aiming to solve all of these problems. The most well-known of these methods, Fully Homomorphic Encryption, can in theory offer a perfect solution to this, but is extremely inefficient and not currently practical for most real-world situations. Other solutions include Oblivious RAM and special forms of identity- based encryption, but the seare both still pretty expensive.[10]

2.6.1 Symmetric Searchable Encryption (SSE)

Which we looked at in this week's study group, stands at the other end of the scale. It is extremely practical, capable of handling extremely large (hundreds of GB) databases, and even has performance comparable to MySQL. However it does compromise somewhat on leakage. It is similar in some ways to deterministic encryption, but ensures that leakage only occurs when a query is made (rather than as soon as the database is uploaded), and has some clever techniques to minimize the quantity of this leakage.[10]

Scenario

A client holds a database (or collection of documents) and wishes to store this, encrypted, on a third party server. The client later wants to search the database for certain keywords and be sent an encrypted list of documents that contain these keywords. Note that we're not searching the entire contents of the database, only a set of keywords that have been indexed in advance (just as Google doesn't look in every HTML file on the web for your search terms, instead using precomputed data structures based on keywords). Also note that the server only returns a set of encrypted identifiers, rather than the documents themselves. In the real world the client would then request the documents from the server. The remarkable aspect about the Crypto paper is that it aims to do the above in time proportional to the number of documents matching the least frequent keyword in the search query -- independent of the database size! Efficiency-wise, this seems optimal -- you can't do much better this, even on plaintext data. The compromise, of course, comes with leakage, which we'll look at shortly.[10]

2.6.2 Asymmetric searchable encryption

Currently, there are several applications that need more than a symmetric searchable encryption. We can imagine a person who wants to retrieve in a public data base for an existing encrypted document she didn’t store herself. This situation would be impossible to solve unless she has a common secret with the person who encrypted these data. This scheme introduced by Boneh & Al[9]. by analogy to cryptographic scheme, allows a number of users who have a public key to store data in the untrusted server but only the person who had the secret key can perform a search to test of the occurrence of a word. This scheme underwent a number of evolutions either in complexity or search features that allow searching for a set of keyword at once.[10]

2.6.3 PIR: Private Information Retrieval

This approach is slightly different from schemes because the data stored in the outsourced servers were unencrypted. The focus is instead solely on allowing the user to perform a search to retrieve documents without revealing the access pattern. Following the news, access to Google alerts, getting updates without revealing the user’s interest and consequently preserving anonymity.[10]

2.6 Related works

Table 2.1 Related works

Date	Author	Title	Description	Published
August, 2005	Mehmet Ucal[11]	Searching on Encrypted Data	keyword based schemes have very small overhead when compared to non-keyword based schemes. Also, their search time is much lower.In this section, we propose several improvements to the scheme discussed in the previous section in order to increase performance in encryption time, search time, and storage space. Our modifications are based on the observations that we have made in our implementation in the previous section along with the benefits offered by the keyword-based schemes when used over large datasets. Our intention is to optimally combine the advantages of both schemes (keyword and non-keyword) to get better performance and storage results.	Department of Electrical and Computer Engineering University of Maryland College Park, MD

Date	Author	Title	Description	Published
2010	Rei Yoshida Et-al [12]	Practical Searching Over Encrypted Data By Private Information Retrieval	Since the privacy-preserving techniques areof significant importance in information retrieval, in this paper we investigate the CRYPTO'07 scheme to show that it is not practical at all even with the latest technology due to the expensive computation cost involved. By our experimental result, Boneh et al.'s protocol seems not to perform better than a trivial solution, which will deny the usability of system proposed by D.Boneh	IEEE
2015	Salam et-al[1]	Implementation of searchable symmetric encryption for privacy-preserving keyword search on cloud storage	This paper aims to study privacy preserving keyword search over encrypted cloud data. Also, we present our implementation of a privacy preserving data storage and retrieval system in cloud computing. For our implementation, we have chosen one of the symmetric key primitives due to its efficiency in mobile environments.	Springer open journal
2016	P. Anitha and D.Vijayalakshmi [13]	Generating secret key for multi-keyword ranked search over encrypted cloud data	The search process of the UDMRS scheme is a recursive method upon the tree, named as "Greedy Depth first Search (GDFS)" algorithm.- Based on the UDMRS scheme, build the basic dynamic multi-keyword ranked search(BDMRS) scheme with using the secure kNN algorithm.-The BDMRS scheme can defend the Index Confidentiality and Query Confidentiality in the identified cipher text model.	International Research Journal of Engineering andTechnology (IRJET)

CHAPTER THREE

METHODOLOGY

CHAPTER THREE

METHODOLOGY

3.1 Introduction

In this chapter, searches in one communication round were trying to be achieved, with high efficiency in respect to time and space. Furthermore the security for the indexes and the trapdoor need to be adequate. The chapter will also discuss the technical information about the system, including the system and software design decisions taken. As well as the structure of the system, showing the various modules and database organization.

3.2 System design

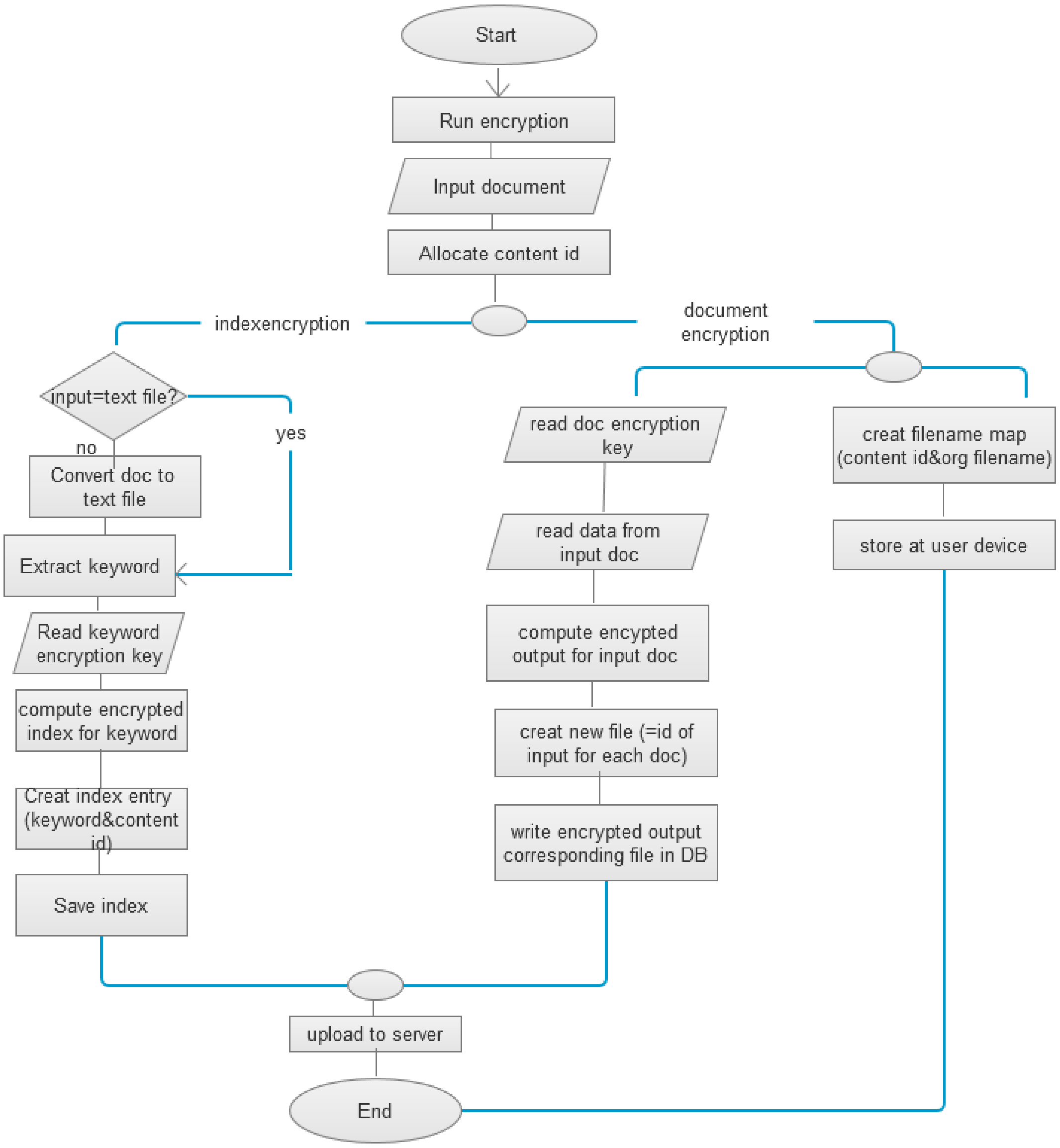
This section provides a detailed background on the search over encrypted data system scheme. In particular, discuss the proposed framework, implementation details, system architecture, and security requirements for searchable encryption scheme.

3.3 Frame work

The main Modules contain four parts: flowchart, RC4 Encryption and Grams-Based Technique and jaccard Technique

3.3.1 Flowchart

Figure 3.1 below show the flowchart for the document and index encryption process. It shows the flowchart of the encryption module for the system. It provides details work-flow of the module to compute the encrypted document and encrypted index



Start
Run encryption
Input document
Allocate content id
indexencryption
document encryption
input=text file?
yes
no
Convert doc to text file
Extract keyword
Read keyword encryption key
compute encrypted index for keyword
Creat index entry (keyword&content id)
Save index
read doc encryption key
read data from input doc
compute encypted output for input doc
creat new file (=id of input for each doc)
write encrypted output corresponding file in DB
creat filename map (content id&org filename)
store at user device
upload to server
End

Figure (3.1) flowchart for the document and index encryption process

3.3.2 RC4 algorithm

RC4 generates a pseudo-random stream of bits (a key-stream). As with any stream cipher, these can be used for encryption by combining it with the plaintext using bit-wise exclusive-or. Decryption is performed the same way (since exclusive-or is a symmetric operation). To generate the key stream, the cipher makes use of a secret internal state which consists of two parts:

1- A permutation of all 256 possible bytes.

2- Two 8-bit index-pointers.

The permutation is initialized with a variable length key, typically between 40 and 256 bits, using the key-scheduling algorithm (KSA). Then the stream of bits is generated by a pseudo-random generation algorithm.[14]

3.3.3 Gram – Based Technique

One of the most efficient technique for constructing fuzzy set is based on grams. The gram is a substring of a string that is used for making approximate search. While gram has been widely used for constructing inverted list for approximate string search, we use gram for the matching purpose. We shall utilize the fact that any primitive edit operation will affect at most one specific character of the keyword, leaving all the remaining characters unused. In other words, the relative order of the remaining characters after the primitive operations is always kept the same as it was before the operation. [15]

To generate the keyword set, use the concept of N-grams index, which is used to perform queries on plain-text files. N-grams is a sequence of k characters.

For example, “cou”, “our”, “urs” and “rse” are all the 3-grams of the word “course”. We use the character $ to denote the beginning or the end of a word. Thus, the set of 3-grams generated is: “$co”, “cou”, “our”, “urs”, “rse” and “se$”. In a N-grams index, our dictionary contains all the N-grams of every word in the collection. For each N-grams, create a posting list of all the words in the collection that contain all the characters in the gram. For instance, in Figure 5the 3-gram “emp” would point to all the words such as employable and employee. During the indexing process, system first constructs the dictionary of all the N-grams in the collection. Posting list for each N-grams are then generated. All of these posting lists compose the N-gram index, that called safe index. This predefined index will be used to generate keyword set [16]

3.3.4 Jaccard Technique

Let’s assume that a user queries the keyword *K*. First, we generate the k-grams for keyword *K*, called $G (K)$. For every gram $g_i \in G_k (K)$, the system looks for g_i in the k-gram index introduced above and returns the list of words containing the gram g_i. To reduce our search space, we only want to retrieve vocabularies that are closely similar to the user’s query. If W is one of the words in our k-gram index that contain the gram gi, use the Jaccard Coefficient (|A∩B| / |A∪B|)to measure the similarity of the word *K* and the word *W*. Sets *A* and *B* represent the set of k-grams for *K* and *W*, respectively. If *W* is equal to *K*, *W* will have the highest Jaccard coefficient value compare to the other words in the index. If the Jaccard coefficient of *W*, λ_W , is bigger than our threshold value $\lambda_{\min}$, *i.e*

$$\lambda_w = |A_W \cap B_K| / |A_W \cup B_K| > \lambda_{\min}$$ [16]

add W to keyword set F^k.. Because each word in the keyword set generated for the word K has its own Jaccard coefficient (λ), our fuzzy keyword set is sorted in descending order based on the words' (λ)values.[16]

3.4 Implementation details

The scheme is implemented and tested under the following environment: php codegniter, Windows 7 64-bit operating system .

The system consists of the following modules: encryption, search and decryption. These modules are integrated with a Key Generator which provides the encryption/ decryption keys.

3.4.1 Encryption:

This module provides the document and index encryption functionality. First, the key generator is initiated to generate key. Since symmetric key generation does not require much computational power, we assume that the key generator is located at the user device. This resolves the key distribution issue. The key generation is a one-time process to generate and store the keys. After the key generation, a user inputs a set of document for the encryption and index creation. When user inputs the document set, the Encryption module uses the Keyword Extractor block to pull out the keyword and builds an index of all the words in the document. After keyword extraction, input documents and the document index are fetched to the Encryption block. The Encryption block uses the keys generated by the key generator to create the encrypted document set and encrypted document index. Concurrently, the Encryption block also creates a filename mapping object/ filename index which contains the mapping of document IDs to the corresponding original filenames and extensions. This is stored in plaintext at the user device. Finally, the encrypted files are uploaded to the server.

3.4.2 Search:

The search module provides the functionality of searching a keyword over the encrypted data stored at the cloud server. When user inputs the searched keyword(s), Search Token Generation computes a search token(s)/ trapdoor(s) using the keyword/index encryption key. This generated search token is then sent to the cloud server to perform the search operation. Upon receiving the search token, the cloud server find out pointer(s) to the appropriate encrypted documents by comparing the search token with the encrypted index values. Following this, the cloud server retrieves the encrypted document(s) containing the searched keyword(s) and sends it back to the client along with the corresponding document ID(s).

3.4.3 Decryption :

The Decryption module that is located at the user's device provides the decryption functionality for a given set of encrypted documents. The Decryption module requires Encrypted Document, corresponding Document ID, Document Decryption Key and Filename Mapping Object as input. First, the Decryption module uses the filename mapping object and the document ID to retrieve the original filename and extension of the encrypted document. Then it decrypts the document using the document decryption key and stores it to the user device.

3.5 Architecture:

The system architecture contains three main components: owner, user and server as shown in figure (3.2).

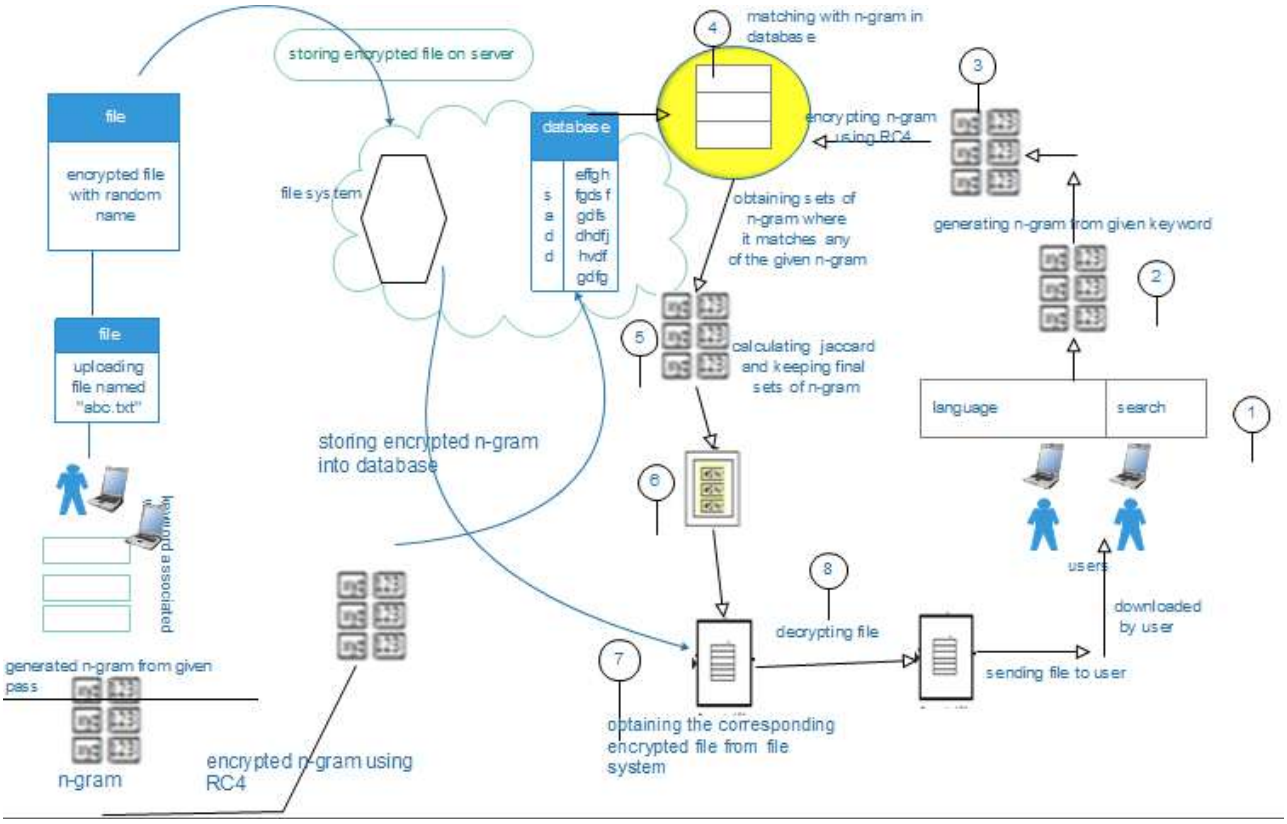

storing encrypted file on server
file
encrypted file with random name
file
uploading file named "abc.txt"
keyword associated
generated n-gram from given pass
n-gram
file system
database
s a d d
effgh fgds f gdfs dhdfj hvdf gdfg
storing encrypted n-gram into database
encrypted n-gram using RC4
4
matching with n-gram in databas e
3
encrypting n-gram using RC4
obtaining s ets of n-gram where it matches any of the given n-gram
generating n-gram from given keyword
2
5
calculating jaccard and keeping final sets of n-gram
language
search
1
6
users
8
decrypting file
downloaded by user
7
sending file to user
obtaining the corresponding encrypted file from file system

Figure (3.2) General architecture and components of a search over encrypted data.

3.5.1 admin (data owner):

After owner login :

Uploading file named “abc.txt”

Encrypting file using base 64 encode

Encrypted file with random name

Storing encrypted file on server file system with secret name

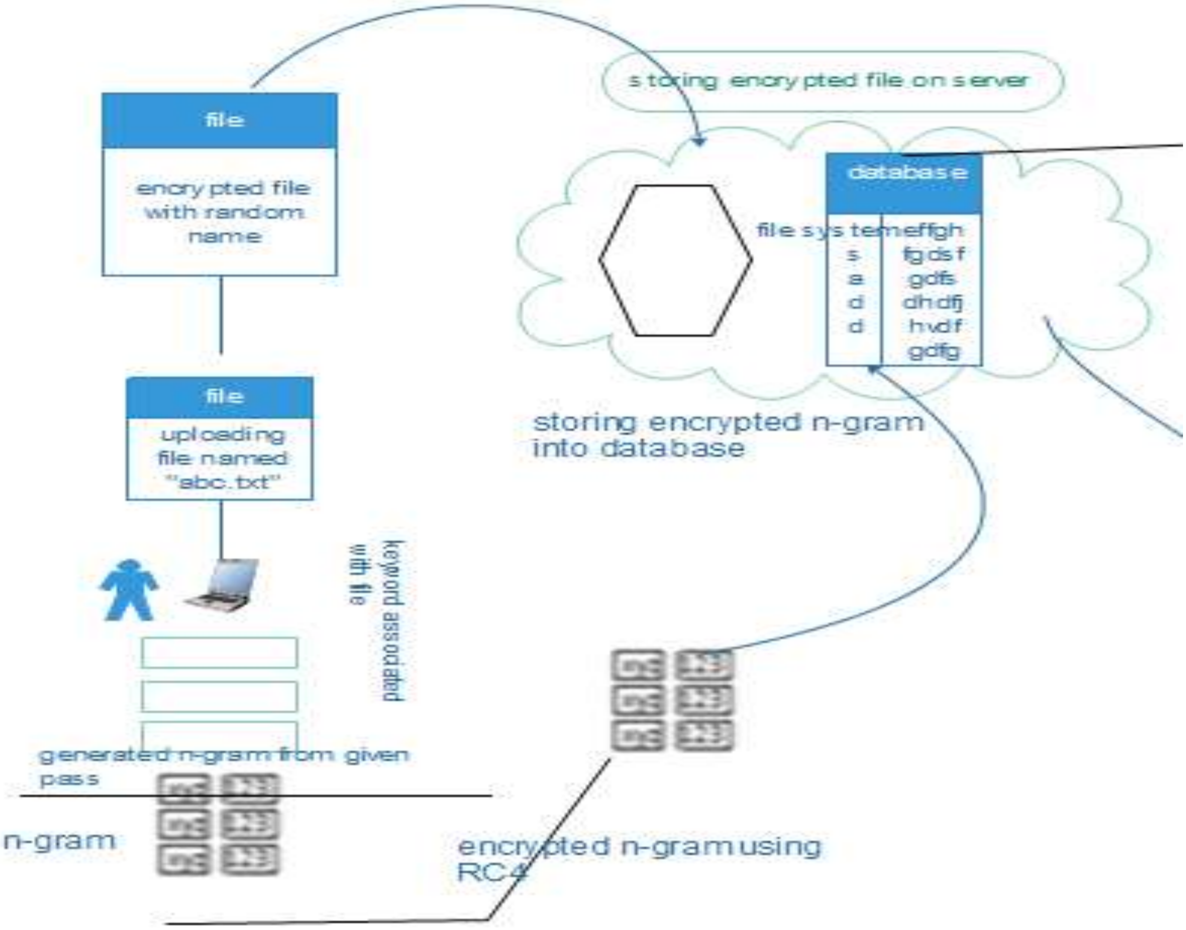
In the same time when the user uploading file; keyword generated and associated with file

Generated n-grams from given keyword

Encrypting n-grams with RC4 encryption using secret key

Storing encrypted n-grams into database

The figure bellow shows the functionality of data owner and how to upload file and the n-gram generated from given key word and stored in the server



file
encrypted file with random name
storing encrypted file on server
database
file system
s a d d
effgh fgdsf gdfs dhdfj hvdf gdfg
file
uploading file named "abc.txt"
storing encrypted n-gram into database
keyword associated with file
generated n-gram from given pass
n-gram
encrypted n-gram using RC4

Figure (3.3) Functionality of data owner and stored data into the server

3.5.2 User:

Step 1: User login:

Step 2: User search on specific file using keyword

Step 3: Generated n-grams from given keyword

Step 4: Encrypting n-grams with RC4 encryption using secret key

Step 5: Matching with n-grams storing in database

Step 6: Obtaining the sets of n-grams where it matches any of the given n-gram for keyword

Step 7: Keeping final set of n-gram for key word

Step 8: Obtaining the corresponding the encrypted file from server file system

Step 9: Decrypting file using base 64 decode

Step 10: Sending file to user

Step 11: File downloaded by user

Figure (3.4) explain how to search in encrypted data and how to generated the n-gram from given keyword search and encrypt the n-gram to compared with n-gram stored in data base that associated with file .

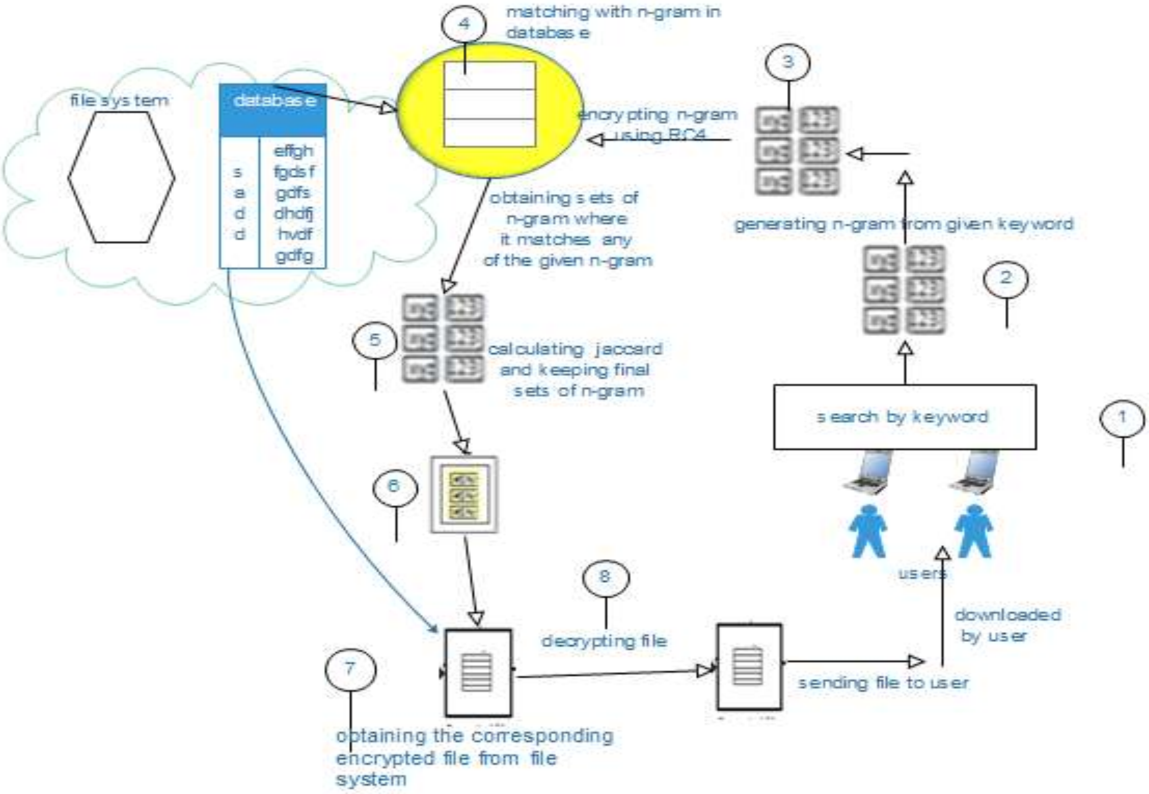

4
matching with n-gram in databas e
3
file sys tem
databas e
s
a
d
d
effgh
fgds f
gdfs
dhdfj
hvdf
gdfg
encry pting n-gram using RC4
obtaining s ets of n-gram where it matches any of the given n-gram
generating n-gram from given key word
2
5
calculating jaccard and keeping final sets of n-gram
s earch by keyword
1
6
users
8
downloaded by us er
decrypting file
7
sending file to us er
obtaining the corresponding encrypted file from file system

Figure (3.4) search in encrypted data scenario

3.5.3 Server

The data storage and retrieval service to user Consists of cloud data server and service manager. Is used to store the outsourced encrypted data receiving the encrypted search queries from the data user Tests the encrypted n-gams from user and encrypted n-grams in the cloud storage.

The encrypted data that satisfies the search criteria is retrieved and sent back to the data owner upon completion of the test.

Should not learn any information from the operation.

3.6 Security requirements

In general, the following requirements should be satisfied when constructing a search over encrypted data.

Retrieved data: Server should not be able to distinguish between documents and determine search contents.

Search query: Server should not learn anything about the keyword being searched for the server can retrieve nothing other than pointers to the encrypted content that contains the keyword.

Query generation: Server should not be able to generate a coded query. The query can be generated by only those users with the relevant secret key.

Search query outcome: Server should not learn anything about the contents of the search outcome.

Access patterns: Server should not learn about the sequences and frequency of documents accessed by the user.

CHAPTER FOUR

IMPLEMENTATION

CHAPTER FOUR

IMPLEMENTATION

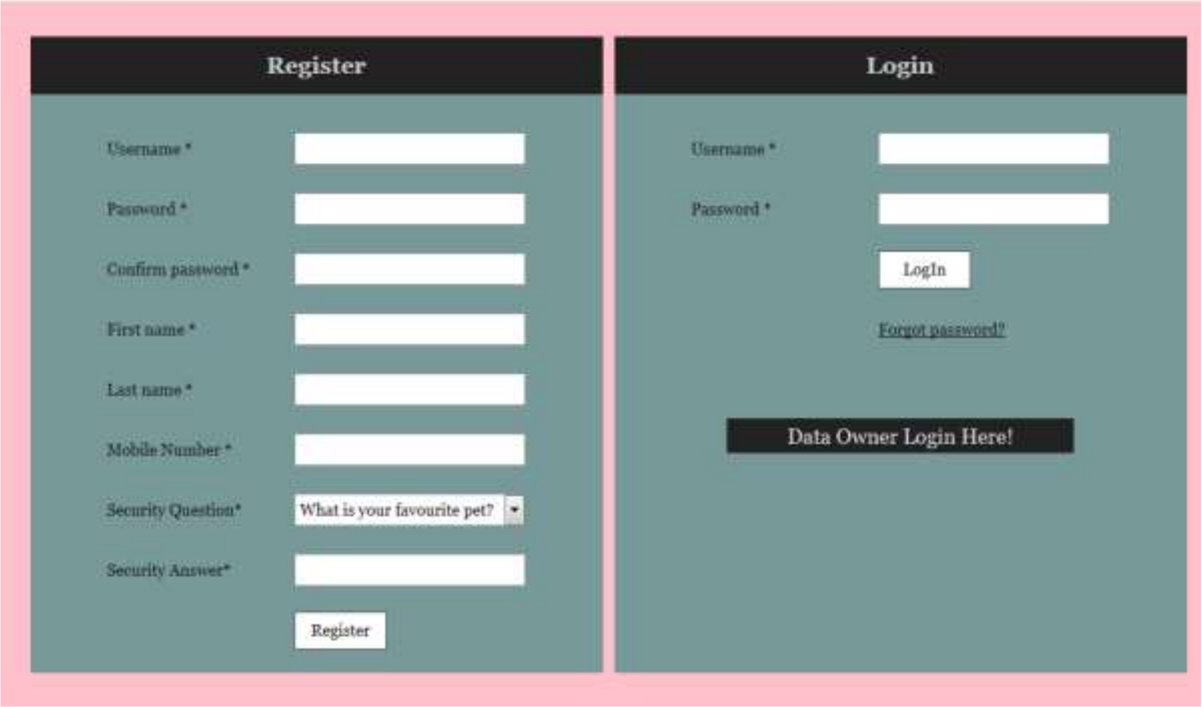
4.1 Introduction

In this chapter, the implementation of the proposed system is introduced, the screen shot shows some steps in the implementation process. Finally the results and security analysis is presented and analyzed for proposed scheme. This can be measured using the performance of the implementation, in terms of time, space overheads and security.

4.2 System implementation

4.2.1 Client side procedures

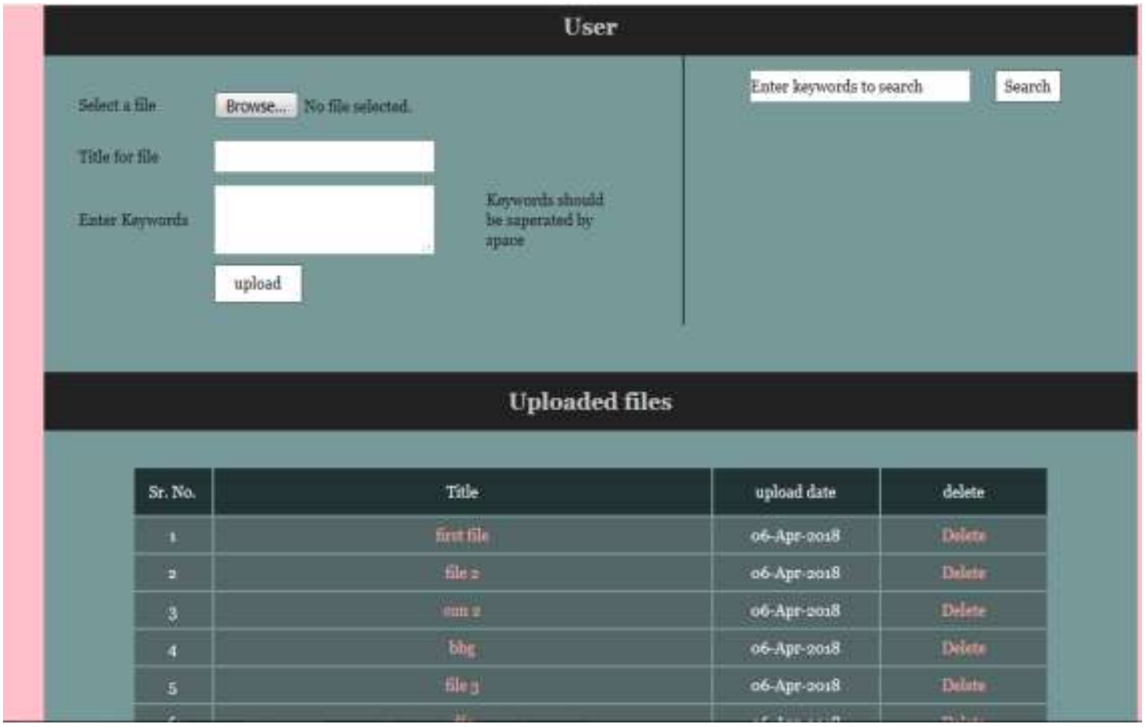
Figure (4.1) explain the procedures in the client side. Figure shows the system login screen. If you are not registered before you are entitled to register and in case you are registered you are asked for your username and password. Also includes the system login screen as the data owner



Security Question* What is your favourite pet?

Figure (4.1): System screen

The figure (4.2) shows the user screen that enables it to upload the file into the database. The system requests a title for file and a password to be generate the n-gram from given password, then the file is encoded with base 64 encode, the n-gram is encrypted with the RC4 algorithm, stored them into the server.



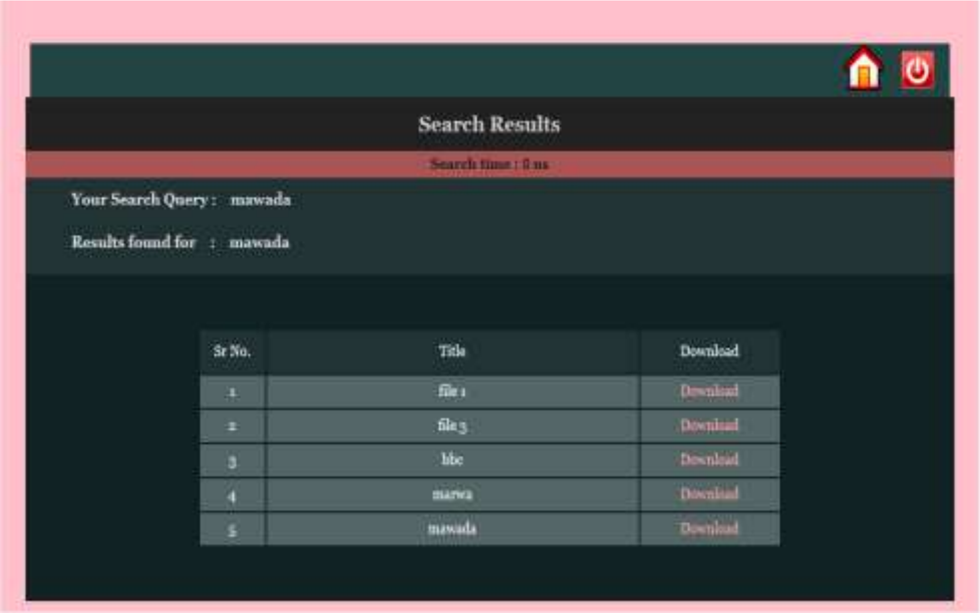
No file selected.

Keywords should be saperated by space

Sr. No.	Title	upload date	delete
1	first file	06-Apr-2018	Delete
2	file 2	06-Apr-2018	Delete
3	ssm 2	06-Apr-2018	Delete
4	bhg	06-Apr-2018	Delete
5	file 3	06-Apr-2018	Delete

Figure (4.2) uploaded file procedure

Figure (4.3) Displays the search result in the encrypted data by using the password for the file. You can also download the file from the server by command (download)



Search Results

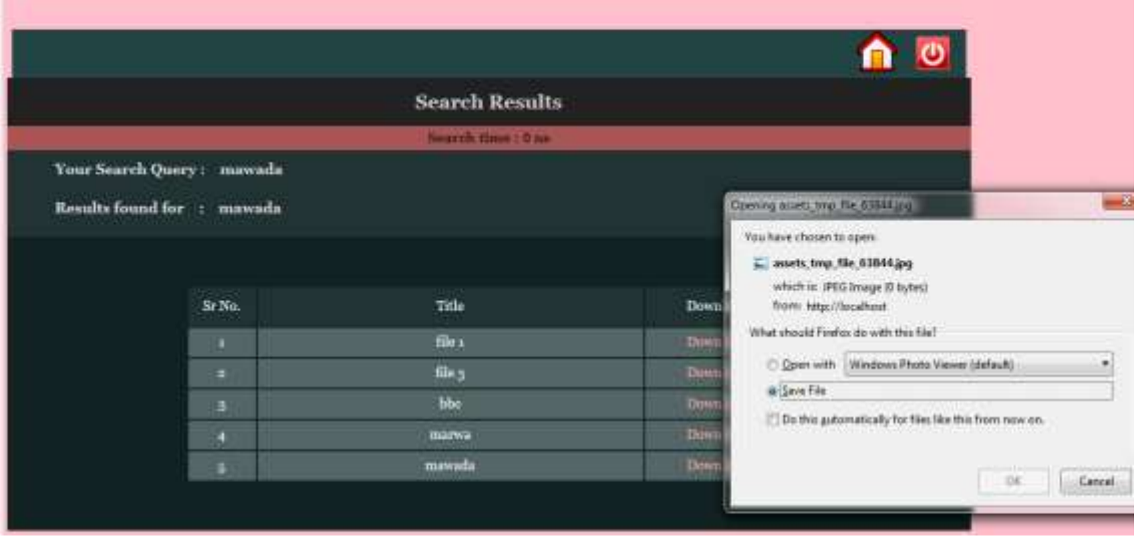
Search time : 0 ms

Your Search Query : mawada

Results found for : mawada

Sr No.	Title	Download
1	file 1	Download
2	file 3	Download
3	bbc	Download
4	marwa	Download
5	mawada	Download

Figure (4.3) Searching results



Search Results

Search time : 0 ms

Your Search Query : mawada

Results found for : mawada

Sr No.	Title	Down
1	file 1	Down
2	file 3	Down
3	bbc	Down
4	marwa	Down
5	mawada	Down

Opening assets_tmp_file_63844.jpg

You have chosen to open:

assets_tmp_file_63844.jpg

which is: JPEG Image (0 bytes)

from: http://localhost

What should Firefox do with this file?

○ Open with Windows Photo Viewer (default)

● Save File

□ Do this automatically for files like this from now on.

Figure (4.4) download screen

Figure (4.5) shows the data owner login screen. System asked owner for username and password

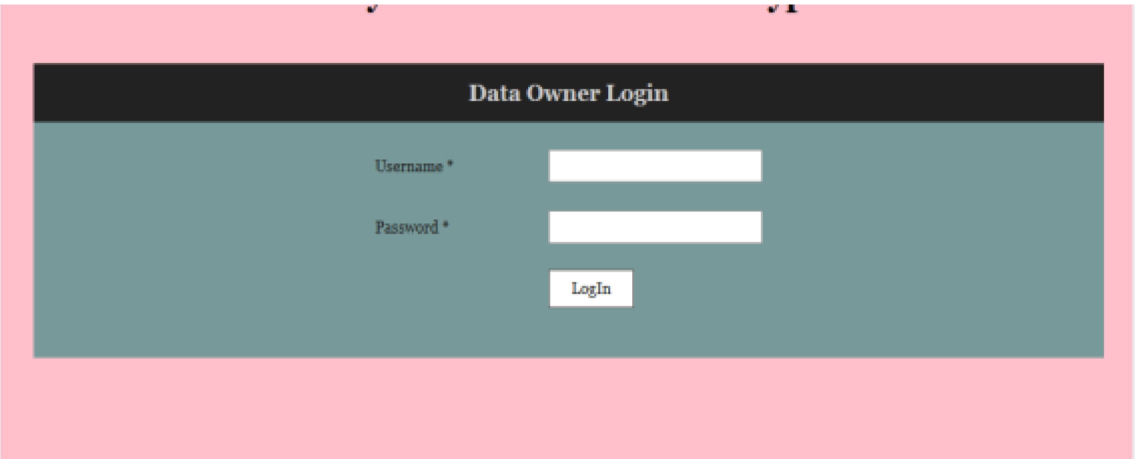
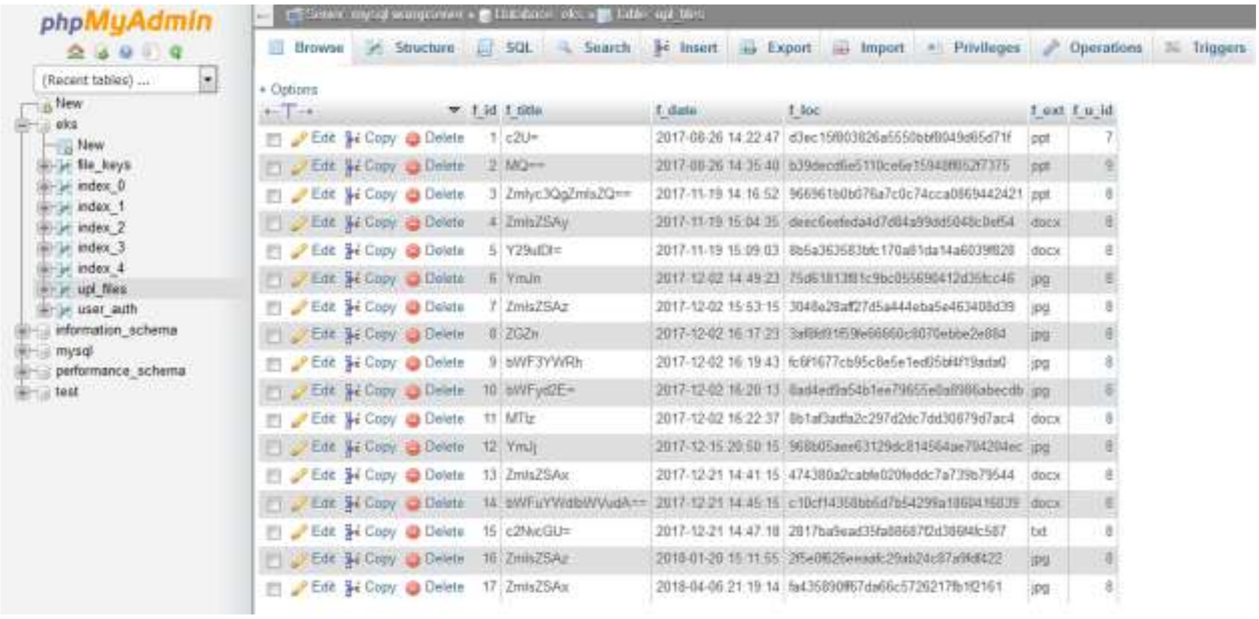


Figure (4.5) data owner login

.

4.2.2 Server side procedures

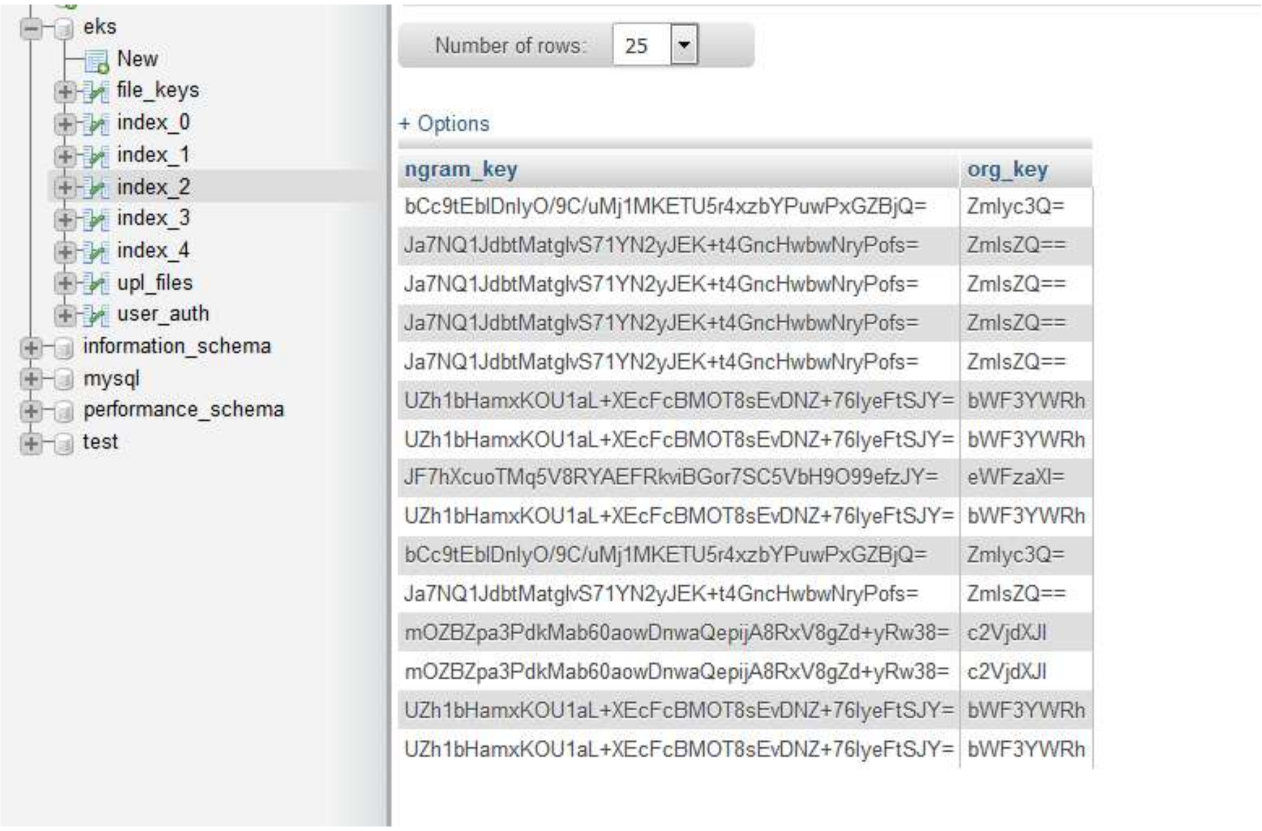
Figure (4.6) shows the table of file upload located in the database contains file location in encrypted form and file ID .



f_id	f_title	f_date	f_loc	f_ext	f_u_id
1	c2U=	2017-08-26 14:22:47	d3ec15f803826a5550bbf8049d65d71f	ppt	7
2	MQ==	2017-08-26 14:35:40	b39decdfe5110ce6e15948f852f7375	ppt	8
3	Zmlyc 3QgZmlsZQ==	2017-11-19 14:16:52	966961b0b076a7c0c74cca0869442421	ppt	8
4	ZmlsZSAy	2017-11-19 15:04:35	deec6eefeda4d7d84a99dd5048c0ef54	docx	8
5	Y29uIDI=	2017-11-19 15:09:03	8b5a363583bfc170a81da14a6039f828	docx	8
6	YmJn	2017-12-02 14:49:23	75d61813f81c9bc055690412d35fcc46	jpg	8
7	ZmlsZSAz	2017-12-02 15:53:15	3048e28aff27d5a444eba5e463408d39	jpg	8
8	ZGZn	2017-12-02 16:17:23	3af8d91f59e66660c8070ebbe2e884	jpg	8
9	bWF3YWRh	2017-12-02 16:19:43	fc6f1677cb95c8e5e1ed05bf4f19ada0	jpg	8
10	bWFyd2E=	2017-12-02 16:20:13	8ad4ed9a54b1ee79655e0a8986abecdb	jpg	8
11	MTIz	2017-12-02 16:22:37	8b1af3adfa2c297d2dc7dd30879d7ac4	docx	8
12	YmJj	2017-12-15 20:50:15	968b05aee63129dc814564ae794204ec	jpg	8
13	ZmlsZSAx	2017-12-21 14:41:15	474380a2cabfe020feddc7a739b79544	docx	8
14	bWFuYWdlbWVudA==	2017-12-21 14:45:15	c10cf14358bb6d7b54299a18694f6839	docx	8
15	c2NvcGU=	2017-12-21 14:47:18	2817ba9ead35fa88687f2d3869f4c587	txt	8
16	ZmlsZSAz	2018-01-20 15:11:55	2f5e0f626eeadc29ab24c87a9fdf422	jpg	8
17	ZmlsZSAx	2018-04-06 21:19:14	fa4358908f67da66c5726217fb1f2161	jpg	8

Figure (4.6) file uploaded table

Figure (4.7) shows the index of n-gram keys located in the database contains n-gram keys and original document ID



Number of rows: 25

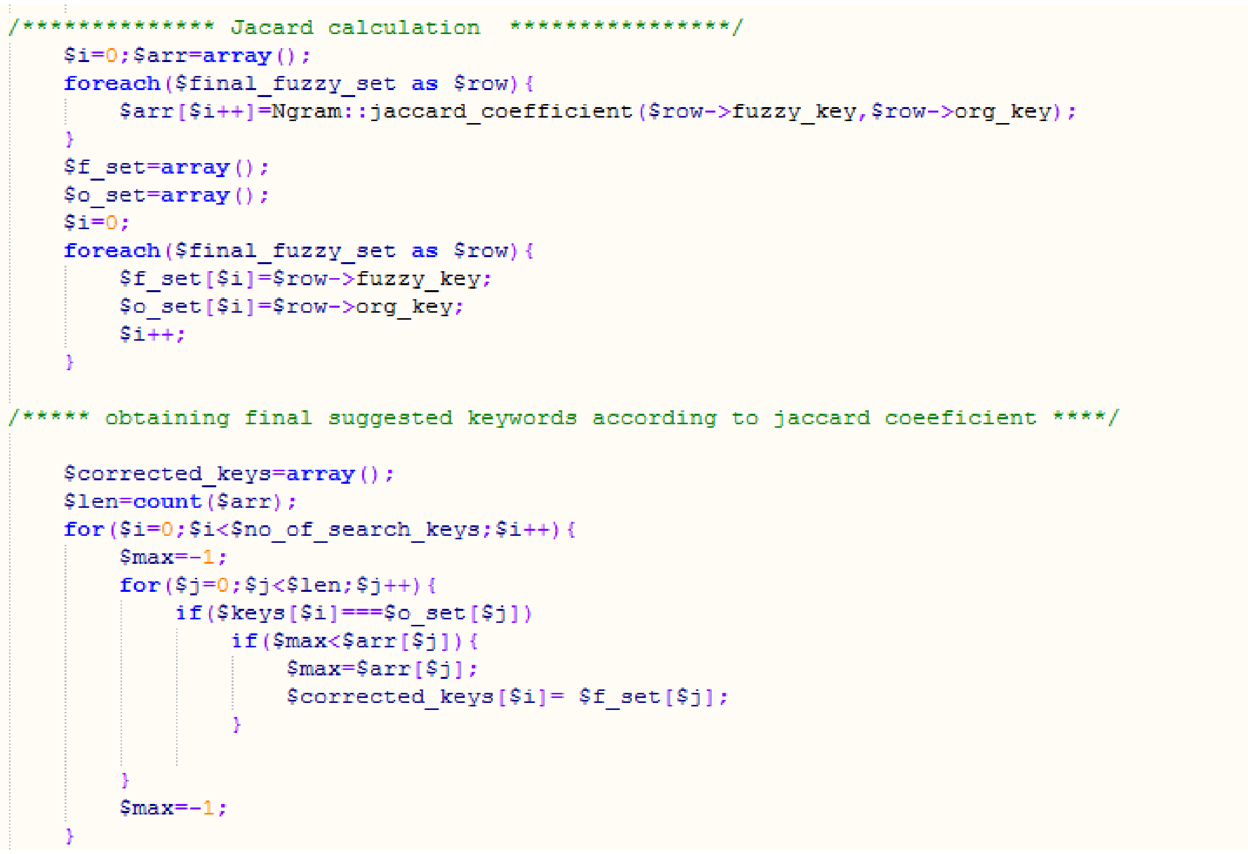
ngram_key	org_key
bCc9tEblDnlyO/9C/uMj1MKETU5r4xzbYPuwPxGZBjQ=	Zmlyc3Q=
Ja7NQ1JdbtMatglvS71YN2yJEK+t4GncHwbwNryPofs=	ZmlsZQ==
Ja7NQ1JdbtMatglvS71YN2yJEK+t4GncHwbwNryPofs=	ZmlsZQ==
Ja7NQ1JdbtMatglvS71YN2yJEK+t4GncHwbwNryPofs=	ZmlsZQ==
Ja7NQ1JdbtMatglvS71YN2yJEK+t4GncHwbwNryPofs=	ZmlsZQ==
UZh1bHamxKOU1aL+XEcFcBMOT8sEvDNZ+76lyeFtSJY=	bWF3YWRh
UZh1bHamxKOU1aL+XEcFcBMOT8sEvDNZ+76lyeFtSJY=	bWF3YWRh
JF7hXcuoTMq5V8RYAEFRkviBGor7SC5VbH9O99efzJY=	eWFzaXI=
UZh1bHamxKOU1aL+XEcFcBMOT8sEvDNZ+76lyeFtSJY=	bWF3YWRh
bCc9tEblDnlyO/9C/uMj1MKETU5r4xzbYPuwPxGZBjQ=	Zmlyc3Q=
Ja7NQ1JdbtMatglvS71YN2yJEK+t4GncHwbwNryPofs=	ZmlsZQ==
mOZBZpa3PdkMab60aowDnwaQepijA8RxV8gZd+yRw38=	c2VjdXJl
mOZBZpa3PdkMab60aowDnwaQepijA8RxV8gZd+yRw38=	c2VjdXJl
UZh1bHamxKOU1aL+XEcFcBMOT8sEvDNZ+76lyeFtSJY=	bWF3YWRh
UZh1bHamxKOU1aL+XEcFcBMOT8sEvDNZ+76lyeFtSJY=	bWF3YWRh

Figure (4.7) keys located in the database contains n-gram key

4.3 Result

In this section, a thorough experimental evaluation of the proposed techniques. The proposed solution introduced by Md Iftekhar Salam et-al[1] aims to study privacy preserving keyword search over encrypted cloud data. Present our implementation of a privacy preserving data storage and retrieval system in cloud computing. Chosen asymmetric/public key encryption schemes (RSA algorithm) . This is a keyword based scheme ensuring faster search functionality; however, limiting the search capability. Also, the scheme is computationally expensive and it reveals the user access pattern.

And in the proposed solution aim trying to be achieved, with high efficiency in respect to time and space. Furthermore the security for the indexes and the trapdoor need to be adequate . Chosen the symmetric/private key encryption schemes (RC4 algorithm) for the encryption/decryption keyword index. This scheme aims to accelerate the search time by looking into the previously searched keywords. For this, the cloud service provider caches the previously searched keywords to avoid the search on all the stored ciphertext. And also using the n-gram for keywords indexes to approximate search the implementation indexes the whole document rather than a set of keyword from each document and using jaccard technique for similarity. Figure 4.8 shows jaccard calculation for similarity

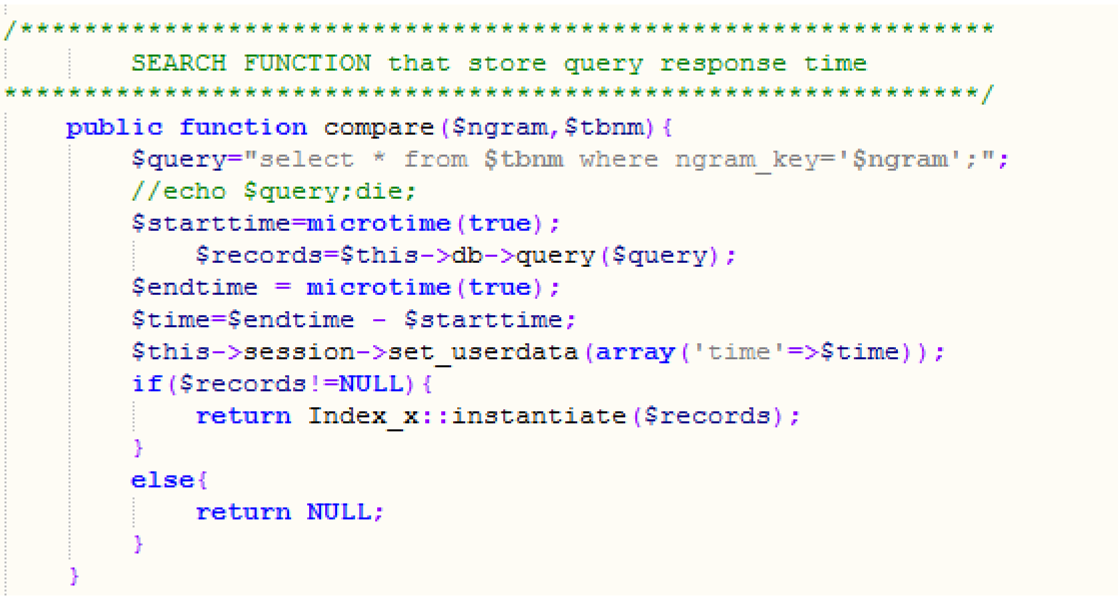


```
/************** Jacard calculation  ****************/
    $i=0;$arr=array();
    foreach($final_fuzzy_set as $row){
        $arr[$i++]=Ngram::jaccard_coefficient($row->fuzzy_key,$row->org_key);
    }
    $f_set=array();
    $o_set=array();
    $i=0;
    foreach($final_fuzzy_set as $row){
        $f_set[$i]=$row->fuzzy_key;
        $o_set[$i]=$row->org_key;
        $i++;
    }

/***** obtaining final suggested keywords according to jaccard coeeficient ****/

    $corrected_keys=array();
    $len=count($arr);
    for($i=0;$i<$no_of_search_keys;$i++){
        $max=-1;
        for($j=0;$j<$len;$j++){
            if($keys[$i]===$o_set[$j])
                if($max<$arr[$j]){
                    $max=$arr[$j];
                    $corrected_keys[$i]= $f_set[$j];
                }

        }
        $max=-1;
    }
```

Figure (4.8) Jaccard calculation

The solution proposed by [1] using tree-based structures to construct indexes for encrypted data search schemes, the corresponding secure search algorithm could be devised to achieve more efficient search than the linear search schemes And using Similarity-Based Ranking To enhance user searching experience and meet more effective data retrieval

In the proposed solution to achieved high efficiency in respect to time. A time function is used to calculate time search for document .figure 4.8 shows search function that store query response time



```
/***************************************************************
        SEARCH FUNCTION that store query response time
***************************************************************/
    public function compare($ngram,$tbnm){
        $query="select * from $tbnm where ngram_key='$ngram';";
        //echo $query;die;
        $starttime=microtime(true);
            $records=$this->db->query($query);
        $endtime = microtime(true);
        $time=$endtime - $starttime;
        $this->session->set_userdata(array('time'=>$time));
        if($records!=NULL){
            return Index_x::instantiate($records);
        }
        else{
            return NULL;
        }
    }
```

Figure (4.9) time search function

4.4 Security analysis

In this subsection, analyze the security of the scheme from three aspects, query privacy, data files, and the searchable index.

4.4.1 Query Privacy

For implementing strong privacy protection, we require that anyone else cannot obtain the query contents other than the query user himself, including the cloud server, data owners, and other data users. Therefore, proposed trapdoor construction achieves strong privacy protection for every time query uses in encrypted form .

4.4.2 Data Files

Data owner uses the semantically secure symmetric encryption scheme to encrypt files before outsourcing. The semantic security of secure symmetric encryption guarantees data files confidentiality.

4.4.3 Searchable Index

Give a keyword set W_i of data file F_i, for each keyword w $\in W_i$, data owner first encrypts the w to be index and calculates n-gram from given keyword. Obviously, the cloud cannot obtain the key word associated with file as long as the n-gram is kept secretly. Thus, in secure index, the cloud cannot determine the key positions of inserting the keyword w. In addition, an adversary may be able to reveal the approximate number of keywords of each data file. To hide the information on how much each data file has keywords in encrypted form.

CHAPTER FIVE

CONCLUSION & RECOMMENDATION

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Searchable encryption represents a new concept for improving storage outsourced servers such as Cloud Computing infrastructures. Cloud Computing storage has several advantages features as a total access to data available everywhere, every time, with scalability and reliability. The main idea from Searchable encryption data is to be able to perform an encrypted query without having to download the whole encrypted data. To solve this problem, chosen the RC4 for the encryption/decryption purpose. and also using the n-gram for keywords indexes the implementation indexes the whole document rather than a set of keyword from each document as proposed in the SSE (symmetric/private key encryption) scheme, which provides the user capability to search any keyword from the document with the trade-off of a slightly larger index size. Also, the client does not need to maintain a keyword index on its side. And allow the client to store encrypted data on an un-trusted server and still be able to securely perform server-side searches within the documents without needing to download and decrypt these documents. the index update process is static in our implementation, which does not allow the addition of new files or updating files.

5.2 Recommendation

There are two areas in which this project could be extended. First implementing the dynamic search that can updating file without downloading it. Second implementing homographic algorithm that can updating data in encrypted form.

REFERENCE

Reference

[1] Salam, I., Yau, W. C., Chin, J. J., Heng, S. H., Ling, H. C., Phan, R. C. W., … Yap, W. S. (2015). Implementation of searchable symmetric encryption for privacy - preserving keyword search on cloud storage. *Human-Centric Computing and Information Sciences*. http://doi.org/10.1186/s13673-015-0039-9

[2] Qiang Tang. (nov 2012). Search in Encrypted Data: Theoretical Models and Practical Applications. *APSIA Group, SnT, University of Luxembourg*, 22.

[3] Tang, Q. (2012). Search in Encrypted Data : Theoretical Models and Practical Applications.

[4] Mykletun, E., Narasimha, M., & Tsudik, G. (2004.). Authentication and Integrity in Outsourced Databases. *Department School Of Information And Computer Science University of California, Irvine*, 10.

[5] system Model (April /3/2018) http://sprout.ics.uci.edu/past_projects/odb/index.html

[6] Outsourced Database Model (April /3/2018) 3 pages Retrieved from http://sprout.ics.uci.edu/past_projects/odb/index.html

[7] Pathak, A. R., & Padmavathi, B. (2014). Analysis of Security Techniques Applied in Database Outsourcing, *Ajeet Ram Pathak et al, / (IJCSIT) International Journal of Computer Science and Information Technologies 5*(1), 665–670.

[8] Advantages and Disadvantages of Outsourcing (April 2018) Retrieved from https://www.flatworldsolutions.com/articles/advantages-disadvantages-outsourcing.php

[9] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public Key Encryption with Keyword Search. In proceedings of Eurocrypt 2004, 2004.

[10] Abdullatif, S., & Labs, B. (2012). Searchable encryption, *research master's computer science (January)*, 1–15.

[11] Mehmet Ucal. (August 2005.). Searching on Encrypted Data. Department of Electrical and Computer Engineering University of Maryland College Park, MD, 18.

[12] Yoshida, R., & Et-al. (2010). Practical Searching Over Encrypted Data By Private Information Retrieval. IEEE, 5.

[13] Anitha, P. (2016). generating secret key for multi-keyword ranked search over encrypted cloud data, *International Research Journal of Engineering and Technology (IRJET)*, 6, 879–884.

[14] RC4 (Mars /0/2018). Retrieved from https://paginas.fe.up.pt/~ei10109/ca/rc4.html

[15] Tiwari, M. (2016.). Fuzzy keyword search over encrypted data. *Impact Journal*, 6.

[16] Zhou, W., Liu, L., Jing, H., Zhang, C., Yao, S., & Wang, S. (2013). K-Gram Based Fuzzy Keyword Search over Encrypted Cloud Computing, *2013*(January), 29–32.